Document Title	System Template
Document Category	Template specification (TPS)
Document UID	63
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• removed FrArTp • Added support for Connectivity Association and Secure Channel for MACsec and CANsec • Added support for TimeBase Cloning • removed SenderReceiverCompositeElementToSignalMapping
2025-11-27	R25-11	AUTOSAR Release Management	• Removed E2E Wrapper and E2E ComCallout configuration options • Reworked DDS configuration • Introduced CyclicHandlingComDataToOsTaskProxyMapping • Replaced LdCom by Large Data Path in Com
2024-11-27	R24-11	AUTOSAR Release Management	• Enhanced PduR Fanout chapters • Added imposition times to constraints • Introduced J1939ProtectedIPdu • Reworked Ethernet Switch configuration: – Added Frame preemption support – Reworked CouplingPortTrafficClassAssignment
2023-11-23	R23-11	AUTOSAR Release Management	• Added support for DDS configuration • Added support for Firewall configuration • Added support for ACL checks • Added support for modeling of IEEE1722Tp streams and Ethernet Switch Filtering and Policing

▽

△

2022-11-24	R22-11	AUTOSAR Release Management	• Added support for MACsec • Added support for Can XL • Clarification of usage of transfer property for signal groups and group signals • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2021-11-25	R21-11	AUTOSAR Release Management	• Rework of Log and Trace model • Rework of TLS modeling using IANA Parameters • Introduction of Affinity Constraints • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2020-11-30	R20-11	AUTOSAR Release Management	• Added support for 10-BASE-T1S • Added support for Software Clusters • Improved RTE Fan-in and RTE Fan-out description • Introduced modeling approach for Service Discovery Service Interfaces on VFB level
2019-11-28	R19-11	AUTOSAR Release Management	• Rework of Ethernet communication model • Added support for Signal-To-Service Translation • Added support for IPsec configuration • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	• Added support for BusMirorring • Reworked the modeling of LinSlaves • Introduced Crypto Infrastructure for SecuredIPdu • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

▽

△

2017-12-08	4.3.1	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Added support for new E2E Profiles 7, 11 and 22 • Improved configuration of Ethernet Switch Ports • Introduced Security Profiles • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2015-07-31	4.2.2	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Introduction of data transformation • Introduction of SecuredIPdu • Introduction of Switch Configuration • Introduction of Global Time Synchronization • Improved support for CanFD • Minor corrections / clarifications / editorial changes; For details please refer to the BWCStatement
2014-03-31	4.1.3	AUTOSAR Release Management	• Various fixes and clarifications
2013-10-31	4.1.2	AUTOSAR Release Management	• Set CanNmCluster.nmChannelActive, FlexrayArTpChannel.timeFrIf and FlexrayArTpChannel.maxFrIf to deprecated • Added SoAd Pdu Collection attributes to SocketConnection • Added SoAdRouting-Group.eventGroupControlType • Introduced SocketAddress.multicastConnector • Clarified usage of ISignal.dataTypePolicy • Described the handling of ComSpecs during flattening

▽

			• Introduced new Pdu types: GeneralPurposePdu and GeneralPurposeIPdu
			• Made RootSwCompositionPrototype.calibrationParameterValueSet "atpSplitable" • Made RootSwCompositionPrototype.flatMap "atpSplitable" • Added new Ethernet addressing attributes to SocketConnection to help to derive the Ecu Configurations for the Server and the Clients
2013-03-15	4.1.1	AUTOSAR Administration	• Added support for remote activation of RunnableEntitys • Added support VLANs and Service Discovery • Reworked the SoAd configuration • Introduced SenderReceiverCompositeElementToSignalMapping and ClientServerToSignalMapping • Added support for CAN FD • Reworked the J1939 TP configuration • Clarification of the usage of swDataDefProps on ISignals and SystemSignals • Added support for Complex Drivers in the Topology • Updated IPduM to allow only static part reception • Added LinSlaveConfig class to the LinMaster • Clarified meaning of PduToFrameMapping.startPosition

△

2011-12-22	4.0.3	AUTOSAR Administration	• Added support for Partial Networking • Added support for Complex Drivers • Added support for new COM transfer properties • Added support for transmission mode switch via Com_SwitchIpduTxMode COM API • Added support for treating byte arrays with primitive type mapping • Added support for partial routing in signal gateways • Added support for FlexRay AUTOSAR TP • Added rules for creation of Pdu Triggerings and Pdu Ports • Explained the general approach of bit counting
2009-12-18	4.0.1	AUTOSAR Administration	• updated System class category names • Changed specification of PduLength parameter from bits to bytes • Made Flexray channel specific attributes optional • Clarified the usage of EcuPorts in System Extract/Ecu Extract • Allowed to define sending and receiving connections to EcuPorts for NmPdus, XcpPdus • Aligned FrTP model to AUTOSAR FrTp SWS • Replaced ComProcessingPeriod by three timebase parameters • Reworked E2E protection of selected I-PDUs • Corrected AssignFrameIdRange configuration in LIN model • Clarified the routing of ISignalGroups in the Signal Gateway

▽

△

			△ • Extended the enumeration "TransferPropertyEnum" with the element "triggeredOnChange" • Added a subchapter to the appendix about special use cases that are supported by the System Template • Reworked SenderReceiverToSignalGroupMapping and ClientServerToSignalGroupMapping • Changed multiplicity between System and SystemMapping from 1 to 0..1.
2009-12-04	3.1.4	AUTOSAR Administration	• Implemented support for LIN 2.1 • Implemented support for Network Management (FlexRayNm, CanNm, LinNm, UdpNm) • adapted IPdu Multiplexer model to ASAM Fibex 3.1 • Reworked "ECU Extract" chapter • Introduced "System Extract" • Introduced EndToEndProtection for ISignalIPdus • Reworked "Transport Layer" chapter • Implemented Variant Handling concept • Implemented Documentation support concept • Implemented support for J1939 communication • Implemented support for TTCan • Implemented support for for TCP/IP and DoIP. • Introduced Pdu Counter and Pdu Replication • Implemented VMM/AMM concept • Introduced low-level routing of NPdu's • Implemented support for dynamic signals • Introduced PdurIPduGroups

▽

△

2009-02-04	3.1.2	AUTOSAR Administration	• Clarified semantics of Data Mappings • Added inheritance from Identifiable to PduToFrameMapping • Added "FlexRayChannelName" attribute to FlexRayPhysicalChannel element.
2008-08-13	3.1.1	AUTOSAR Administration	• Added the boolean attribute "payloadPreambleIndicator" to the "FlexrayFrameTriggering". • Added extension that allows the assignment of IPduGroups to ECUs. • Added missing reference from "ClientServerComposite-TypeMapping" to "ArgumentPrototype" • Alignment with AUTOSAR IPduM SWS
2008-02-01	3.0.2	AUTOSAR Administration	• Moved "canAddressingMode" attribute from "CanCluster" to the "CanFrameTriggering" element • Clarified the descriptions of several elements and attributes.
2007-12-21	3.0.1	AUTOSAR Administration	• Communication part reworked from scratch • Alignment with ECU Configuration • Added support for Transport Protocols • Major changes in Topology chapter after harmonisation with Fibex (removed complex Topologies) • Document meta information extended • Small layout adaptations made
2006-11-28	2.1	AUTOSAR Administration	• Support for Signal Groups added. • Rework of the Topology Description • Introduction of PDUs. Description of the PDU Multiplexer, PDU Gateway. • FlexRay: multiple transmission of a frame within one communication cycle is supported now. • Removed the concept of Variant Descriptions (Properties) and CompToECUMappingConstraints relying on the property concept.

▽

▽

△

			△ • Split SwCompToEcuMapping in two classes in order to allow separation of SWC-to-ECU mapping and Implementation-to-SWC mapping. • Removed preliminary chapter on MOST as it is not part of the standard. • For all Instance References in the System Template added diagrams to the meta-model containing detailed representations of these references.
2005-05-31	1.0	AUTOSAR Administration	• Initial Release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction 28
1.1 Methodology for Defining Formal Template 28
1.2 Scope 29
1.3 UML Meta-Model 32
1.4 Document Conventions 34
1.4.1 Detailed Representation of InstanceRef Associations 36
1.4.2 Variant Handling 36
1.4.3 Timing Extensions 36
1.4.4 Documentation Support 37
1.4.5 Stereotype atpSplitable in the System Template 38
1.5 AUTOSAR System Template and ASAM FIBEX 38
2 System 40
2.1 Data interpretation of bus content in different contexts 44
2.1.1 Nm NID/CBV 44
2.2 ClientIdDefinitionSet 45
2.3 System ComSpec Definition 46
2.3.1 Server ComSpec 47
2.3.2 Queued Receiver ComSpec 49
2.3.3 ModeSwitch Sender ComSpec 50
2.4 InterpolationRoutingMappingSet 52
3 Topology 54
3.1 ECUs and their communication capabilities 55
3.1.1 ECU Instance 55
3.1.2 Communication Controller 59
3.1.3 Communication Connector 60
3.1.4 Internal ECU handling upon a communication wakeup 63
3.2 Communication Clustering 70
3.2.1 Communication Cluster 70
3.2.2 Physical Channel 71
3.3 Specialized Attributes of the Topology Entities 73
3.3.1 CAN 74
3.3.1.1 CAN Cluster 75
3.3.1.2 CAN Communication Controller 76
3.3.1.3 CAN Physical Channel 86
3.3.1.4 CAN Communication Connector 86
3.3.2 SAE J1939 87
3.3.2.1 J1939Node 89
3.3.3 FlexRay 96
3.3.3.1 FlexRay Cluster 97
3.3.3.2 FlexRay Communication Controller 102

3.3.3.3 FlexRay Communication Connector . . . 107
3.3.3.4 FlexRay Physical Channel . . . 107
3.3.4 LIN . . . 109
3.3.4.1 LIN Cluster . . . 110
3.3.4.2 LIN Communication Controller . . . 111
3.3.4.3 LIN Master . . . 112
3.3.4.4 LIN Slave . . . 114
3.3.4.5 LIN Communication Connector . . . 116
3.3.4.6 LIN Physical Channel . . . 118
3.3.5 Ethernet . . . 119
3.3.5.1 Ethernet Cluster . . . 121
3.3.5.2 Ethernet Physical Channel . . . 122
3.3.5.3 Ethernet Coupling Elements and Coupling Ports . . . 125
3.3.5.4 Ethernet Communication Controller . . . 134
3.3.5.5 Ethernet Communication Connector . . . 136
3.3.5.6 Ethernet Switch Driver . . . 137
3.3.5.7 TcpIp stack configuration properties . . . 172
3.3.5.8 Ethernet wake-up and sleep on dataline . . . 188
3.3.6 10BASE-T1S Ethernet . . . 199
3.3.7 Connectivity Association for MACsec and CANsec . . . 203
3.3.7.1 MACsec Connectivity Association . . . 203
3.3.7.2 CANsec Connectivity Association . . . 211
3.3.8 MACsec . . . 218
3.3.9 CDD . . . 226
3.4 Mapping of Topology Entities onto Hardware Elements . . . 228
3.4.1 ECU Mapping . . . 229
3.4.2 Communication Controller Mapping . . . 231
3.4.3 HW-Port Mapping . . . 231
4 Top-level Software Composition 233
5 Mapping 235
5.1 Software Component Mapping . . . 243
5.1.1 SW Component to ECU Mapping . . . 244
5.1.2 Software Component to Implementation Mapping . . . 248
5.1.3 SW Component to Partition Mapping . . . 249
5.1.4 Software Component Mapping Constraints . . . 252
5.1.4.1 ComponentClustering . . . 253
5.1.4.2 ComponentSeparation . . . 254
5.1.5 Affinity Constraints . . . 256
5.1.5.1 RteEvent to OsTaskProxy mapping constraints in the context of a Software Composition . . . 261
5.1.5.2 RteEvent to OsTaskProxy mapping constraints in the context of the System . . . 263
5.2 Data Mapping . . . 266

5.2.1 Mapping of Variable Data Prototypes on System Signals 276
5.2.1.1 Mapping of Variable Data Prototypes with primitive datatypes on System Signals (Sender-Receiver Communication) 279
5.2.1.2 Mapping of Variable Data Prototypes with composite datatypes (Sender-Receiver Communication) 286
5.2.1.3 Mapping of Client Server Operations to System Signals . . . 298
5.2.1.4 Mapping of Trigger to SystemSignal 306
5.2.2 Signal Path Constraint 309
5.2.2.1 CommonSignalPath 311
5.2.2.2 ForbiddenSignalPath 314
5.2.2.3 PermissibleSignalPath 315
5.2.2.4 SeparateSignalPath 316
5.3 Compatibility Rules for DataMapping between a DataPrototype and a SystemSignal 317
5.3.1 DataPrototype typed by category VALUE 317
5.3.2 DataPrototype typed by category BOOLEAN 320
5.3.3 DataPrototype typed by category STRING 322
5.3.4 DataPrototype typed by an ARRAY 322
5.3.5 Compatibility of SwDataDefProps 323
5.4 RTE and basic software resource estimations 325
5.5 Communication Management Mapping 330
5.5.1 Partial Networking 330
5.5.1.1 Partial Networking and managed Ethernet switch 338
5.5.1.2 Partial Network Gateway 342
5.5.1.3 Checking the consistency of partial network clusters (PNC) . 343
5.5.1.4 Dynamic Partial Network Cluster Mapping 349
5.5.2 Com Management Mapping 352
5.6 Software Cluster Mapping 353
5.7 Dds Mapping 362
5.8 EcuPartition to Core Mapping 365
5.9 Cyclic Handling of Communication Data in the RTE Mapping 366
6 Communication 370
6.1 Triggerings and Ports 372
6.1.1 Ownership of PduTriggerings 385
6.1.2 Ownership of ISignalTriggerings 387
6.2 ISignals 389
6.2.1 Large Data Path in COM 414
6.2.2 Big Endian and Little Endian memory layout of Pdus and Frames 417
6.3 PDUs 421
6.3.1 ContainerIPdu 436
6.3.2 SecuredIPdu 447
6.3.2.1 Crypto Infrastructure for SecuredIPdu 459
6.3.3 J1939ProtectedIPdu 466

6.3.4 GeneralPurposeConnection . . . 469
6.4 IPdu Timing . . . 470
6.4.1 Data Filter configuration . . . 476
6.4.2 Cyclic Timing . . . 478
6.4.3 EventControlled Timing . . . 479
6.4.4 Configuration of a trigger for COM_TriggerIPduSend API call . . . 481
6.5 I-Pdu Multiplexer . . . 483
6.5.1 I-Pdu Multiplexer in System Extract/ECU Extract . . . 497
6.6 Frames . . . 499
6.7 Specialized Attributes of the Communication Entities . . . 502
6.7.1 FlexRay specific description . . . 502
6.7.2 LIN specific description . . . 509
6.7.2.1 LIN Frames . . . 510
6.7.2.2 LIN Schedule Table . . . 515
6.7.2.3 Configuration Services . . . 519
6.7.3 CAN specific description . . . 526
6.7.3.1 SAE J1939 Protocol specific description . . . 531
6.7.3.2 CAN XL specific description . . . 532
6.7.4 Ethernet specific description . . . 534
6.7.4.1 ApplicationEndpoint . . . 540
6.7.4.2 NetworkEndpoint . . . 546
6.7.4.3 Service Instances . . . 552
6.7.4.4 Multicast Subscription . . . 557
6.7.4.5 SOME/IP specific description . . . 561
6.7.4.6 DDS Specific Description . . . 605
6.7.4.7 StaticSocketConnection . . . 636
6.7.4.8 Service Discovery Message Configuration . . . 641
6.7.4.9 NmPdu data exchange over StaticSocketConnection . . . 645
6.7.4.10 Diagnostics over IP . . . 647
6.7.4.11 Transport Layer Security . . . 654
6.7.4.12 IPsec . . . 666
6.7.4.13 EthernetFrameType based communication . . . 675
6.7.4.14 Restriction in usage of `ProvidedServiceInstance.majorVersion` . . . 677
6.7.4.15 Summary of supported configuration options between ApplicationEndpoints . . . 678
6.7.4.16 Firewall . . . 679
6.7.4.17 ACL check . . . 688
6.8 Transport Layer . . . 690
6.8.1 Transport Layer Routing . . . 692
6.8.2 FlexRay ISO Transport Layer . . . 693
6.8.3 CAN Transport Layer . . . 702
6.8.4 LIN Transport Layer . . . 710

6.8.5 Ethernet Transport Layer . . . 715
6.8.6 SOME/IP segmenter . . . 717
6.8.7 SAE J1939 Transport Layer . . . 720
6.8.7.1 Configuration of J1939Tp NPDUs . . . 728
6.8.8 Unicast TP Example . . . 735
6.8.9 Multicast TP Example . . . 739
6.8.10 Diagnostic Connection . . . 741
6.8.11 IEEE1722 Transport Layer . . . 744
6.8.11.1 Audio Video Transport . . . 749
6.8.11.2 Audio Video Transport CRF . . . 751
6.8.11.3 Audio Video Transport AAF . . . 753
6.8.11.4 Audio Video Transport using IIDC and IEC61883 . . . 759
6.8.11.5 Audio Video Transport RVF . . . 761
6.8.11.6 Control Data Transport . . . 767
6.8.11.7 Control Data Transport CAN . . . 771
6.8.11.8 Control Data Transport LIN . . . 779
6.9 Network Management . . . 782
6.9.1 FlexRay Network Management . . . 793
6.9.2 CAN Network Management . . . 797
6.9.3 UDP Network Management . . . 802
6.9.4 J1939 Network Management . . . 808
6.9.4.1 J1939SharedAddressCluster . . . 810
6.9.5 Managed Channels . . . 812
6.9.5.1 Ethernet VLANs . . . 812
6.9.5.2 Ethernet tunneling through CAN XL . . . 813
6.10 Bus Mirroring . . . 813
6.10.1 CAN Destination Channel . . . 817
6.10.2 FlexRay Destination Channel . . . 821
6.10.3 Ethernet Destination Channel . . . 823
6.10.4 User Defined Destination Channel . . . 824
6.11 Fan-out . . . 825
6.11.1 Signal fan-out . . . 825
6.11.1.1 RTE signal fan-out . . . 825
6.11.1.2 COM Signal Gateway fan-out . . . 831
6.11.2 Pdu fan-out . . . 833
6.11.2.1 Pdu Router fan-out . . . 833
6.11.2.2 Flexray Interface fan-out . . . 837
6.11.2.3 Socket Adapter fan-out . . . 838
6.11.3 Frame fan-out . . . 841
6.12 Fan-in . . . 841
6.12.1 RTE fan-in . . . 841
6.12.2 Pdu fan-in . . . 845
6.12.3 IPdu Container fan-in . . . 845

6.13 Log and Trace . . . 846
6.14 Support of Complex Drivers . . . 851
6.15 MetaData in SenderReceiverInterface . . . 852
6.16 Signal Service Translation . . . 853
6.16.1 Architectural setup . . . 854
6.16.1.1 Method handling . . . 856
6.16.2 Mapping description . . . 856
6.16.2.1 Filters and Transmission Triggers . . . 861
6.16.2.2 Mapping description from several sources . . . 866
6.16.2.3 Mapping description and data conversion . . . 869
6.16.2.4 Implementation of PassThroughSwConnectors . . . 869
6.16.3 Service discovery control . . . 871
6.16.3.1 Service control at ECU start . . . 872
6.16.3.2 Service control due to availability of partial networks . . . 873
6.16.3.3 Service control due to availability of related service instance . . . 875
6.16.4 Translation behavior . . . 877
6.16.4.1 COM-Stack translation behavior . . . 878
6.16.4.2 Translation Application Software Component translation behavior . . . 878
6.16.5 End-to-End considerations . . . 879
6.16.5.1 Safety . . . 879
6.16.5.2 Security . . . 884
7 Data Transformation 885
7.1 Outline . . . 885
7.1.1 Configuration of the Communication Layout . . . 885
7.1.2 Data Transformation by Software . . . 886
7.2 Use Cases . . . 887
7.2.1 Transmission of large composite data types over networks with large PDUs (e.g Ethernet) . . . 887
7.2.2 Support of transmission from one sender to multiple receivers with Signal Fan-out . . . 888
7.2.3 Support of transmission from one sender to multiple receivers with PDU Fan-out . . . 888
7.2.4 Transformer Chaining . . . 889
7.2.5 Signal Group Based interaction of the transformer with the Com module . . . 890
7.3 Transformer configuration . . . 890
7.3.1 Generic Transformer . . . 906
7.3.2 SOME/IP Transformer . . . 907
7.3.2.1 SOME/IP Transformation Properties on the level of DataPrototypes . . . 913
7.3.2.2 Network Representation . . . 924
7.3.3 COM Based Transformer . . . 934

7.3.4 DDS Transformer . . . 938
7.3.5 E2E Transformer . . . 940
7.3.5.1 E2E state machine settings . . . 956
7.3.5.2 E2E recommended configuration settings . . . 957
7.3.6 UserDefined Transformer . . . 967
7.3.7 Support for TLV Encoding . . . 968
7.3.7.1 Assignment of TLV Data Ids . . . 968
7.3.7.2 Assignment of TLV Wire Type . . . 974
8 Gateways 976
8.1 Frame Mapping . . . 978
8.2 IPdu Mapping . . . 980
8.2.1 Examples of IPdu Gateway configurations . . . 983
8.2.2 Usage of IPduMapping.pduMaxLength . . . 986
8.2.3 Routing and processing of Diagnostics Pdus . . . 989
8.3 Signal Mapping . . . 990
8.3.1 Partial Signal Group Mapping . . . 992
9 Global Time Synchronization 994
9.1 Introduction . . . 994
9.2 The big Picture . . . 995
9.2.1 Cloning of Global Time Information . . . 1012
9.2.2 Reverse Sync Method . . . 1014
9.3 Detailed Description of Global Time Synchronization . . . 1015
9.3.1 Time Synchronization over CAN . . . 1015
9.3.2 Time Synchronization over Ethernet . . . 1017
9.3.2.1 Time Synchronization and Ethernet propagation delay . . . 1022
9.3.2.2 Time Synchronization and Ethernet connection . . . 1022
9.3.2.3 Time Synchronization and managed Ethernet switch . . . 1023
9.3.2.4 Fine-Grained GlobalTimeDomain Ethernet switch configuration . . . 1028
9.3.3 Time Synchronization over FlexRay . . . 1030
9.3.4 Time Synchronization by user defined Timebase Provider . . . 1033
9.3.5 Time Synchronization Common Properties . . . 1034
10 Description of Service Discovery Services in Classic Platform 1036
10.1 Representation of Service Interfaces on VFB level . . . 1036
10.1.1 Representation of Events . . . 1037
10.1.2 Representation of Methods . . . 1037
10.1.3 Representation of Fire and Forget Methods . . . 1037
10.1.4 Representation of Fields . . . 1038
10.2 Representation of Service Interfaces on network level . . . 1040

11 Software Cluster 1046
11.1 Big Picture . . . 1046
11.2 Software Cluster Resources . . . 1054
11.3 Software Cluster Binary Manifest . . . 1065
11.4 Software Cluster Extraction . . . 1081
11.4.1 Software Cluster Extraction and DataMappings . . . 1084
12 Usage of the System Template 1086
12.1 System Constraint Description . . . 1087
12.2 Abstract System Description . . . 1088
13 System Extract of the System Configuration Description 1090
13.1 OEM/Supplier Collaboration Scenario . . . 1091
13.2 Data Mapping in the System Extract . . . 1093
13.3 SW component inclusion and top level data mapping . . . 1096
13.4 Port-Interface Mapping in the System Extract . . . 1098
14 ECU Extract of the System Configuration Description 1100
14.1 Topology . . . 1101
14.2 Top-level Software Composition . . . 1102
14.2.1 ECU Flat view . . . 1104
14.2.2 Internal Communication . . . 1105
14.2.3 External Communication . . . 1107
14.2.4 Port Groups . . . 1112
14.2.5 Service Needs . . . 1113
14.3 Extending the ECU Software Composition . . . 1114
14.4 Communication . . . 1115
14.4.1 Frame . . . 1116
14.4.2 PDU . . . 1116
14.4.3 ISignals and ISignalGroups . . . 1116
14.4.4 SystemSignal and SystemSignalGroup . . . 1117
14.4.5 Gateways . . . 1118
14.4.6 TP configuration . . . 1118
14.4.7 NM configuration . . . 1118
14.5 Naming Issues . . . 1119
14.5.1 Package Structure . . . 1119
14.5.2 Naming of Measurement and Calibration Data . . . 1120
14.5.3 Naming of Derived Elements . . . 1120
14.5.4 Re-use of short names assigned in previous iterations . . . 1121
14.6 ECU Extract in subsequent Cycles of Iterative Development . . . 1122
14.6.1 Traceability of model elements created in ECU Extract . . . 1122
14.6.2 Mapping of AUTOSAR attributes to ASAM ASAP2 . . . 1129
14.6.3 Assigning communication graphs to RTE Implementation Plug-Ins 1129
14.7 Variant Handling in ECU Extract . . . 1132
14.7.1 System Constants . . . 1132

14.7.2 Nested Whole/Part class variants . . . 1133
14.7.3 Multiple instances of calibration parameters in system scope . . . 1133
15 Supported special use-cases 1135
15.1 Support of sending / receiving same Can/Flexray Frame on same channel (Pdu Gateway Use-Case) . . . 1135
15.2 Support of sending / receiving same Can/Flexray Frame on same channel (bidirectional routing in COM) . . . 1136
15.3 Support of dynamic CAN IDs . . . 1139
15.4 N:1 Sender Receiver communication description in a System Extract over one PhysicalChannel . . . 1139
A Reference Material 1142
A.1 Abbreviations . . . 1142
A.2 Imposition Times of Constraints . . . 1142
B Detailed Representation of InstanceRef Associations in the System Template 1144
B.1 Usage of InstanceRefs in Data Mapping diagrams . . . 1144
B.2 Usage of InstanceRefs in SW Mapping diagrams . . . 1146
B.3 Usage of InstanceRefs in Signal Path Constraint diagrams . . . 1147
B.4 Usage of InstanceRefs in PncMapping . . . 1147
B.5 Usage of InstanceRefs in ComManagementMapping . . . 1148
B.6 “SWC in System” InstanceRef . . . 1149
B.7 “ModeDeclarationGroupPrototype in System” InstanceRef . . . 1151
B.8 “Operation in System” InstanceRef . . . 1152
B.9 “VariableDataPrototype” InstanceRef . . . 1154
B.10 “PortGroup in System” InstanceRef . . . 1157
B.11 “DataPrototype in PortInterface” InstanceRef . . . 1158
B.12 Affinity Constraints InstanceRef . . . 1159
C Harmonization between Upstream Templates and ECU Configuration 1161
C.1 Can . . . 1162
C.2 CanIf . . . 1172
C.3 CanNm . . . 1176
C.4 CanSM . . . 1191
C.5 CanTSyn . . . 1193
C.6 CanTp . . . 1205
C.7 CanTrcv . . . 1219
C.8 Cdd . . . 1223
C.9 Com . . . 1223
C.10 ComM . . . 1271
C.11 Csm . . . 1281
C.12 Dcm . . . 1282
C.13 Dds . . . 1284
C.14 Dlt . . . 1354

C.15 DoIP 1361
C.16 EcuC 1371
C.17 EcuM 1376
C.18 Eth 1378
C.19 EthIf 1389
C.20 EthSM 1393
C.21 EthSwt 1394
C.22 EthTSyn 1430
C.23 EthTrcv 1452
C.24 Firewall 1465
C.25 Fr 1512
C.26 FrIf 1529
C.27 FrNm 1549
C.28 FrSM 1563
C.29 FrTSyn 1564
C.30 FrTp 1574
C.31 FrTrcv 1585
C.32 IEEE1722Tp 1585
C.33 IKE 1612
C.34 ISO15118Chrg 1614
C.35 IdsM 1615
C.36 IpduM 1615
C.37 J1939Dcm 1644
C.38 J1939Nm 1650
C.39 J1939Rm 1661
C.40 J1939Tp 1666
C.41 Lin 1682
C.42 LinIf 1684
C.43 LinTp 1704
C.44 LinTrcv 1708
C.45 Mirror 1709
C.46 Mka 1737
C.47 Nm 1748
C.48 Os 1759
C.49 PduR 1761
C.50 Rte 1764
C.51 Sd 1766
C.52 SecOC 1799
C.53 SoAd 1822
C.54 SomeIpTp 1845
C.55 StbM 1851
C.56 SwCluC 1857
C.57 TcpIp 1858

C.58 UdpNm . . . 1936
C.59 VdpCmR . . . 1949
C.60 Xcp . . . 1950
C.61 Xfrm . . . 1952
D Glossary 1954
E History of Constraints and Specification Items 1957
E.1 Constraint and Specification Item History of this document according to AUTOSAR R4.0.1 . . . 1957
E.1.1 Changed Constraints in R4.0.1 . . . 1957
E.1.2 Added Constraints in R4.0.1 . . . 1957
E.1.3 Deleted Constraints in R4.0.1 . . . 1957
E.2 Constraint and Specification Item History of this document according to AUTOSAR R4.0.2 . . . 1958
E.2.1 Changed Constraints in R4.0.2 . . . 1958
E.2.2 Added Constraints in R4.0.2 . . . 1958
E.2.3 Deleted Constraints in R4.0.2 . . . 1958
E.3 Constraint and Specification Item History of this document according to AUTOSAR R4.0.3 . . . 1958
E.3.1 Changed Constraints in R4.0.3 . . . 1958
E.3.2 Changed Specification Items in R4.0.3 . . . 1958
E.3.3 Added Constraints in R4.0.3 . . . 1958
E.3.4 Added Specification Items in R4.0.3 . . . 1959
E.3.5 Deleted Constraints in R4.0.3 . . . 1959
E.3.6 Deleted Specification Items in R4.0.3 . . . 1959
E.4 Constraint and Specification Item History of this document according to AUTOSAR R4.1.1 . . . 1959
E.4.1 Changed Constraints in R4.1.1 . . . 1959
E.4.2 Changed Specification Items in R4.1.1 . . . 1959
E.4.3 Added Constraints in R4.1.1 . . . 1959
E.4.4 Added Specification Items in R4.1.1 . . . 1961
E.4.5 Deleted Constraints in R4.1.1 . . . 1964
E.4.6 Deleted Specification Items in R4.1.1 . . . 1964
E.5 Constraint and Specification Item History of this document according to AUTOSAR R4.1.2 . . . 1965
E.5.1 Changed Specification Items in R4.1.2 . . . 1965
E.5.2 Added Specification Items in R4.1.2 . . . 1965
E.5.3 Added Constraints in R4.1.2 . . . 1965
E.5.4 Changed Constraints in R4.1.2 . . . 1966
E.5.5 Deleted Constraints in R4.1.2 . . . 1966
E.6 Constraint and Specification Item History of this document according to AUTOSAR R4.1.3 . . . 1966
E.6.1 Changed Specification Items in R4.1.3 . . . 1966
E.6.2 Added Specification Items in R4.1.3 . . . 1966

E.6.3 Deleted Specification Items in R4.1.3 . . . 1966
E.6.4 Added Constraints in R4.1.3 . . . 1967
E.6.5 Changed Constraints in R4.1.3 . . . 1967
E.6.6 Deleted Constraints in R4.1.3 . . . 1967
E.7 Constraint and Specification Item History of this document according to AUTOSAR R4.2.1 . . . 1967
E.7.1 Added Specification Items in R4.2.1 . . . 1967
E.7.2 Changed Specification Items in R4.2.1 . . . 1969
E.7.3 Deleted Specification Items in R4.2.1 . . . 1970
E.7.4 Added Constraints in R4.2.1 . . . 1970
E.7.5 Changed Constraints in R4.2.1 . . . 1973
E.7.6 Deleted Constraints in R4.2.1 . . . 1974
E.8 Constraint and Specification Item History of this document according to AUTOSAR R4.2.2 . . . 1974
E.8.1 Added Specification Items in R4.2.2 . . . 1974
E.8.2 Changed Specification Items in R4.2.2 . . . 1975
E.8.3 Deleted Specification Items in R4.2.2 . . . 1975
E.8.4 Added Constraints in R4.2.2 . . . 1975
E.8.5 Changed Constraints in R4.2.2 . . . 1976
E.8.6 Deleted Constraints in R4.2.2 . . . 1978
E.9 Constraint and Specification Item History of this document according to AUTOSAR R4.3.0 . . . 1978
E.9.1 Added Specification Items in R4.3.0 . . . 1978
E.9.2 Changed Specification Items in R4.3.0 . . . 1980
E.9.3 Deleted Specification Items in R4.3.0 . . . 1980
E.9.4 Added Constraints in R4.3.0 . . . 1981
E.9.5 Changed Constraints in R4.3.0 . . . 1983
E.9.6 Deleted Constraints in R4.3.0 . . . 1985
E.10 Constraint and Specification Item History of this document according to AUTOSAR R4.3.1 . . . 1985
E.10.1 Added Specification Items in R4.3.1 . . . 1985
E.10.2 Changed Specification Items in R4.3.1 . . . 1986
E.10.3 Deleted Specification Items in R4.3.1 . . . 1986
E.10.4 Added Constraints in R4.3.1 . . . 1987
E.10.5 Changed Constraints in R4.3.1 . . . 1988
E.10.6 Deleted Constraints in R4.3.1 . . . 1988
E.11 Constraint and Specification Item History of this document according to AUTOSAR R4.4.0 . . . 1988
E.11.1 Added Specification Items in R4.4.0 . . . 1988
E.11.2 Changed Specification Items in R4.4.0 . . . 1989
E.11.3 Deleted Specification Items in R4.4.0 . . . 1990
E.11.4 Added Constraints in R4.4.0 . . . 1990
E.11.5 Changed Constraints in R4.4.0 . . . 1992
E.11.6 Deleted Constraints in R4.4.0 . . . 1993

E.12 Constraint and Specification Item History of this document according to AUTOSAR R19-11 . . . 1993
E.12.1 Added Specification Items in R19-11 . . . 1993
E.12.2 Changed Specification Items in R19-11 . . . 1996
E.12.3 Deleted Specification Items in R19-11 . . . 1998
E.12.4 Added Constraints in R19-11 . . . 1998
E.12.5 Changed Constraints in R19-11 . . . 2001
E.12.6 Deleted Constraints in R19-11 . . . 2002
E.13 Constraint and Specification Item History of this document according to AUTOSAR R20-11 . . . 2003
E.13.1 Added Specification Items in R20-11 . . . 2003
E.13.2 Changed Specification Items in R20-11 . . . 2006
E.13.3 Deleted Specification Items in R20-11 . . . 2007
E.13.4 Added Constraints in R20-11 . . . 2007
E.13.5 Changed Constraints in R20-11 . . . 2011
E.13.6 Deleted Constraints in R20-11 . . . 2013
E.14 Constraint and Specification Item History of this document according to AUTOSAR R21-11 . . . 2013
E.14.1 Added Specification Items in R21-11 . . . 2013
E.14.2 Changed Specification Items in R21-11 . . . 2015
E.14.3 Deleted Specification Items in R21-11 . . . 2016
E.14.4 Added Constraints in R21-11 . . . 2017
E.14.5 Changed Constraints in R21-11 . . . 2018
E.14.6 Deleted Constraints in R21-11 . . . 2020
E.15 Constraint and Specification Item History of this document according to AUTOSAR R22-11 . . . 2021
E.15.1 Added Specification Items in R22-11 . . . 2021
E.15.2 Changed Specification Items in R22-11 . . . 2022
E.15.3 Deleted Specification Items in R22-11 . . . 2023
E.15.4 Added Constraints in R22-11 . . . 2023
E.15.5 Changed Constraints in R22-11 . . . 2024
E.15.6 Deleted Constraints in R22-11 . . . 2025
E.15.7 Added Advisories in R22-11 . . . 2025
E.15.8 Changed Advisories in R22-11 . . . 2025
E.15.9 Deleted Advisories in R22-11 . . . 2025
E.16 Constraint and Specification Item History of this document according to AUTOSAR R23-11 . . . 2026
E.16.1 Added Specification Items in R23-11 . . . 2026
E.16.2 Changed Specification Items in R23-11 . . . 2027
E.16.3 Deleted Specification Items in R23-11 . . . 2028
E.16.4 Added Constraints in R23-11 . . . 2028
E.16.5 Changed Constraints in R23-11 . . . 2038
E.16.6 Deleted Constraints in R23-11 . . . 2039
E.16.7 Added Advisories in R23-11 . . . 2039

E.16.8 Changed Advisories in R23-11 . . . 2039
E.16.9 Deleted Advisories in R23-11 . . . 2039
E.17 Constraint and Specification Item History of this document according to AUTOSAR R24-11 . . . 2039
E.17.1 Added Specification Items in R24-11 . . . 2039
E.17.2 Changed Specification Items in R24-11 . . . 2041
E.17.3 Deleted Specification Items in R24-11 . . . 2041
E.17.4 Added Constraints in R24-11 . . . 2041
E.17.5 Changed Constraints in R24-11 . . . 2043
E.17.6 Deleted Constraints in R24-11 . . . 2045
E.17.7 Added Advisories in R24-11 . . . 2045
E.17.8 Changed Advisories in R24-11 . . . 2045
E.17.9 Deleted Advisories in R24-11 . . . 2045
E.18 Constraint and Specification Item History of this document according to AUTOSAR R25-11 . . . 2046
E.18.1 Added Specification Items in R25-11 . . . 2046
E.18.2 Changed Specification Items in R25-11 . . . 2048
E.18.3 Deleted Specification Items in R25-11 . . . 2049
E.18.4 Added Constraints in R25-11 . . . 2050
E.18.5 Changed Constraints in R25-11 . . . 2052
E.18.6 Deleted Constraints in R25-11 . . . 2053
E.18.7 Added Advisories in R25-11 . . . 2054
E.18.8 Changed Advisories in R25-11 . . . 2054
E.18.9 Deleted Advisories in R25-11 . . . 2054
E.19 Constraint and Specification Item History of this document according to AUTOSAR R26-11 . . . 2054
E.19.1 Added Specification Items in R26-11 . . . 2054
E.19.2 Changed Specification Items in R26-11 . . . 2055
E.19.3 Deleted Specification Items in R26-11 . . . 2055
E.19.4 Added Constraints in R26-11 . . . 2056
E.19.5 Changed Constraints in R26-11 . . . 2057
E.19.6 Deleted Constraints in R26-11 . . . 2058
E.19.7 Added Advisories in R26-11 . . . 2059
E.19.8 Changed Advisories in R26-11 . . . 2059
E.19.9 Deleted Advisories in R26-11 . . . 2059
F Mentioned Class Tables 2060
G Non-Splitable Elements in this Template 2152
H Variation Points in the Scope of this Document 2154

References

[1] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate

[2] AUTOSAR XML Schema Production Rules
AUTOSAR_FO_TPS_XMLSchemaProductionRules

[3] Methodology for Classic Platform
AUTOSAR_CP_TR_Methodology

[4] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[5] Specification of ECU Resource Template
AUTOSAR_CP_TPS_ECUResourceTemplate

[6] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[7] Specification of Timing Extensions for Classic Platform
AUTOSAR_CP_TPS_TimingExtensions

[8] ASAM Fibex – Field Bus Exchange Format, Version 3.1
http://www.asam.net

[9] ISO 17987:2016 (all parts), Road vehicles – Local Interconnect Network (LIN)
https://www.iso.org

[10] CAN specifications
http://www.can-cia.org

[11] MOST Specification, Version 2.5
http://www.mostnet.de

[12] FlexRay Protocol Specification
http://www.flexray.com

[13] SAE-J1708 – Serial Data Communications between Microcomputer Systems in heavy-duty Vehicle Applications

[14] OPEN Sleep/Wake-up Specification for Automotive Ethernet
http://www.opensig.org/Automotive-Ethernet-Specifications/

[15] IEEE 802.1Q-2022 - IEEE Standard for Local and Metropolitan Area Network - Bridges and Bridged Networks
https://ieeexplore.ieee.org/

[16] Specification of Ethernet Switch Driver
AUTOSAR_CP_SWS_EthernetSwitchDriver

[17] Specification of COM Based Transformer
AUTOSAR_CP_SWS_COMBasedTransformer

[18] SOME/IP Protocol Specification
AUTOSAR_FO_PRS_SOMEIPProtocol

[19] Specification of DDS Service Communication Protocol
AUTOSAR_FO_PRS_DDSCommunicationProtocol

[20] Specification of Data Distribution Service Transformer
AUTOSAR_CP_SWS_DataDistributionServiceTransformer

[21] Specification of Data Distribution Service for Classic Platform
AUTOSAR_CP_SWS_DataDistributionService

[22] Basic Software Module Description Template
AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate

[23] Specification of RTE Software
AUTOSAR_CP_SWS_RTE

[24] ASAM MCD-2 MC (ASAP2 / A2L)
http://www.asam.net
ASAM_AE_MCD-2_MC_BS_V1-7-1.pdf

[25] Specification of Communication
AUTOSAR_CP_SWS_COM

[26] Specification of I-PDU Multiplexer
AUTOSAR_CP_SWS_IPDUMultiplexer

[27] SAE J1939-21 Data Link Layer

[28] Procedures for the Management of the Service Name and Transport Protocol Port Number Registry
https://www.rfc-editor.org/rfc/rfc6335.txt

[29] SOME/IP Service Discovery Protocol Specification
AUTOSAR_FO_PRS_SOMEIPServiceDiscoveryProtocol

[30] Specification of Service Discovery
AUTOSAR_CP_SWS_ServiceDiscovery

[31] DDS Interoperability Wire Protocol, Version 2.2
http://www.omg.org/spec/DDSI-RTPS/2.2

[32] Data Distribution Service (DDS), Version 1.4
http://www.omg.org/spec/DDS/1.4

[33] Specification of DDS Service Discovery Protocol
AUTOSAR_FO_PRS_DDSServiceDiscoveryProtocol

[34] Specification of Diagnostic over IP
AUTOSAR_CP_SWS_DiagnosticOverIP

[35] Transport Layer Security (TLS) Parameters
https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

[36] Record Size Limit Extension for TLS
https://tools.ietf.org/html/rfc8449

[37] Specification of Cryptography
AUTOSAR_AP_SWS_Cryptography

[38] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

[39] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[40] ISO 15765-2 – Road vehicles – Diagnostics on Controller Area Networks (CAN) – Part2: Network layer services

[41] IEEE Standard 1722-2016 - IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks

[42] Specification of Socket Adaptor
AUTOSAR_CP_SWS_SocketAdaptor

[43] Specification of Communication Manager
AUTOSAR_CP_SWS_COMManager

[44] General Specification of Transformers
AUTOSAR_CP_SWS_TransformerGeneral

[45] Specification of Secure Onboard Communication
AUTOSAR_CP_SWS_SecureOnboardCommunication

[46] Specification of TCP/IP Stack
AUTOSAR_CP_SWS_TcpIp

[47] Specification of Module E2E Transformer
AUTOSAR_CP_SWS_E2ETransformer

[48] Specification of SW-C End-to-End Communication Protection Library
AUTOSAR_CP_SWS_E2ELibrary

[49] Specification of CRC Library
AUTOSAR_CP_SWS_CRCLibrary

[50] E2E Protocol Specification
AUTOSAR_FO_PRS_E2EProtocol

[51] Specification of Synchronized Time-Base Manager
AUTOSAR_CP_SWS_SynchronizedTimeBaseManager

[52] Specification of Time Synchronization over Ethernet
AUTOSAR_CP_SWS_TimeSyncOverEthernet

[53] Explanation of Time Sensitive Network features
AUTOSAR_FO_EXP_TimeSensitiveNetworkFeatures

[54] Glossary
AUTOSAR_FO_TR_Glossary

[55] Software Process Engineering Meta-Model Specification
http://www.omg.org/spec/SPEM/2.0/

1 Introduction

1.1 Methodology for Defining Formal Template

Figure 1.1 illustrates the overall methodology used to define formal templates. As is explained in the "Generic Structure Template" [1], it is important to separate a precise and concise model of the information that needs to be captured from the concrete XML-DTDs, XML-Schemas or other technology that is used to define the actual templates.

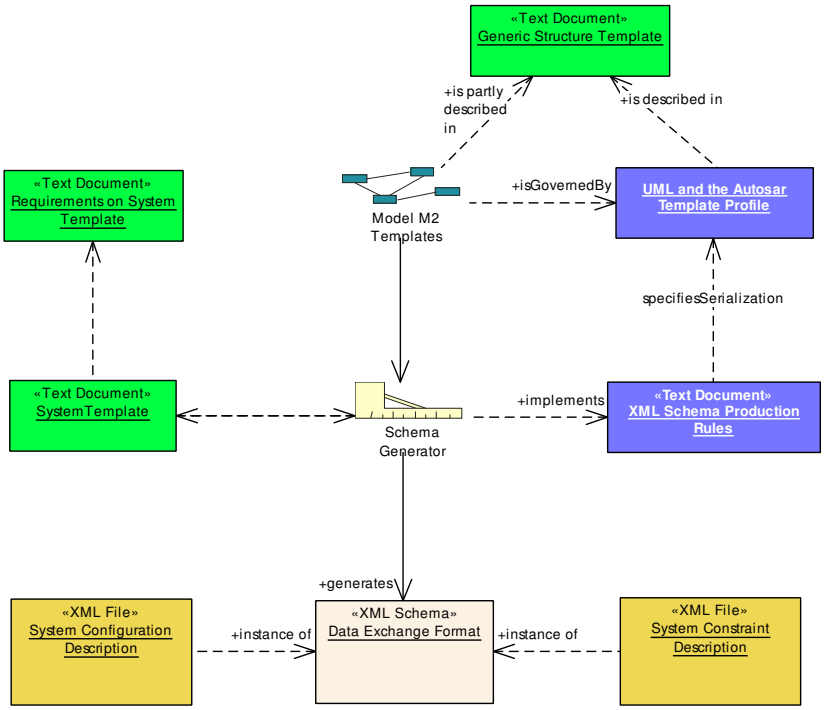

Figure 1.1: Methodology to define templates in AUTOSAR

The following documents describe the various aspects of the methodology:

1. The document called System Template (this document) describes the information that can be captured in the "system constraint" and "system configuration" description, independently from the mapping of this model on XML-technology. This document is based upon the AUTOSAR meta-model and contains an elaborate description of the semantics (the precise meaning) of all the information that can be captured within the relevant parts of this meta-model.

2. The UML and the AUTOSAR Template Profile [1] describes the basic concepts that should be used when creating content of the meta-model.

3. The document called "XML Schema Production Rules" [2] describes how XML is used and how the meta-model designed in the "System Template" should be translated by the "Schema Generator" (MMT) into XML-Schema (XSD) "Data Exchange Format". This "formalization strategy" is to be used for all data that is formally described in the meta-model. In particular this document is

worth to read in order to understand the mapping of the meta-model and the XML based System template.

4. The "Generic Structure Template" [1] describes the top level structure which is common to all AUTOSAR templates and provides AUTOSAR standard mechanisms of modeling elements and patterns.

5. The concrete "Template", the "Data Exchange Format" is an XML schema which is generated out of the meta-model described in the "System Template" using the approach and the patterns defined in the "XML Schema Production Rules". This schema is typically used as input to tools. The M1-level system descriptions are XML files which can be validated against the schema. In that sense they are instances of the schema defining the XML representation of the template.

1.2 Scope

This document describes the system template and its use for the System Constraint Description and the System Configuration Description. In general a filled system template defines the relationship between the pure Software View on the System (represented by a top level SW Component Composition) and a Physical System Architecture with networked ECU instances. The system template is used in two stages of the "AUTOSAR Methodology" [3] (see Figure 1.2).

- As System Constraint Description it serves as input to the AUTOSAR system generator
- As System Configuration Description it defines the output of the AUTOSAR System Configuration Generator and serves as input to the AUTOSAR ECU Configuration Generator for the different ECUs defined in the description.
- As ECU Extract of the System Configuration Description it describes the ECU specific view on the System Description. It is individually generated for each of the System's ECU as the output of the AUTOSAR ECU Configuration Generator.

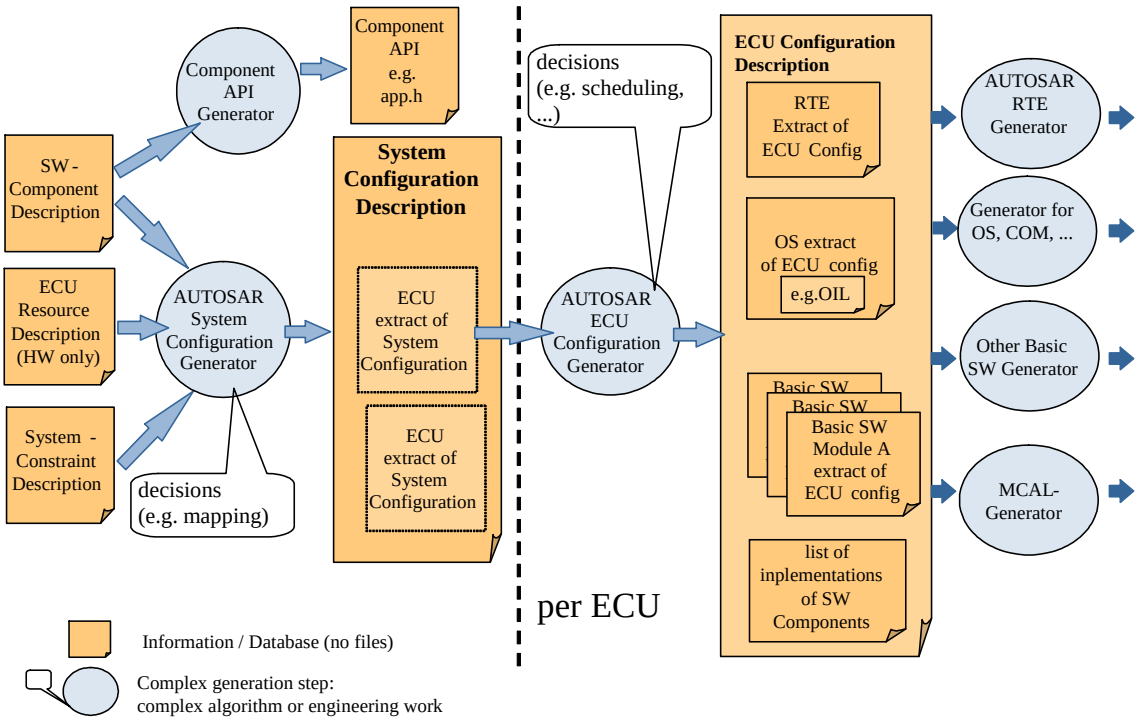

Figure 1.2: AUTOSAR Methodology

The System Template defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints, which will be defined in detail in the following chapters. Figure 1.3 gives an overview how these are used in the two different descriptions.

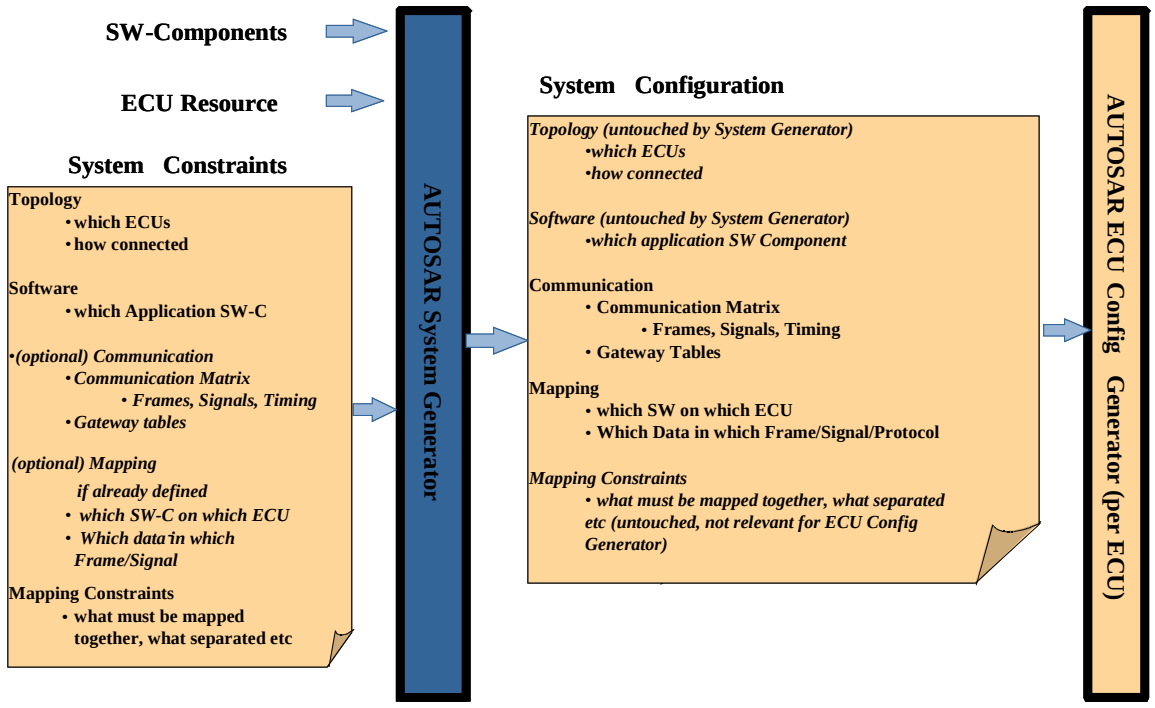

Figure 1.3: Scope of System Constraint Description and System Configuration Description

On Figure 1.3 some of the elements are marked `optional` for the System Constraint Description. If one starts with a new AUTOSAR project, these elements may not be present in the System Constraint Description. No (at least partial) functionality has been mapped yet, thus the communication matrix is not populated. But in most cases, many functional mappings are already predefined and contribute to the population of the communication matrix with their associated signals, thus being present in the System Constraint Description.

Reasons for such a predefinition are manifold. In some cases, hardware setup dictates where certain functionality resides, in some cases, a partial or complete communication matrix and/or completely configured ECUs (HW and SW) of another system (vehicle) has to be taken over. This approach is eased by the fact that System Configuration and System Constraint Description use the same format. That way it is possible to reuse parts of a System Configuration Description of the other system/vehicle in the actual System Constraint Description.

Furthermore, in the figure some of the elements are marked `untouched` for the System Configuration Description. This can have two reasons:

- The System Generator does not modify neither the Topology (networked ECUs) nor the Software, so these parts are just moved from System Constraint Description to System Configuration Description during the generation step.
- In a completed System Configuration Description, all SW components and all ECU-to-ECU communication have been mapped. Thus mapping constraints that limit the flexibility in the mapping phase of the system generator are obsolete

and will not be used in subsequent generator steps. They may however still be present for documentation and validation reasons.

Even if the communication matrix is determined as the result of the system configuration, the ECUs still have to be configured. This is done by the ECU configuration generator, which takes the System Configuration description as input and generates the ECU configuration description. The following guiding principles have been used to determine which information shall be part of the System Configuration Description and which goes into the ECU Configuration Description:

- Information that is common for several ECUs and has to be agreed, shall be part of the System Configuration Description and is thus covered by the System Template.
- Information, that only has ECU-local relevance is part of the ECU Configuration Description.

Thus the ECU Configuration Description will include the OS-schedule, the RTE-configuration and last but not least the configuration of the ECU basic software including the concrete communication drivers on that ECU.

1.3 UML Meta-Model

This chapter gives an overview of the AUTOSAR Unified Modeling Language (UML) meta-model. All AUTOSAR templates use a common meta-model. The templates describe software components, ECU resources, the Basic Software Modules, the ECU Configuration Parameters (ECU Configuration Description and ECU Configuration Parameter Definition) and the System.

The System Template defines all elements, their parameters and their relations, which are necessary for the System Constraint Description and the System Configuration Description.

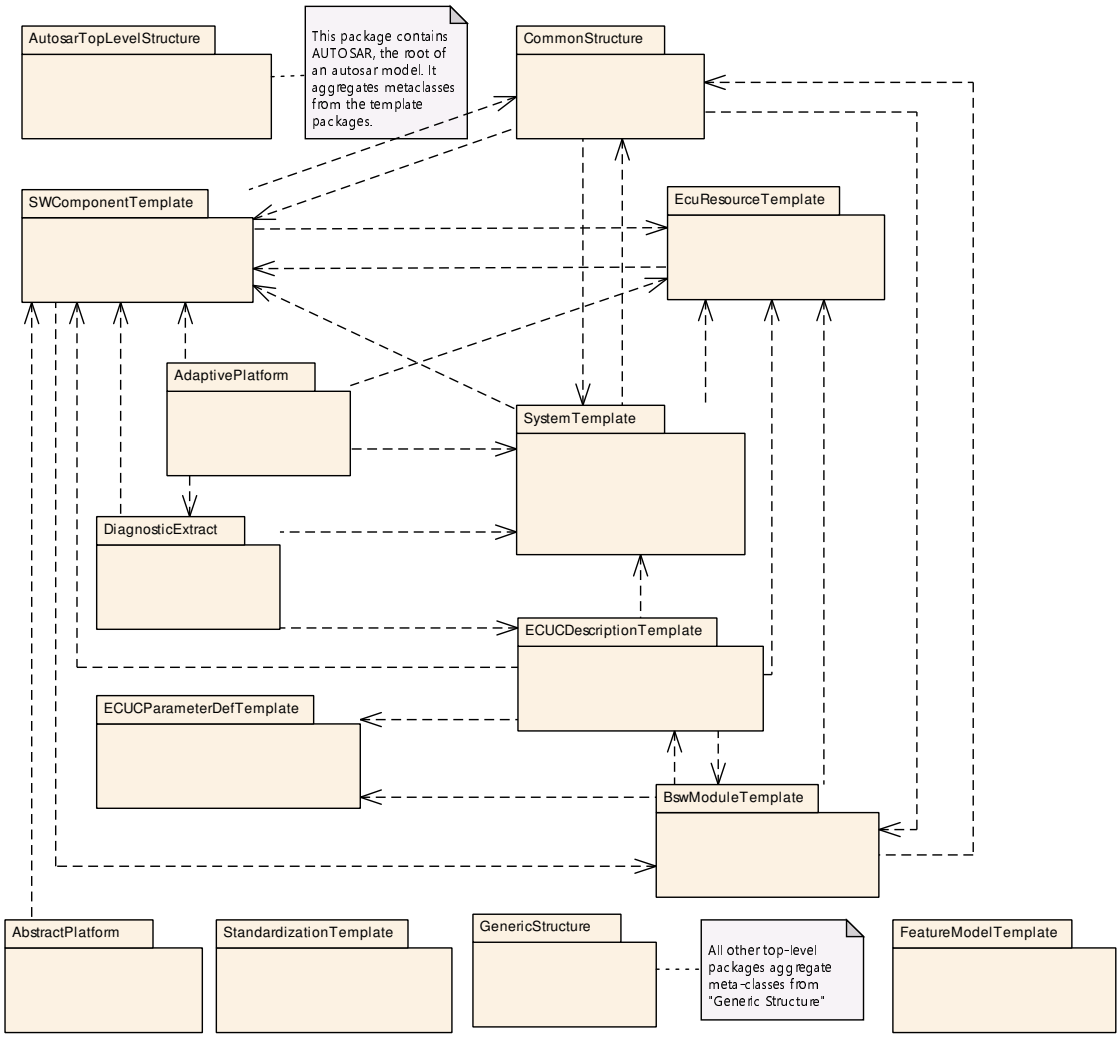

Figure 1.4: AUTOSAR Package Overview

Figure 1.4 shows the overall structure of the meta-model.

The dashed arrows in the diagram describe dependencies in terms of import-relationships between the packages within the meta-model. For example, the package `SystemTemplate` imports meta-classes defined in the packages `GenericStructure` [1], `SWComponentTemplate` [4] and `ECUResourceTemplate` [5].

For clarification, please note that the package `GenericStructure` contains some fundamental infrastructure meta-classes and common patterns that are described in [1]. As these are used by all other template specification the dependency associations are not depicted in the diagram for the sake of clarity.

Generic Structure provides details about

- AUTOSAR Top level structure,
- Commonly used meta-classes and primitives
- Variant Handling

- Documentation

The ECU Resource Template deals with the description of the hardware resources of an ECU. The collection of all ECUs, which are integrated in the car, are described in the topology part of the System Configuration Description/System Constraint Description. Each of these ECUInstances uses the ECU Resource Template to describe the hardware resources. That's the reason, why the topology part has references to the ECU Resource Description.

The SW component description describes the SW components as well as their communication by data elements. The top-level software composition (RootSwCompositionPrototype) is part of the System Template (Software). This top-level software composition contains the functionality of the full system and describes the complete application software architecture of this system. The definition of the top level software composition uses the elements defined in the SW Component Template, like e.g. SwComponentType, PortInterface, AssemblySwConnector and DelegationSwConnector. That's why the System Description has references to the Software Component Description. The top level software composition is described in more detail in chapter 4.

Every template starts with an element AUTOSAR. While the models created in accordance to this guide are independent of the used formalization, it may still help the reader's understanding to note that AUTOSAR would also typically be the root element of a XML Schema generated from such a model. AUTOSAR can then contain one or more nested packages, simply allowing to further structure the contents of the M1 model.

1.4 Document Conventions

Technical terms are typeset in mono spaced font, e.g. PortPrototype. As a general rule, plural forms of technical terms are created by adding "s" to the singular form, e.g. PortPrototypes. By this means the document resembles terminology used in the AUTOSAR XML Schema.

This document contains constraints in textual form that are distinguished from the rest of the text by a unique numerical constraint ID, a headline, and the actual constraint text starting after the ⌈ character and terminated by the ⌋ character.

The purpose of these constraints is to literally constrain the interpretation of the AUTOSAR meta-model such that it is possible to detect violations of the standardized behavior implemented in an instance of the meta-model (i.e. on M1 level).

Makers of AUTOSAR tools are encouraged to add the numerical ID of a constraint that corresponds to an M1 modeling issue as part of the diagnostic message issued by the tool.

The attributes of the classes introduced in this document are listed in form of class tables. They have the form shown in the example of the top-level element AUTOSAR:

Please note that constraints are not supposed to be enforceable at any given time in an AUTOSAR workflow. During the development of a model, constraints may legitimately be violated because an incomplete model will obviously show inconsistencies.

However, at specific points in the workflow, constraints shall be enforced as a safeguard against misconfiguration.

The points in the workflow where constraints shall be enforced, sometimes also known as the "imposition time" of the constraint, are different for each model category, e.g. on the classic platform, the constraints defined for software-components are typically enforced prior to the generation of the RTE while the constraints against the definition of an Ecu extract shall be applied when the Ecu configuration for the Com stack is created.

For each document, possible imposition times of constraints are defined and the imposition times are typically mentioned in the constraint themselves to give a proper orientation for implementers of AUTOSAR authoring tools.

Let `AUTOSAR` be an example of a typical class table. The first rows in the table have the following meaning:

Class: The name of the class as defined in the UML model.

Package: The UML package the class is defined in. This is only listed to help locating the class in the overall meta model.

Note: The comment the modeler gave for the class (class note). Stereotypes and UML tags of the class are also denoted here.

Base Classes: If applicable, the list of direct base classes.

The headers in the table have the following meaning:

Attribute: The name of an attribute of the class. Note that AUTOSAR does not distinguish between class attributes and owned association ends.

Type: The type of an attribute of the class.

Mul.: The assigned multiplicity of the attribute, i.e. how many instances of the given data type are associated with the attribute.

Kind: Specifies, whether the attribute is aggregated in the class (`aggr` aggregation), an UML attribute in the class (`attr` primitive attribute), or just referenced by it (`ref` reference). Instance references are also indicated (`iref` instance reference) in this field.

Note: The comment the modeler gave for the class attribute (role note). Stereotypes and UML tags of the class are also denoted here.

Please note that the chapters that start with a letter instead of a numerical value represent the appendix of the document. The purpose of the appendix is to support the explanation of certain aspects of the document and does not represent binding conventions of the standard.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see [6, Standardization Template].

Please note: By intent, TPS documents (and their traceable items) do not trace up to an RS (requirement item).

1.4.1 Detailed Representation of InstanceRef Associations

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [1]. Each "instanceRef" association can both be represented by the short form and by an detailed representation. For readability the diagrams in the main body of the specification use the short form. The detailed descriptions can be found in the Appendix B.

1.4.2 Variant Handling

The System Template supports the creation of Variants in many of its model elements. In the Metamodel all locations that may exhibit variability are marked with the stereotype `atpVariation`. This allows the definition of possible variation points. Tagged Values are used to specify additional informations.

There are four types of locations in the metamodel which may exhibit variability:

- Aggregations
- Associations
- Attribute Values
- Classes providing property sets

The reasons for the attachment of the stereotype `atpVariation` to certain model elements and the consequences for other model elements are explained in class tables in the following chapters. More details about the AUTOSAR Variant Handling Concept can be found in the AUTOSAR Generic Structure Template [1].

1.4.3 Timing Extensions

With AUTOSAR Release 4.0 a new set of concepts for the description and analysis of end-to-end timing constraints is introduced by the Specification of Timing Extensions. A subset of these extensions aims for the system level and can be used to enhance the descriptions that are already available in the System Template.

A dedicated description of the timing extensions that can be used at system level is given in chapter 3 (System timing) in the Specification of Timing Extensions [7].

1.4.4 Documentation Support

With AUTOSAR Release 4.0 the AUTOSAR XML schema provides support for integrated and well structured documentation. More details about the AUTOSAR Documentation Support concept can be found in the AUTOSAR Generic Structure Template [1]. An optional documentation block can be applied to any identifiable element. Furthermore, as shown in figure 1.5, the System Template provides the possibility of adding additional documentation to several non-identifiable elements. The documentation of a `System` is composed of several chapters.

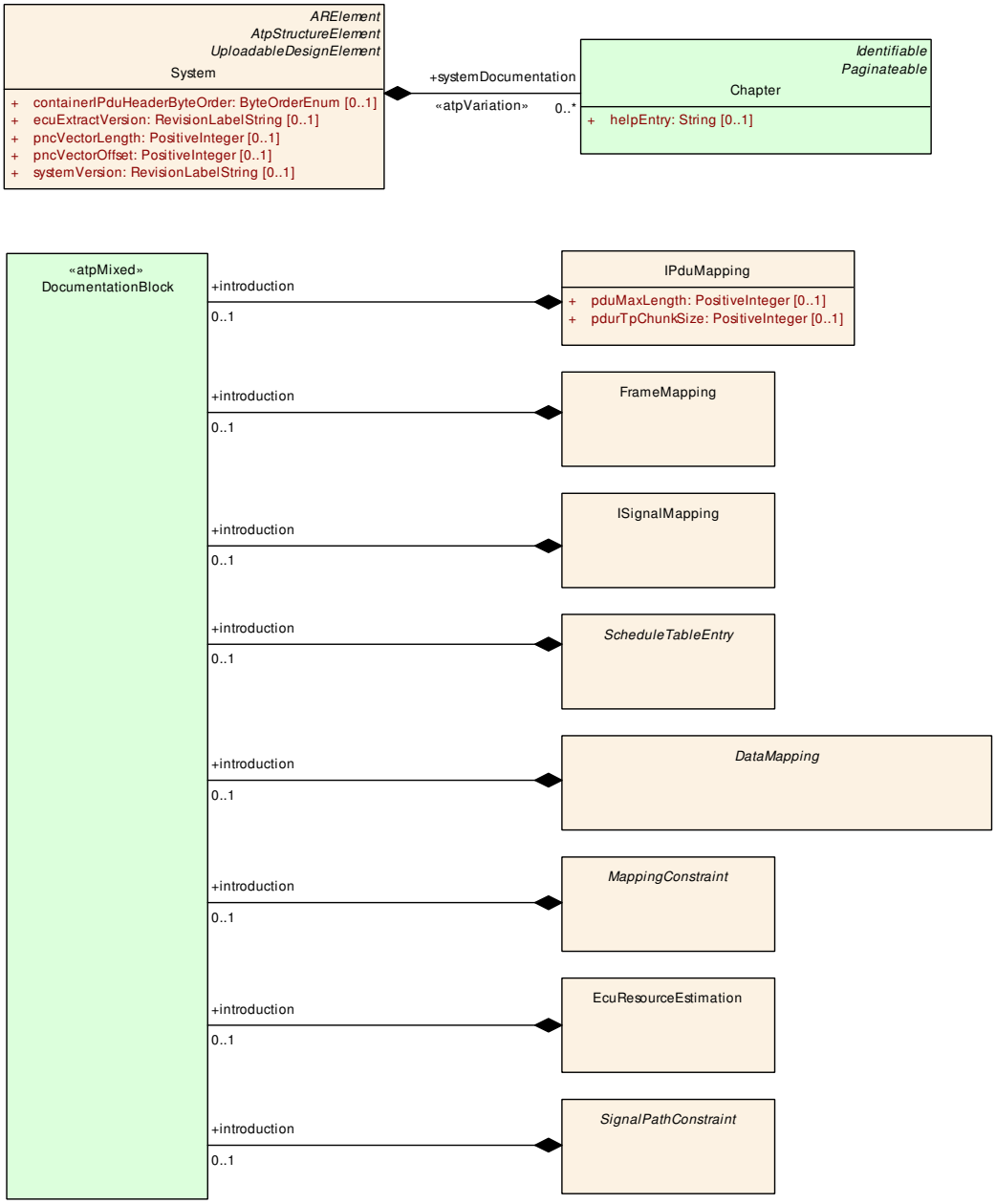

Figure 1.5: System Template Documentation Support

1.4.5 Stereotype atpSplitable in the System Template

The stereotype ≪atpSplitable≫ is used in the System Template to support stepwise processes, where the System Configuration Description is completed incrementally over a development process. Example:

1) Description of Communication only consists of interaction signals (ISignal). This is enough information to create an individual ECU's RTE, and even contains enough information to configure an ECU where the actual Frame/Pdu communication is being handled post-build.

2) In a second step, the communication matrix is being completed for a concrete vehicle. Pdus and Frames, along with their Triggerings are being added to the previous System Description. This model then contains the full information about an ECU's communication, especially containing the additional information to generate the post build information.

So, in this 2-step approach, an OEM could deliver the incomplete ECU extract from step (1) to the ECU integrator, who can then build a complete software image for the ECU. In the 2nd step, the ECU extract will be completed by the previously missing information, but as the first extract will still be valid due to the ≪atpSplitable≫ construct, the ECU including the flashed image from step (1) can be (re)used as it is, and just will be completed with the post build information, e.g. Frames and Pdus.

Further details about the ≪atpSplitable≫ stereotype can be found in the Generic Structure Template [1].

1.5 AUTOSAR System Template and ASAM FIBEX

FIBEX (Field Bus Exchange Format) [8] is an XML exchange format proposed for data exchange between tools that deal with bus communication Systems. The format supports the most common automotive data buses: LIN [9], CAN [10], MOST [11], FlexRay [12]. The covered areas of the exchange format are the functional network, system topology and the communication level. The functional network describes the software architecture of the system. In the system topology the logical layout of the system is described. This means it is documented which ECU is connected to which bus. The central purpose of a communication system is the exchange of frames with certain properties. The format is able to describe frames and their timing properties.

In future versions of the System Template a common subset between ASAM FIBEX and AUTOSAR will be harmonized. The current version of the System Template contains already the ASAM FIBEX description for communication and topology. Due to requirements of AUTOSAR some extensions were made to those descriptions. For instance the communication part is extended by a concept for PDUs (I-Pdus and N-Pdus). The harmonization between ASAM FIBEX and AUTOSAR System Template is not finalized at this time.

In the UML Meta-Model the FIBEX contents are located in an own FIBEX UML Package. The top level `FibexElement` is referenced by the top level element `System` of the System Template. Similar to the usage of the `ARElement`, specializations of the `FibexElement` represent elementary building blocks within the FIBEX package. Each of this elements will be described in more detail in the following chapters.

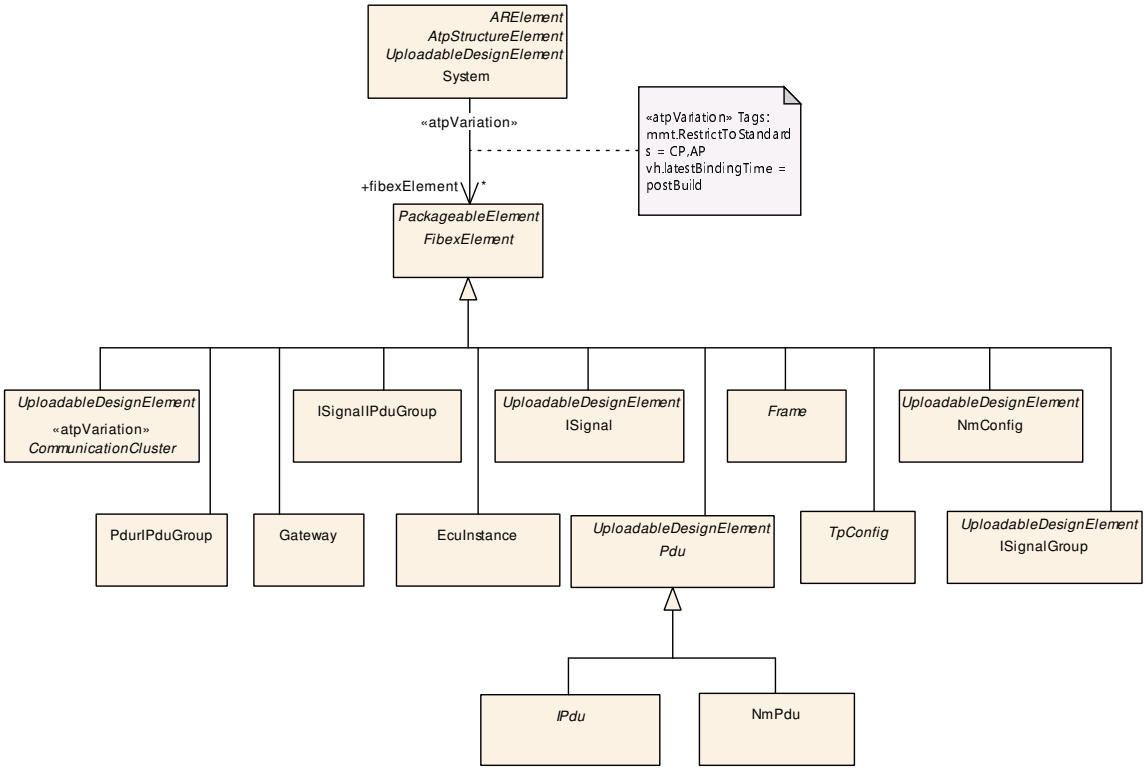

Figure 1.6: Fibex Elements

2 System

The top level element of the System Template is the class `System`, as shown in figure 2.1.

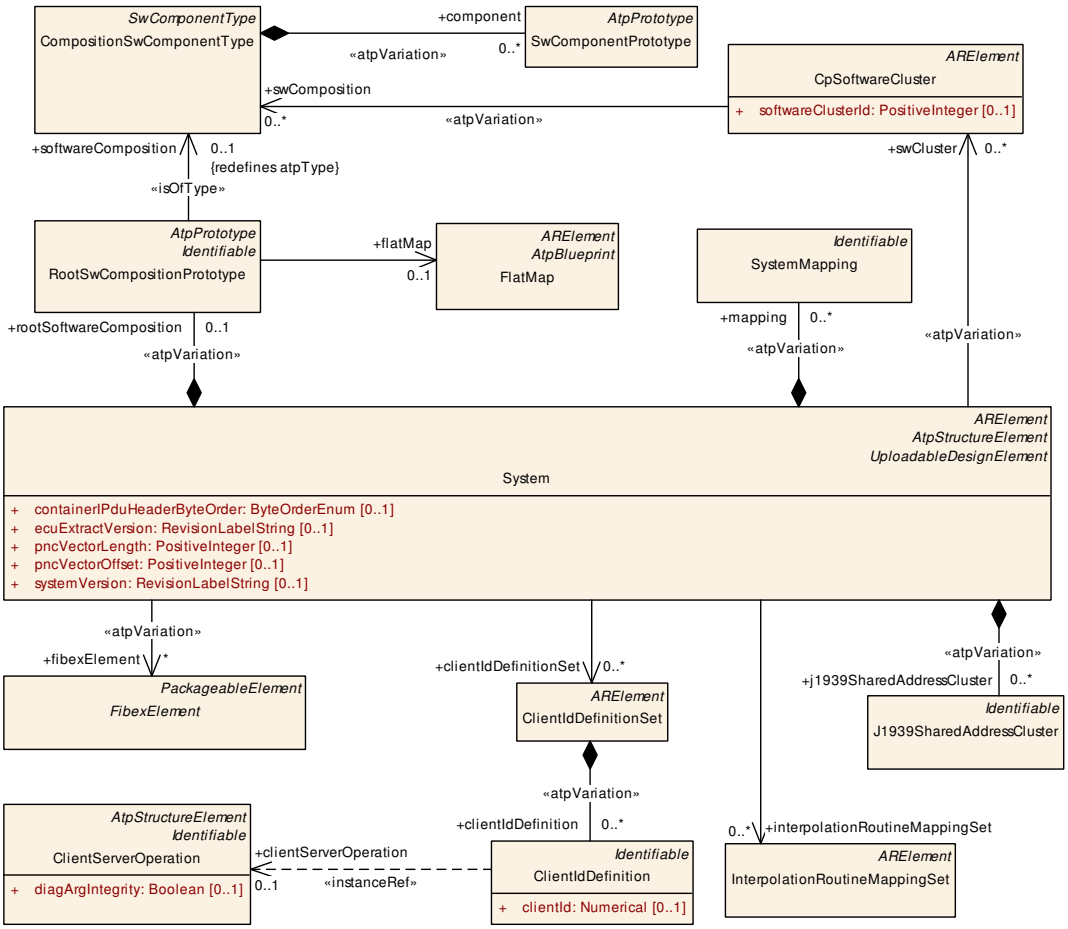

Figure 2.1: System Template Overview

Class	**System**			
Note	The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints. The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology. **Tags:** atp.recommendedPackage=Systems			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	System			
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. **Tags:** atp.Splitkey=clientIdDefinitionSet This Attribute is only used by the AUTOSAR Classic Platform.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus. This Attribute is only used by the AUTOSAR Classic Platform.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract. This Attribute is only used by the AUTOSAR Classic Platform.
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology. All Fibex Elements used within a System Description shall be referenced from the System Element. atpVariation: In order to describe a product-line, all Fibex Elements can be optional. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=fibexElement, fibexElement.fibexElement, fibexElement.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolation Routine MappingSet	InterpolationRoutine MappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System. **Tags:** atp.Splitkey=interpolationRoutineMappingSet This Attribute is only used by the AUTOSAR Classic Platform.
j1939Shared AddressCluster	J1939SharedAddress Cluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=j1939SharedAddressCluster, j1939Shared AddressCluster.shortName, j1939SharedAddress Cluster.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
mapping	SystemMapping	*	aggr	Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints). In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping. This element is not required when the System description is used for a network-only use-case. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mapping, mapping.shortName, mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	System			
pncVector Length	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftware Composition	RootSwComposition Prototype	0..1	aggr	Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case. atpVariation: The RootSwCompositionPrototype can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rootSoftwareComposition, rootSoftware Composition.shortName, rootSoftware Composition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
swCluster	CpSoftwareCluster	*	ref	CP Software Clusters of this System **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swCluster, swCluster.cpSoftwareCluster, sw Cluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
systemCom SpecDefinition	SystemComSpec DefinitionSet	*	ref	Reference to the set of ComSpec definitions that are used for inter-ECU communication in the System. **Tags:** atp.Splitkey=systemComSpecDefinition
system Documentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemDocumentation, system Documentation.shortName, system Documentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10 This Attribute is only used by the AUTOSAR Classic Platform.
systemVersion	RevisionLabelString	0..1	attr	Version number of the System Description.

Table 2.1: System

`System` has relationships to all elements that define a system constraint description or system configuration description. It aggregates the `SystemMapping` and `RootSwCompositionPrototype` elements. `SystemMapping` deals with mapping of software components to ECUs as well as with the mapping of data elements that are to be exchanged between software components onto signals and frames. The `RootSwCompositionPrototype` element contains a reference to the top level software composition.

[TPS_SYST_02364] Scope of the `System` ⌈The `System` defines a vehicle representation and describes the software related parts of the vehicle. This includes the

Software Components that are deployed to ECUs of the vehicle but also the means for the Software to communicate with each other, such as network topology and communication matrix.⌋

[constr_3028] `FibexElements`

Imposition time: CP: `IT_SysDesc`

⌈Each `FibexElement` that is used in the System Description shall be referenced by the `System` element in the role `FibexElement`.⌋

`FibexElement`s can be defined in a stand alone and reusable way (hence they can simply be created in any package like ARElements), but on the other hand it shall be clear that a certain `FibexElement` actually belongs to a certain System Description. Thus, all `FibexElement`s used within a System Description (i.e. contributing to the specification of the System communication and topology) shall be referenced from the `System` element. More details about the integration of FIBEX into the System Template will be given in chapter 1.5.

[TPS_SYST_01002] `System Category` ⌈The `System` shall have a `category` element defined which indicates the role of this work product.⌋

[TPS_SYST_01003] `Standardized System Category Definitions` ⌈

category	*Meaning*
SYSTEM_CONSTRAINTS	The `System` class is used to describe System Constraints. In this usage, it forms the core element of a System Constraints Description, serving as an input to the AUTOSAR System Generator.
SYSTEM_DESCRIPTION	The `System` class is used to describe the System Configuration of a complete AUTOSAR System. In this usage, it forms the core element of a System Description, the output of the AUTOSAR System Generator.
SYSTEM_EXTRACT	The `System` class is used to describe a subsystem specific view on the complete System Description. The System Extract is not fully decomposed and still contains compositions. The SYSTEM_EXTRACT is the basis for designing subsystems.
ECU_EXTRACT	The `System` class is used to describe the ECU specific view on the complete System Description. In this usage, it forms the core element of ECU Extract, the output of the AUTOSAR ECU Configuration Extractor. The ECU Extract is fully decomposed and contains only atomic software components. The ECU Extract is the basis for setting up the ECU Configuration.
ABSTRACT_SYSTEM_DESCRIPTION	The `System` is used to describe a functional (solution-independent/abstract) system design. It can be taken as basis for the development of the SYSTEM_DESCRIPTION. No structural constraints are applied on the transformation of the ABSTRACT_SYSTEM_DESCRIPTION to the SYSTEM_DESCRIPTION.
ECU_SYSTEM_DESCRIPTION	The System is used to describe the closed view on one ECU (note that an AUTOSAR ECU is defined being one microprocessor running one AUTOSAR Stack). It can be derived from a SYSTEM_EXTRACT or it can be designed independently and mapped to a SYSTEM_EXTRACT. The ECU_SYSTEM_DESCRIPTION is not fully decomposed and still may contain compositions.
SW_CLUSTER_SYSTEM_DESCRIPTION	`System` that describes the content of a single `CpSoftwareCluster`.

▽

△

category	*Meaning*
RPT_SYSTEM	System which describes the rapid prototyping algorithm in the format of AUTOSAR Software Components. For more details see the Software Component Template [4] and TR_Methodology [3].

⌋

Note: SYSTEM_EXTRACT does not prescribe the number of micro controllers / cores for one ECU from the OEM perspective.

- Supplier decides to design one AUTOSAR ECU with multicore support leads to one ECU_EXTRACT supporting one AUTOSAR stack
- Supplier decides to design two AUTOSAR ECUs (i.e., two micro-controllers) in one box leads to two ECU_EXTRACTs supporting two AUTOSAR stacks

[constr_3027] Existence of `ecuExtractVersion`

Imposition time: CP: `IT_EcuExt`

⌈In case the category of the System is SYSTEM_EXTRACT or ECU_EXTRACT the `ecuExtractVersion` attribute shall be defined.⌋

2.1 Data interpretation of bus content in different contexts

A System Description can be used for different purposes:

- as input for the creation of the Base Ecu Configuration
- as input for bus monitoring and network analysis tools

In some cases the System Description is interpreted in a different way between Ecuc tools and network analysis tools. This chapter collects such use cases and describes the differences in the interpretation of the configuration data.

2.1.1 Nm NID/CBV

[TPS_SYST_02366] NID/CBV signals shall be ignored by Ecuc tools ⌈If NID/CBV are enabled (`nmCbvPosition` and `nmNidPosition` are configured), the Signals that are defined at the position of the respective NID/CBV bytes shall be ignored by Base Ecuc generators since these signals are not processed by COM.⌋

Please note that there may be the use case to define the Signals that represent the NID and CBV in the `NmPdu` for bus monitoring or network analysis purposes.

2.2 ClientIdDefinitionSet

In the ClientIdDefinitionSet all Client Identifiers of the transaction handle used for a inter-ECU client server communication can be defined that belong to the System that refers the ClientIdDefinitionSet.

Class	ClientIdDefinitionSet			
Note	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. **Tags:** atp.recommendedPackage=ClientIdDefinitionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientId Definition	ClientIdDefinition	*	aggr	Definition of a Client Identifier that will be used by the RTE in a inter-ECU client-server communication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clientIdDefinition, clientIdDefinition.shortName, clientIdDefinition.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 2.2: ClientIdDefinitionSet

Class	ClientIdDefinition			
Note	Several clients in one client-ECU can communicate via inter-ECU client-server communication with a server on a different ECU, if a client identifier is used to distinguish the different clients. The Client Identifier of the transaction handle that is used by the RTE can be defined by this element.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ClientIdDefinitionSet.clientIdDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientId	Numerical	0..1	attr	The Client Identifier of the transaction handle used for an inter-ECU client server communication is defined by this attribute. If defined the RTE generator shall use this client Id.
clientServer Operation	ClientServerOperation	0..1	iref	Reference to the ClientServerOperation that is called by the client. **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef

Table 2.3: ClientIdDefinition

[constr_3117] Allowed value of attribute clientId

Imposition time: CP: IT_SysDesc

⌈Within the context of one ClientIdDefinition, the value of attribute clientId shall be in the range of ClientIdRange.lowerLimit and ClientIdRange.upperLimit for the ClientIdRange that is aggregated by the EcuInstance onto which the SwComponentPrototypes included in the ClientIdDefinition.clientServerOperation are mapped if the EcuInstance aggregates a ClientIdRange.⌋

Please note that the clientId is bound to the ClientServer relationship and does not represent a globally unique identifier of the Client call. ClientIds can be reused in the context of a different ClientServer relationship.

[constr_3118] Valid reference target for ClientIdDefinition.clientServerOperation.contextPort

Imposition time: CP: IT_SysDesc

⌈In the context of the definition of a ClientIdDefinition, the reference clientServerOperation.contextPort shall only refer to an RPortPrototype.⌋

Rationale: the definition of a client ID does only make sense in the context of a client of a ClientServerOperation.

[constr_5393] Existence of clientId

Imposition time: CP: IT_SysDesc

⌈For each ClientIdDefinition, the attribute clientId shall exist.⌋

[constr_5394] Existence of clientServerOperation

Imposition time: CP: IT_SysDesc

⌈For each ClientIdDefinition, the attribute clientServerOperation shall exist.⌋

2.3 System ComSpec Definition

Application software components are designed to be deployable to different system contexts. Thus several communication attributes may be depending on the actual deployment scenario the application software component is used in.

The SystemComSpecDefinitionSet is used to collect the definition of attributes specific to a dedicated System and define attribute values for application software component usage in that System. The reference System.systemComSpecDefinition defines which SystemComSpecDefinitionSets are applicable for this System.

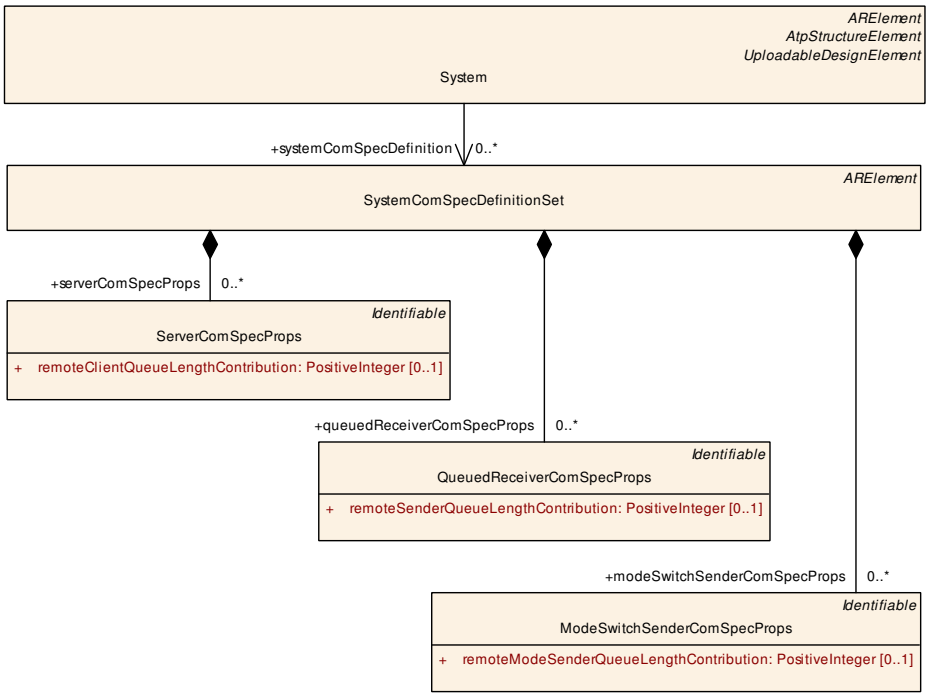

Figure 2.2: ComSpecDefinition

Class	**SystemComSpecDefinitionSet**			
Note	Set of ComSpecDefinitions in the scope of the System. **Tags:** atp.recommendedPackage=SystemComSpecDefinitionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
modeSwitch SenderCom SpecProps	ModeSwitchSender ComSpecProps	*	aggr	Definition of ComSpec attributes for the mode switch sender. **Tags:** atp.Splitkey=modeSwitchSenderComSpecProps, modeSwitchSenderComSpecProps.shortName
queuedReceiver ComSpecProps	QueuedReceiverCom SpecProps	*	aggr	Definition of ComSpec props for the receiverof a data Element. **Tags:** atp.Splitkey=queuedReceiverComSpecProps, queuedReceiverComSpecProps.shortName
serverComSpec Props	ServerComSpecProps	*	aggr	Definition of ComSpec props for the server of a Client ServerOperation **Tags:** atp.Splitkey=serverComSpecProps, serverCom SpecProps.shortName

Table 2.4: SystemComSpecDefinitionSet

2.3.1 Server ComSpec

The meta-class `ServerComSpecProps` is able to define the attribute `remote-ClientQueueLengthContribution` for a specific `ClientServerOperation` instance in the scope of the `System`. The value of `ServerComSpecProps.remote-ClientQueueLengthContribution` is used to define the queue length which the

server of that ClientServerOperation at least needs to configure for the queue length to handle the remote clients in the System.

Additional local clients in the scope of the EcuInstance the server is running on are not covered by the value of remoteClientQueueLengthContribution, these additional local clients need to be considered when configuring the actual queue length during Ecu configuration.

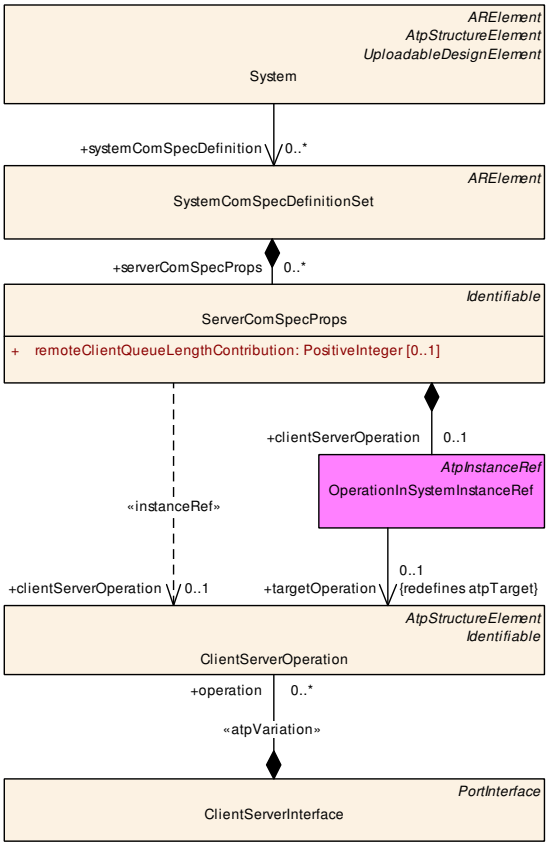

Figure 2.3: ServerComSpecProps

Class	**ServerComSpecProps**			
Note	Definition of ComSpec attributes for the server.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemComSpecDefinitionSet.serverComSpecProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientServer Operation	ClientServerOperation	0..1	iref	Reference to the ClientServerOperation. **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef

△

Class	**ServerComSpecProps**			
remoteClient QueueLength Contribution	PositiveInteger	0..1	attr	Defines the size of queue length contribution of remote clients this ClientServerOperation shall at least be configured at the server. Note that the queue length may be enlarged by local clients of that ClientServerOperation.

Table 2.5: ServerComSpecProps

2.3.2 Queued Receiver ComSpec

The meta-class `QueuedReceiverComSpecProps` is able to define the attribute `remoteSenderQueueLengthContribution` for a specific `dataElement` instance in the scope of the `System`. The value of `QueuedReceiverComSpecProps.remoteSenderQueueLengthContribution` is used to define the queue length which the receiver of that `dataElement` at least needs to configure for the queue length to handle the remote senders in the `System`.

Additional local senders in the scope of the `EcuInstance` the receiver is running on are not covered by the value of `remoteSenderQueueLengthContribution`, these additional local senders need to be considered when configuring the actual queue length during Ecu configuration.

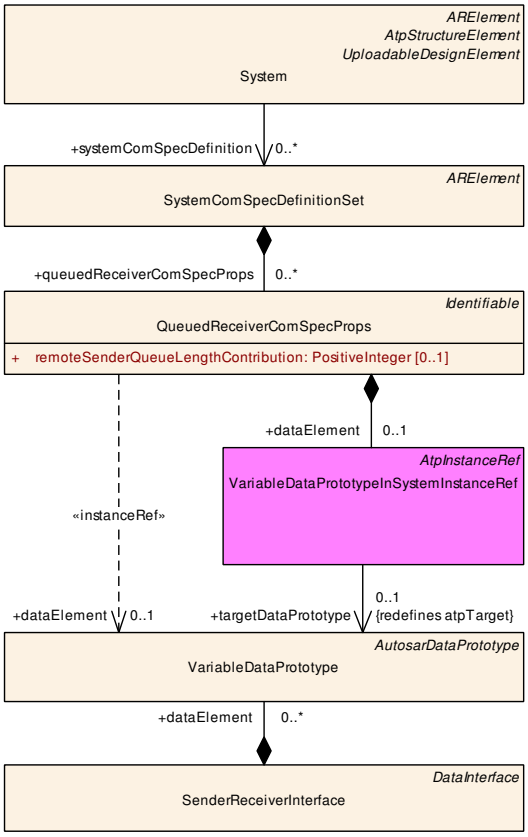

Figure 2.4: QueuedReceiverComSpecProps

Class	**QueuedReceiverComSpecProps**			
Note	Definition of ComSpec attributes for the queued receiver.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemComSpecDefinitionSet.queuedReceiverComSpecProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	0..1	iref	Reference to the dataElement of a SenderReceiver Interface. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
remoteSender QueueLength Contribution	PositiveInteger	0..1	attr	Defines the size of queue length contribution of remote senders this SenderReceiver dataElement shall at least be configured at the receiver. Note that the queue legth may be enlarged by local senders of that dataElement.

Table 2.6: QueuedReceiverComSpecProps

2.3.3 ModeSwitch Sender ComSpec

The meta-class `ModeSwitchSenderComSpecProps` is able to define the attribute `remoteModeSenderQueueLengthContribution` for a specific `modeGroup` instance in the scope of the `System`. The value of `ModeSwitchSenderComSpecProps.remoteModeSenderQueueLengthContribution` is used to define the queue length which the mode user of that `modeGroup` at least needs to configure for the queue length to handle the remote senders in the `System`.

Additional local senders in the scope of the `EcuInstance` the receiver is running on are not covered by the value of `remoteModeSenderQueueLengthContribution`, these additional local senders need to be considered when configuring the actual queue length during Ecu configuration.

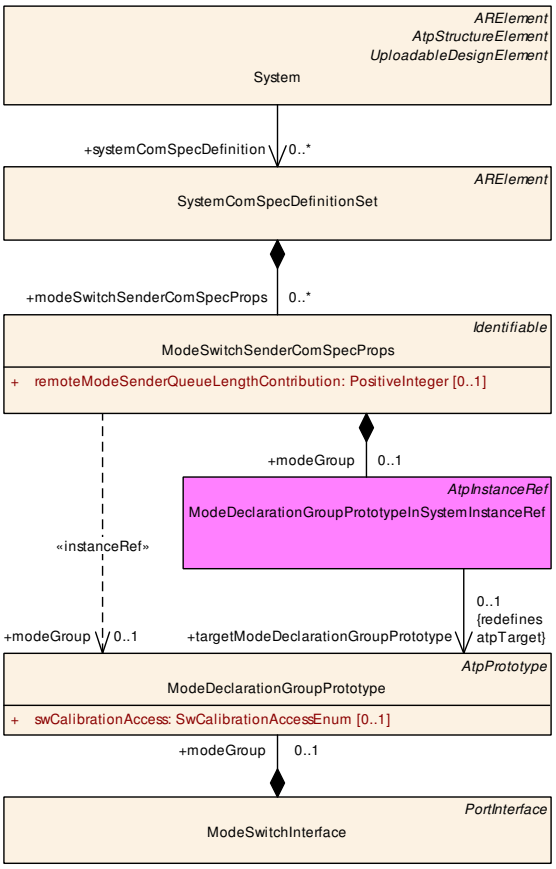

Figure 2.5: ModeSwitchSenderComSpecProps

Class	**ModeSwitchSenderComSpecProps**			
Note	Definition of ComSpec attributes for the mode switch sender.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemComSpecDefinitionSet.modeSwitchSenderComSpecProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	Reference to the ModeDeclarationGroupPrototype. **Tags:** atp.Splitkey=modeGroup, modeGroup.context Composition, modeGroup.contextComponent, mode Group.contextPort, modeGroup.targetModeDeclaration GroupPrototype **InstanceRef implemented by:** ModeDeclarationGroup PrototypeInSystemInstanceRef
remoteMode SenderQueue Length Contribution	PositiveInteger	0..1	attr	Defines the size of queue length contribution of remote mode switch senders this modeGroup shall at least be configured at the sender. Note that the queue length may be enlarged by local receivers of that modeGroup.

Table 2.7: ModeSwitchSenderComSpecProps

Class	ModeDeclarationGroupPrototypeInSystemInstanceRef			
Note				
Base	ARObject, AtpInstanceRef			
Aggregated by	ModeSwitchSenderComSpecProps.modeGroup, PortElementToCommunicationResourceMapping.mode DeclarationGroupPrototype			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	0..1	ref	**Tags:** xml.sequenceOffset=40
targetMode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	ref	**Tags:** xml.sequenceOffset=50

Table 2.8: ModeDeclarationGroupPrototypeInSystemInstanceRef

2.4 InterpolationRoutingMappingSet

The `System` defines with the `interpolationRoutineMappingSet` reference all `InterpolationRoutineMappingSet`s that are relevant in the context of the `System`. More details about the `InterpolationRoutineMappingSet`s, `InterpolationRoutineMapping`s and `InterpolationRoutine`s can be found in the Software Component Template [4].

Class	InterpolationRoutineMappingSet			
Note	This meta-class specifies a set of interpolation routine mappings. **Tags:** atp.recommendedPackage=InterpolationRoutineMappingSets			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
interpolation Routine Mapping	InterpolationRoutine Mapping	*	aggr	This specifies one particular mapping of recordlayout and its matching interpolationRoutines. **Tags:** atp.Splitkey=interpolationRoutineMapping

Table 2.9: InterpolationRoutineMappingSet

Class	InterpolationRoutineMapping
Note	This meta-class provides a mapping between one record layout and its matching interpolation routines. This allows to formally specify the semantics of the interpolation routines. The use case is such that the curves/Maps define an interpolation method. This mapping table specifies which interpolation routine implements methods for a particular record layout. Using this information, the implementer of a software-component can select the appropriate interpolation routine.
Base	ARObject
Aggregated by	InterpolationRoutineMappingSet.interpolationRoutineMapping

▽

△

Class	**InterpolationRoutineMapping**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interpolation Routine	InterpolationRoutine	*	aggr	This is one particular interpolation routine which is mapped to the record layout. **Tags:** atp.Splitkey=interpolationRoutine
swRecord Layout	SwRecordLayout	0..1	ref	This refers to the record layout which is mapped to interpolation routines. **Tags:** atp.Splitkey=swRecordLayout

Table 2.10: InterpolationRoutineMapping

Class	**InterpolationRoutine**			
Note	This represents an interpolation routine taken to evaluate the contents of a curve or map against a specific input value.			
Base	*ARObject*			
Aggregated by	InterpolationRoutineMapping.interpolationRoutine			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interpolation Routine	BswModuleEntry	0..1	ref	This specifies a BswModuleEntry which implements the current interpolation method for the given record layout. **Tags:** atp.Splitkey=interpolationRoutine xml.sequenceOffset=30 This Attribute is only used by the AUTOSAR Classic Platform.
isDefault	Boolean	0..1	attr	This attribute specifies whether the enclosing InterpolationRoutine is considered the default in the context (defined by the System Template) of a given collection InterpolationRoutineMapping that owns the enclosing InterpolationRoutine. **Tags:** xml.sequenceOffset=20
shortLabel	Identifier	0..1	attr	This is the name of the interpolation method which is implemented by the referenced bswModuleEntry. It corresponds to swInterpolationMethod in SwDataDef Props. **Tags:** xml.sequenceOffset=10

Table 2.11: InterpolationRoutine

[constr_5114] Semantics of `InterpolationRoutine.isDefault`

Imposition time: CP: `IT_SysDesc`

⌈For each `SwRecordLayout` that is referenced by one or more `InterpolationRoutineMapping`s that are aggregated by `InterpolationRoutineMappingSet`s that are referenced from a `System` in the role `interpolationRoutineMappingSet`, only one of the collection of aggregated `InterpolationRoutine`s shall have attribute `isDefault` set to True.⌋

3 Topology

This chapter explains how a vehicle's physical System Topology is being modeled in AUTOSAR (Example: Figure 3.1). A topology is formed by a number of EcuInstances that are interconnected to each other in order to form ensembles of ECUs and CommunicationClusters, which are further detailed by providing information on bus-specific properties.

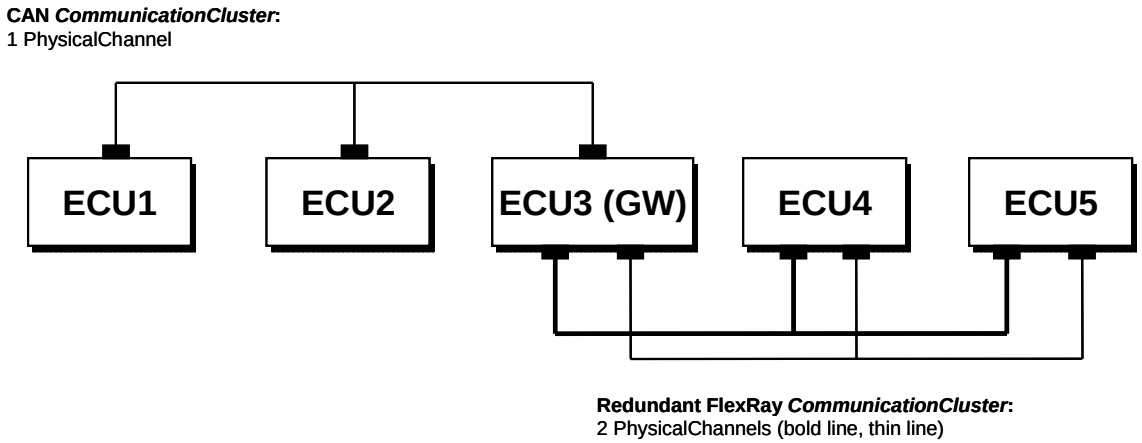

Figure 3.1: Example for a Communication Cluster within a physical network topology

In the AUTOSAR methodology [3] the topology description is one of the inputs for the System Generator. It serves as constraints for mapping the Software Components (see chapter 5.1) contained in the RootSwCompositionPrototype as well as for defining the System Communication matrix (see chapter 6). Gateways which allow the exchange of Signals between CommunicationClusters are covered in chapter 8.

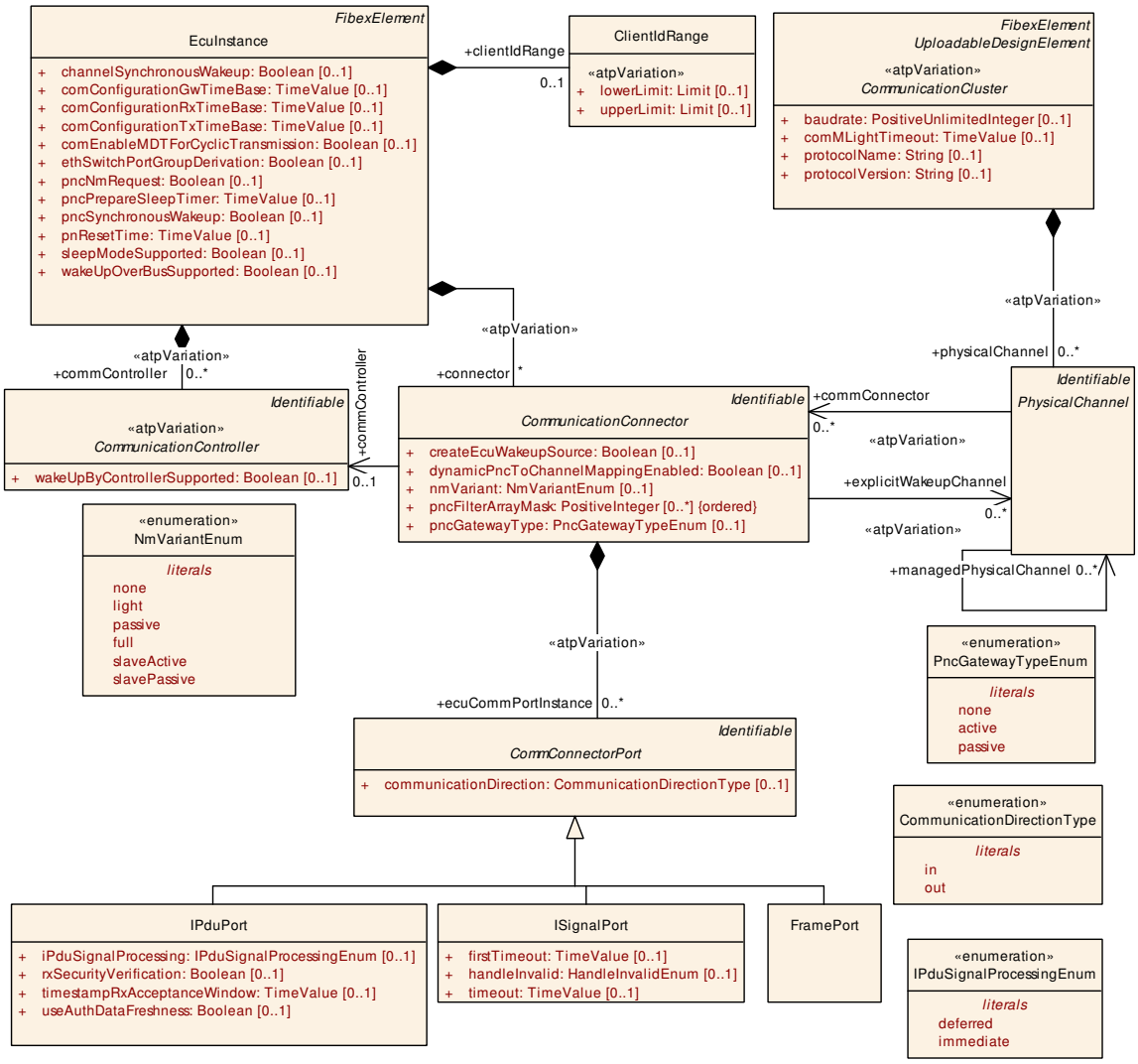

Figure 3.2: Topology elements (Topology)

3.1 ECUs and their communication capabilities

Within a System Topology, the ECUs actually being connected with each other are described in the form of `EcuInstance`s. An `EcuInstance` needs to have one or more `CommunicationController`, the actual hardware device by means of which devices send and receive frames from the communication medium. Furthermore, the `EcuInstance` has one or more `CommunicationConnector`s which describe the bus interfaces of the ECUs and to specify the sending/receiving behavior.

3.1.1 ECU Instance

[TPS_SYST_01005] Definition of `EcuInstance` ⌈The `EcuInstance` defines one instance of the AUTOSAR stack.⌋

The actual description of the ECU hardware resources is done by the means of the ECU Resource Template [5]: It uses the HwElement class and its aggregated hardware elements for defining a specific ECU type. In other words the Ecu Resource Template "Ecu" is used to describe the physical box (HwElement of category Ecu) containing the electronics which may contain several microcontrollers with several AUTOSAR Stack instances running.

[TPS_SYST_01006] Assign ECU type to EcuInstance ⌈The process of assigning an ECU type to EcuInstance is a mapping step (see [TPS_SYST_01019]) and performed latest in the System Generation step.⌋

An EcuInstance can serve as a gateway if it is connected to two or more different clusters by two or more of its CommunicationControllers.

Class	**EcuInstance**			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description. **Tags:** atp.recommendedPackage=EcuInstances			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
associatedComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignal IPduGroups are applicable for which Communication Connector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPdu Groups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup. **Tags:** atp.Splitkey=associatedComIPduGroup This Attribute is only used by the AUTOSAR Classic Platform.
associated Consumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=associatedConsumedProvidedService InstanceGroup, associatedConsumedProvidedService InstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstance Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU. **Tags:** atp.Splitkey=associatedPdurIPduGroup
channel Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.

▽

△

Class	EcuInstance			
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication. **Tags:** atp.Splitkey=clientIdRange
com Configuration GwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationRx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds. This Attribute is only used by the AUTOSAR Classic Platform.
comEnable MDTForCyclic Transmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclic Timing assigned or eventControlledTiming with numberOf Repetitions > 0). This Attribute is only used by the AUTOSAR Classic Platform.
commController	Communication Controller	*	aggr	CommunicationControllers of the ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commController, commController.short Name, commController.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
dltConfig	DltConfig	0..1	aggr	Describes the Dlt configuration on this EcuInstance. **Tags:** atp.Splitkey=dltConfig This Attribute is only used by the AUTOSAR Classic Platform.
doIpConfig	DoIpConfig	0..1	aggr	DoIp configuration on this EcuInstance. **Tags:** atp.Splitkey=doIpConfig atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ecuTaskProxy	OsTaskProxy	*	ref	Reference to OsTaskProxies assigned to the Ecu Instance. **Tags:** atp.Splitkey=ecuTaskProxy This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	EcuInstance			
ethSwitchPort Group Derivation	Boolean	0..1	attr	Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. This Attribute is only used by the AUTOSAR Classic Platform.
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of an EcuInstance. **Tags:** atp.Splitkey=firewallRule
j1939Node	J1939Node	*	aggr	Optional collection of J1939Nodes defined on this Ecu Instance. **Tags:** atp.Splitkey=j1939Node, j1939Node.shortName This Attribute is only used by the AUTOSAR Classic Platform.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu. **Tags:** atp.Splitkey=partition, partition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
pncNmRequest	Boolean	0..1	attr	Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.
sleepMode Supported	Boolean	0..1	attr	Specifies whether the ECU instance may be put to a "low power mode" • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software. This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpIcmpProps	EthTcpIpIcmpProps	0..1	ref	EcuInstance specific ICMP (Internet Control Message Protocol) attributes **Tags:** atp.Splitkey=tcpIpIcmpProps This Attribute is only used by the AUTOSAR Classic Platform.
tcpIpProps	EthTcpIpProps	0..1	ref	EcuInstance specific TcpIp Stack attributes. **Tags:** atp.Splitkey=tcpIpProps This Attribute is only used by the AUTOSAR Classic Platform.
wakeUpOver BusSupported	Boolean	0..1	attr	Driver support for wakeup over Bus. This Attribute is only used by the AUTOSAR Classic Platform.

Table 3.1: EcuInstance

[constr_3008] `EcuInstance` subelements

Imposition time: CP: `IT_SysDesc`

⌈The `CommunicationConnector` and the `CommunicationController` that is referenced by the `CommunicationConnector` shall be owned by the same `EcuInstance`.⌋

Class	**ClientIdRange**			
Note	With this element it is possible to restrict the Client Identifier of the transaction handle that is generated by the client RTE for inter-Ecu Client/Server communication to an allowed range of numerical values.			
Base	*ARObject*			
Aggregated by	EcuInstance.clientIdRange, MachineDesign.clientIdRange			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
upperLimit	Limit	0..1	attr	This specifies the upper limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 3.2: ClientIdRange

[constr_3116] Overlap of `ClientIdRange`s in the context of the enclosing System

Imposition time: CP: `IT_SysDesc`

⌈The `ClientIdRange` defined for an `EcuInstance` shall not overlap with the `ClientIdRange` of any other `EcuInstance` or `MachineDesign`s in the context of the enclosing System.⌋

[constr_5396] Existence of `ClientIdRange.lowerLimit`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdRange`, the attribute `lowerLimit` shall exist.⌋

[constr_5397] Existence of `ClientIdRange.upperLimit`

Imposition time: CP: `IT_SysDesc`

⌈For each `ClientIdRange`, the attribute `upperLimit` shall exist.⌋

3.1.2 Communication Controller

[TPS_SYST_01007] Definition of CommunicationController ⌈A `CommunicationController` is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium.⌋

[TPS_SYST_01008] Assign `CommunicationController` to the AUTOSAR Communication Peripheral ⌈In order to illustrate the relationship of an `CommunicationController` to the `HwElement` with `category` `CommunicationController` defined in the ECU Resource Description, a mapping between these two classes may be specified using the `CommunicationControllerMapping` (see [TPS_SYST_01014]).⌋

Class	«atpVariation» ***CommunicationController*** (abstract)			
Note	The communication controller is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationController*, EthernetCommunicationController, FlexrayCommunicationController, *LinCommunicationController*, UserDefinedCommunicationController			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
wakeUpByController Supported	Boolean	0..1	attr	Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.

Table 3.3: CommunicationController

An `EcuInstance` may be connected to the same `PhysicalChannel` via two or more `CommunicationController`s. In most cases each of these `CommunicationController`s will have a dedicated `CommunicationConnector`.

There may be rare use cases where an `EcuInstance` is connected to a `PhysicalChannel` by one `CommunicationController` that in turn uses more than one `CommunicationConnector`.

3.1.3 Communication Connector

[TPS_SYST_01009] Definition of `CommunicationConnector` ⌈An `EcuInstance` uses `CommunicationConnector` elements in order to describe its bus interfaces and to specify the sending/receiving behavior.⌋

The relationship between an `EcuInstance`, a `CommunicationController`, and a `PhysicalChannel` is expressed by letting a `PhysicalChannel` reference a `CommunicationConnector` (which in turn is aggregated by `EcuInstance`) and which also has the ability to reference a `CommunicationController`.

[constr_5398] Existence of CommunicationConnector.commController

Imposition time: CP: IT_SysDesc

⌈For each CommunicationConnector, the reference in the role commController shall exist.⌋

Class	***CommunicationConnector*** (abstract)			
Note	The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunication Connector, LinCommunicationConnector, UserDefinedCommunicationConnector			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
commController	Communication Controller	0..1	ref	Reference to the communication controller. The CommunicationConnector and referenced CommunicationController shall be aggregated by the same ECUInstance. The communicationController can be referenced by several CommunicationConnector elements. This is important for the FlexRay Bus. FlexRay communicates via two physical channels. But only one controller in an ECU is responsible for both channels. Thus, two connectors (for channel A and for channel B) shall reference to the same controller. **Tags:** atp.Splitkey=commController
createEcu WakeupSource	Boolean	0..1	attr	If this parameter is available and set to true then a channel wakeup source shall be created for the Physical Channel referencing this CommunicationConnector.
dynamicPncTo Channel Mapping Enabled	Boolean	0..1	attr	Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this CommunicationConnector and its respective Physical Channel. **Tags:** atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ecuCommPort Instance	CommConnectorPort	*	aggr	An ECUs reception or send ports. atpVariation: If signals/PDUs/frames are variable, the corresponding ports shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuCommPortInstance, ecuCommPort Instance.shortName, ecuCommPortInstance.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	*CommunicationConnector* (abstract)			
explicitWakeup Channel	PhysicalChannel	*	ref	Defines a restriction which PhysicalChannels shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupChannel, explicitWakeup Channel.physicalChannel, explicitWakeup Channel.variationPoint.shortLabel vh.latestBindingTime=postBuild
explicitWakeup Pnc	PncMappingIdent	*	ref	Defines a restriction which PNCs shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupPnc, explicitWakeupPnc.pnc MappingIdent, explicitWakeupPnc.variationPoint.short Label vh.latestBindingTime=postBuild
nmVariant	NmVariantEnum	0..1	attr	Defines the functionality of Network Management.
pncFilterArray Mask (ordered)	PositiveInteger	*	attr	Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.
pncGateway Type	PncGatewayTypeEnum	0..1	attr	Defines if this EcuInstance shall implement the Pnc Gateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several Ecu Instances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active". This Attribute is only used by the AUTOSAR Classic Platform.

Table 3.4: CommunicationConnector

Enumeration	**PncGatewayTypeEnum**
Note	Defines the PncGateway roles.
Aggregated by	*CommunicationConnector*.pncGatewayType
Literal	***Description***
active	The active PncGateway functionality shall be performed **Tags:** atp.EnumerationLiteralIndex=0
none	No PncGateway functionality shall be performed **Tags:** atp.EnumerationLiteralIndex=1
passive	The passive PncGateway functionality shall be performed **Tags:** atp.EnumerationLiteralIndex=2

Table 3.5: PncGatewayTypeEnum

Note: Use-case for the relation of several `CommunicationConnector`s assigned to one `PhysicalChannel` in the scope of one `EcuInstance`: One safety measure for a safety relevant ECU can be to have two transceivers (and two controllers) connected to the same network (Bus). In case a safety violation is detected one transceiver can be disabled and the respective Frames are blocked. The other transceiver stays active and keeps the ECU alive for diagnostics.

The CommunicationConnector.pncFilterArrayMask is configured per communication connector. This data mask is covering only the PNC Request Bit Vector payload of the NmPdu. Leading bytes (bytes in front of System.pncVectorOffset) and also bytes behind the PNC Request Bit Vector (e.g. bytes behind System.pncVectorOffset + System.pncVectorLength) are not included, as they do not contain PNC Request Bit Vector information.

The CommunicationConnector.pncFilterArrayMask is an ordered list of byte (uint8) values which represent the PNC Request Bit Vector layout. Specifically it is important to preserve this order when deriving the ECU configuration.

The number of CommunicationConnector.pncFilterArrayMask list elements corresponds to NmCluster.pncClusterVectorLength, if defined, or System.pncVectorLength.

[constr_3685] Allowed values for each element of pncFilterArrayMask

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value for each element of CommunicationConnector.pncFilterArrayMask shall be in the range between 0 and 255.⌋

[constr_3686] Allowed number of entries for pncFilterArrayMask

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The number of CommunicationConnector.pncFilterArrayMask elements shall be:

- NmCluster.pncClusterVectorLength, if defined
- System.pncVectorLength, otherwise.

⌋

3.1.4 Internal ECU handling upon a communication wakeup

An ECU may be woken up via communication activities on a communication bus / network. It is possible to define the expected behavior if communication wakeup has been qualified as valid:

- If EcuInstance.channelSynchronousWakeup is set to TRUE, then all channels and all configured PNCs are woken up upon a detection of a valid wakeup on a channel
- If EcuInstance.pncSynchronousWakeup is set to TRUE, then all PNCs and the corresponding channels are woken up upon a detection of a valid wakeup on a channel.
- If CommunicationConnector.explicitWakeupChannel is defined, then a list of to be woken up PhysicalChannels is defined.

- If CommunicationConnector.explicitWakeupPnc is defined, then a list of to be woken up PNCs is defined.

The reference CommunicationConnector.explicitWakeupChannel can only be used to define a dedicated set of to be woken up PhysicalChannels if EcuInstance.channelSynchronousWakeup if set to FALSE. This restriction is based on the basic software ComM specification.

The following specification items define the expected wakeup behavior based on the system description, this is not an exact configuration specification of the ECU State Manager and Communication Manager.

[constr_3824] No explicitWakeupChannel reference for channelSynchronousWakeup = TRUE

Imposition time: IT_SysDesc

⌈If EcuInstance.channelSynchronousWakeup is set to TRUE for an EcuInstance, then all CommunicationConnectors aggregated in the role connector shall not define any CommunicationConnector.explicitWakeupChannel reference.⌋

[TPS_SYST_03143] channelSynchronousWakeup is set to FALSE and no explicitWakeupChannel reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.channelSynchronousWakeup is set to FALSE and no CommunicationConnector.explicitWakeupChannel reference is defined, then there is no definition from the system template on which PhysicalChannels to wakeup. It is up to the integrator to define the to be woken up PhysicalChannels.⌋

[TPS_SYST_03144] channelSynchronousWakeup is set to FALSE and at least one explicitWakeupChannel reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.channelSynchronousWakeup is set to FALSE and at least one CommunicationConnector.explicitWakeupChannel reference is defined, then a dedicated wakeup source for this CommunicationConnector is derived and that wakeup source wakes exactly those PhysicalChannels that are referenced by CommunicationConnector.explicitWakeupChannel.

The CommunicationConnector is assigned to this wakeup source.⌋

[TPS_SYST_03145] channelSynchronousWakeup is set to TRUE ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE and EcuInstance.channelSynchronousWakeup is set to TRUE, then one wakeup source is derived which wakes up all PhysicalChannels of this EcuInstance.

All CommunicationConnectors are assigned to this wakeup source.⌋

The reference CommunicationConnector.explicitWakeupPnc can be used in two ways, depending on whether the EcuInstance.pncSynchronousWakeup attribute is set to TRUE or FALSE:

[TPS_SYST_03146] pncSynchronousWakeup is set to FALSE and no explicitWakeupPnc reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.pncSynchronousWakeup is set to FALSE and no CommunicationConnector.explicitWakeupPnc reference is defined, then there is no definition from the system template on which PNCs to wakeup. It is up to the integrator to define the to be woken up PNCs.⌋

[TPS_SYST_03147] pncSynchronousWakeup is set to FALSE and at least one explicitWakeupPnc reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.pncSynchronousWakeup is set to FALSE and at least one CommunicationConnector.explicitWakeupPnc reference is defined, then a dedicated wakeup source for this CommunicationConnector is derived and that wakeup source wakes exactly those PNCs that are referenced by CommunicationConnector.explicitWakeupPnc.

The CommunicationConnector is assigned to this wakeup source.⌋

[TPS_SYST_03148] pncSynchronousWakeup is set to TRUE and no explicitWakeupPnc reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.pncSynchronousWakeup is set to TRUE and no CommunicationConnector.explicitWakeupPnc reference is defined, then one wakeup source is derived which wakes up all PNCs of this EcuInstance.

Any CommunicationConnector with no CommunicationConnector.explicitWakeupPnc defined is assigned to this wakeup source.⌋

[TPS_SYST_03149] pncSynchronousWakeup is set to TRUE and at least one explicitWakeupPnc reference ⌈If CommunicationConnector.createEcuWakeupSource is set to TRUE, EcuInstance.pncSynchronousWakeup is set to TRUE and at least one CommunicationConnector.explicitWakeupPnc reference is defined, then a dedicated wakeup source for this CommunicationConnector is derived and that wakeup source wakes exactly those PNCs that are referenced by CommunicationConnector.explicitWakeupPnc.

The CommunicationConnector is assigned to this wakeup source.⌋

Note that in case [TPS_SYST_03149]: If further CommunicationConnectors with no CommunicationConnector.explicitWakeupPnc reference exist, then [TPS_SYST_03148] applies.

A wakeup indication shall be forwarded to the ECU State Manager to perform the according wakeup handling. Therefore so-called wakeup sources can be configured. A wakeup source can be referenced by one or multiple channels. The wakeup source itself can reference multiple PNCs or channels. Therefore it is possible by the ECU

State Manager to detect a wakeup on one channel and forward the wakeup indication to other channels or PNCs. Additionally, the Communication Manager provides the possibility to request all its channels and PNCs if one wakeup has been indicated (EcuInstance.channelSynchronousWakeup set to TRUE). It is possible to derive the wakeup source if CommunicationConnector.createEcuWakeupSource is set to TRUE.

If only a synchronous PNC wakeup is used, then EcuInstance.channelSynchronousWakeup should be set to FALSE and EcuInstance.pncSynchronousWakeup should be set to TRUE. A synchronous PNC wakeup can be achieved by setting CommunicationConnector.createEcuWakeupSource and EcuInstance.pncSynchronousWakeup to TRUE. This derives a wakeup source where all PNCs, which are configured for this EcuInstance, are added.

[constr_3741] Exclusive setting of channelSynchronousWakeup or pncSynchronousWakeup

Imposition time: CP: IT_SysDesc

⌈At most one of EcuInstance.channelSynchronousWakeup or EcuInstance.pncSynchronousWakeup shall be set to TRUE.⌋

The settings of EcuInstance.channelSynchronousWakeup, EcuInstance.pncSynchronousWakeup, and CommunicationConnector.createEcuWakeupSource regulates the creation of wakeup sources and the reference to PNCs and Channels.

If a CommunicationCluster has several PhysicalChannels defined, then for all those PhysicalChannels the attribute createEcuWakeupSource shall be set to the same value in the scope of one EcuInstance.

[constr_3742] Value for createEcuWakeupSource in the conntext of a CommunicationCluster

Imposition time: CP: IT_SysDesc

⌈The attribute CommunicationConnector.createEcuWakeupSource shall be set to the same value for all CommunicationConnectors in the scope of one EcuInstance which are referenced by PhysicalChannels that belong to the same CommunicationCluster.⌋

In case of EcuInstance.channelSynchronousWakeup = TRUE the example in figure 3.3 illustrates the derivation of one wakeup source *WS* for the entire EcuInstance.

This wakeup source is associated as being used in case a network wakeup on any of the connected networks (with CommunicationConnector.createEcuWakeupSource set to TRUE) is recognized. The result is, that all the connected PhysicalChannels are woken up.

Note: For the configuration of the ECU State Manager and Communication Manager it is enough to initiate the wakeup on any one of the PhysicalChannels, as channel-SynchronousWakeup = TRUE results in a wakeup on all PhysicalChannels.

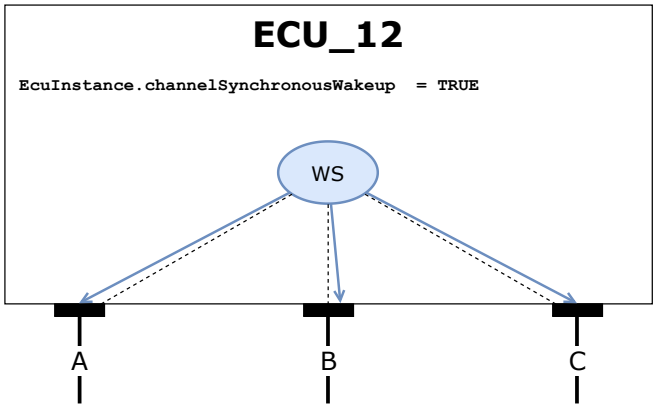

Figure 3.3: Example wakeup source definition of channelSynchronousWakeup

In case of EcuInstance.channelSynchronousWakeup = FALSE CommunicationConnector.explicitWakeupChannel references can be used to define the to be woken up channels:

In figure 3.4 the derived wakeup sources are used to react on a detected wakeup on PhysicalChannels *A*, *B*, or *C* individually. Thus, if a wakeup is detected on a dedicated PhysicalChannel, then the specified PhysicalChannels are woken up.

CommunicationConnector *A* defines the explicitWakeupChannel to be PhysicalChannels *A* and *B*. Thus, if a wakeup is detected on PhysicalChannels *A*, PhysicalChannels *A* and *B* will be woken up.

CommunicationConnector *B* defines the explicitWakeupChannel to be PhysicalChannels *B* and *C*. Thus, if a wakeup is detected on PhysicalChannels *B*, PhysicalChannels *B* and *C* will be woken up.

CommunicationConnector *C* defines the explicitWakeupChannel to be only PhysicalChannel *C*. Thus, if a wakeup is detected on PhysicalChannel *C*, only PhysicalChannel *C* will be woken up.

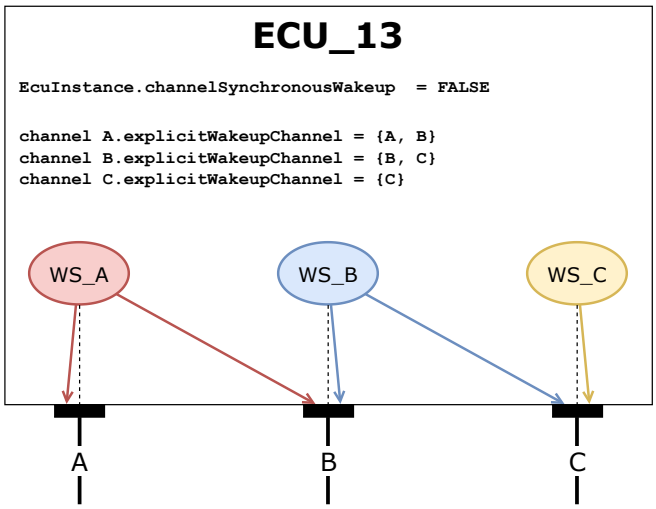

Figure 3.4: Example wakeup source definition of `explicitWakeupChannel` references

In case of `EcuInstance.pncSynchronousWakeup` = TRUE the example in figure 3.5 illustrates the derivation of one wakeup source *WS*.

This wakeup source is associated as being used in case a network wakeup on any of the connected networks (with `CommunicationConnector.createEcuWakeupSource` set to TRUE) is recognized. The result is, that all PNCs (and thus also all connected networks) are woken up.

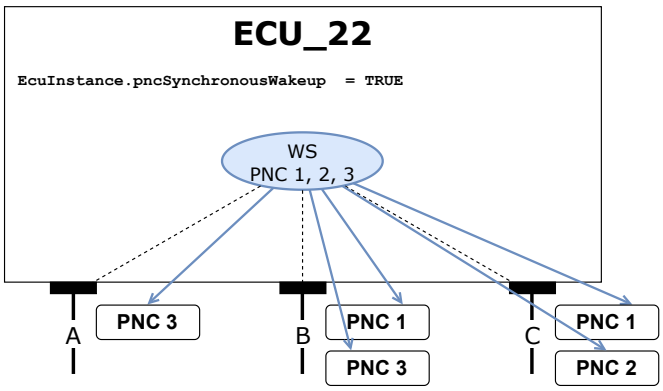

Figure 3.5: Example wakeup source definition of `pncSynchronousWakeup`

Additionally specifying `CommunicationConnector.explicitWakeupPnc` references can be used to limit the to be woken up PNCs:

In figure 3.6 the derived wakeup source *WS* is used to react on a detected wakeup on channels *A* or *B*. Thus, if a wakeup is detected on channel *A* or *B*, all PNCs are woken up.

CommunicationConnector defines explicitWakeupPnc to be only *PNC 2*. Thus, if a wakeup is detected on channel *C*, only *PNC 2* will be woken up. And since *PNC 2* is only active on cannel *C*, only channel *C* will be woken up.

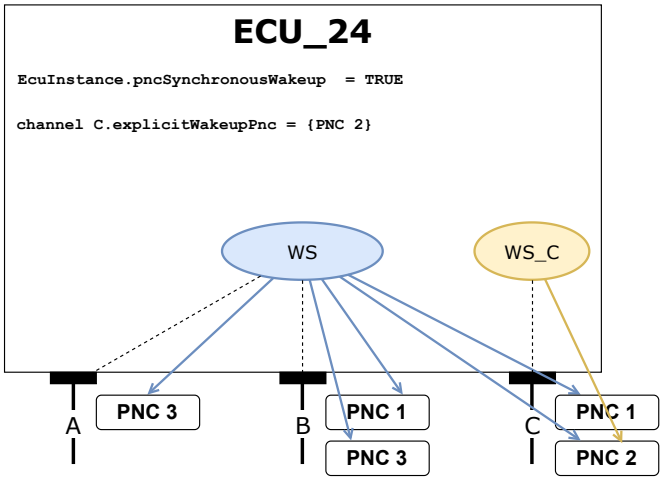

Figure 3.6: Example wakeup source definition of pncSynchronousWakeup and explicitWakeupPnc references

If a PhysicalChannel does not use PNCs (e.g. LIN) there is the possibility to additionally define CommunicationConnector.explicitWakeupChannel references to indicate that, besides the PNCs being woken up, also the referenced PhysicalChannels shall be woken up.

In figure 3.7 the EcuInstance.pncSynchronousWakeup = TRUE case already wakes up *A*, *B*, and *C*. Additionally the CommunicationConnector.explicitWakeupChannel reference to PhysicalChannel *D* defines that PhysicalChannel *D* will be woken up as well (although PhysicalChannel does not participate in any PNC).

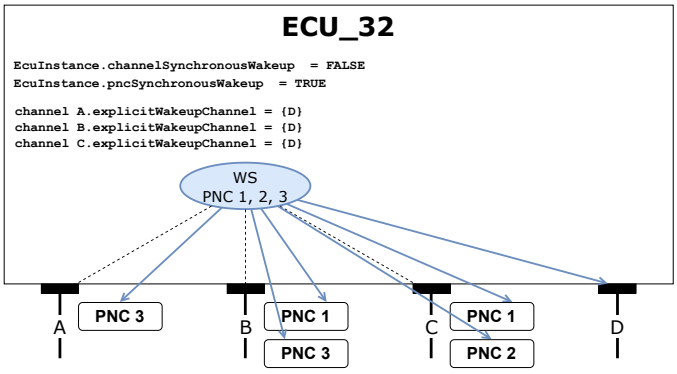

Figure 3.7: Example wakeup source definition of pncSynchronousWakeup and explicitWakeupChannel reference

3.2 Communication Clustering

3.2.1 Communication Cluster

[TPS_SYST_01010] Definition of CommunicationCluster ⌈CommunicationCluster represents a formal way to express that a number of EcuInstances are linked by an arbitrary topology (bus, star, ring, tree). Depending on the communication standard, a CommunicationCluster may either have exactly one or more (redundant) PhysicalChannels.⌋

Note that all ECUs within a CommunicationCluster communicate within the same address range.

Note that the same ECU can participate in more than one CommunicationCluster if it has more than one CommunicationConnector being referenced by PhysicalChannels owned by different CommunicationClusters.

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
Note	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	*AbstractCanCluster*, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine.
physical Channel	PhysicalChannel	*	aggr	This relationship defines which channel element belongs to which cluster. A channel shall be assigned to exactly one cluster, whereas a cluster may have one or more channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=physicalChannel, physicalChannel.shortName, physicalChannel.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

Table 3.6: CommunicationCluster

Some communication clusters need, additional to the general attributes which are valid for all communication clusters, specialized attributes to describe the individual communication cluster properties. The bustype-specific specializations of CommunicationCluster (Figure 3.8) are further detailed in chapter 3.3.

[constr_5395] Existence of `physicalChannel`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈Each `CommunicationCluster` shall aggregate at least one `PhysicalChannel` in the role `physicalChannel`.⌋

[constr_3829] `CommunicationCluster.comMLightTimeout` not applicable for `EthernetCluster`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `CommunicationCluster.comMLightTimeout` shall not be used in `EthernetCluster`.⌋

The reason for [constr_3829] is that in Ethernet a `ComMChannel` is created per VLAN that is represented by the `EthernetPhysicalChannel`.

3.2.2 Physical Channel

[TPS_SYST_01011] Definition of `PhysicalChannel` ⌈`PhysicalChannel` represents the communication medium that is used to send and receive information between communicating ECUs. Each `CommunicationCluster` has at least one `PhysicalChannel`.⌋

[constr_3373] Limitation on the number of `PhysicalChannel`s that are referencing a `CommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CommunicationConnector` shall only be referenced by at most one `PhysicalChannel`.⌋

Class	***PhysicalChannel*** (abstract)			
Note	A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one physical channel. Bus systems like CAN and LIN only have exactly one PhysicalChannel. A FlexRay cluster may have more than one PhysicalChannels that may be used in parallel for redundant communication. An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.#			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanPhysicalChannel*, EthernetPhysicalChannel, FlexrayPhysicalChannel, LinPhysicalChannel, UserDefinedPhysicalChannel			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***PhysicalChannel*** (abstract)			
comm Connector	Communication Connector	*	ref	Reference to the ECUInstance via a Communication Connector to which the channel is connected. atpVariation: Variable assignment of Physical Channels to different CommunicationConnectors is expressed with this variation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commConnector, comm Connector.communicationConnector, comm Connector.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
frameTriggering	FrameTriggering	*	aggr	One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=frameTriggering, frameTriggering.short Name, frameTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
iSignal Triggering	ISignalTriggering	*	aggr	One ISignalTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of ISignaltriggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.short Name, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
managed Physical Channel	PhysicalChannel	*	ref	Reference between a channel with role managing channel and a channel with role managed channel. **Tags:** atp.Splitkey=managedPhysicalChannel This Attribute is only used by the AUTOSAR Classic Platform.
pduTriggering	PduTriggering	*	aggr	One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.shortName, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

Table 3.7: PhysicalChannel

3.3 Specialized Attributes of the Topology Entities

According to their characteristic features, different communication standards like FlexRay, CAN, LIN, J1939 and Ethernet have individual attributes that need to be described additionally to the common topology classes. Figure 3.8 shows the specialization of the `CommunicationCluster` into the more specific `FlexrayCluster`, `CanCluster`, `J1939Cluster`, `LinCluster` and `EthernetCluster`.

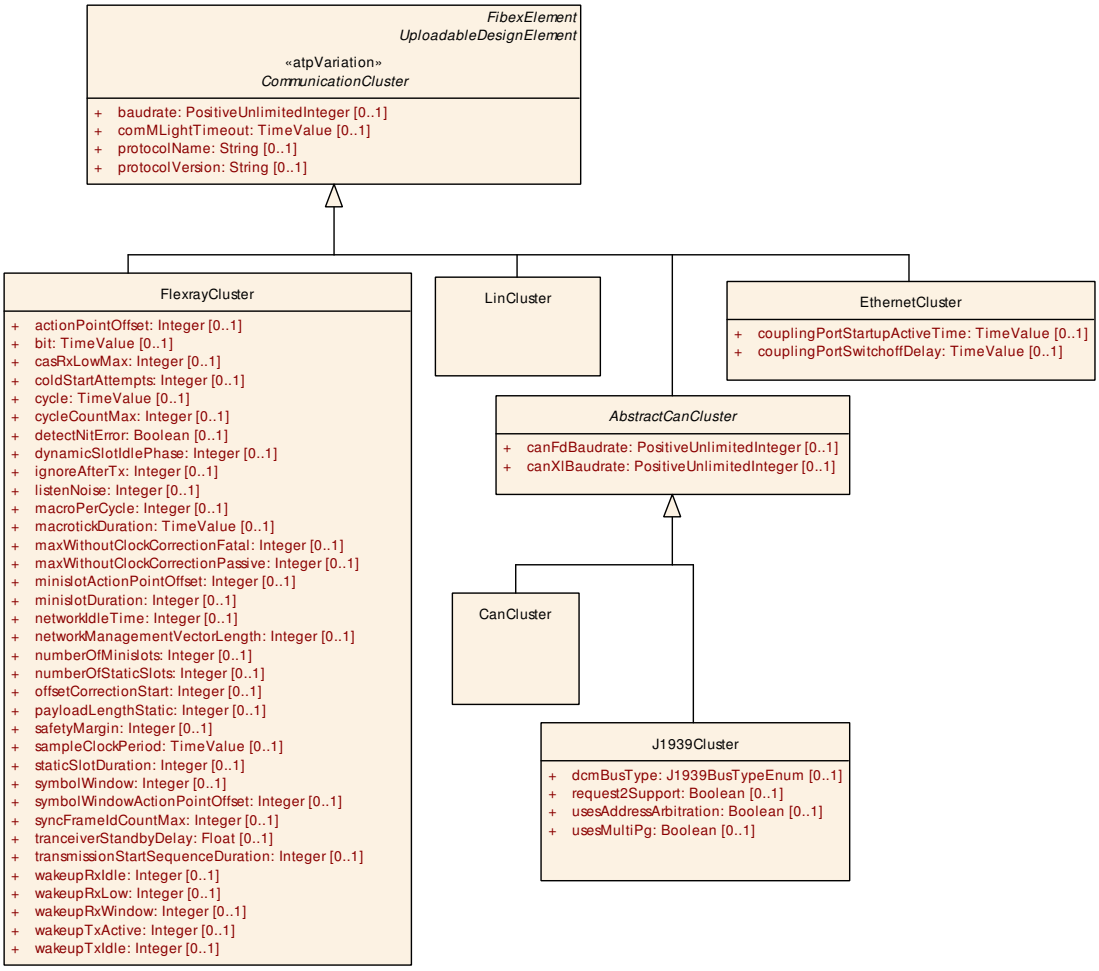

Figure 3.8: Specialized `CommunicationCluster` attributes (TopologyAttributeRefinement)

3.3.1 CAN

Modeling of the CAN bus is supported in the System Template by the means of four specialized meta-model classes: `CanCluster`, `CanCommunicationController`, `CanPhysicalChannel`, `CanCommunicationConnector` (Figure 3.9).

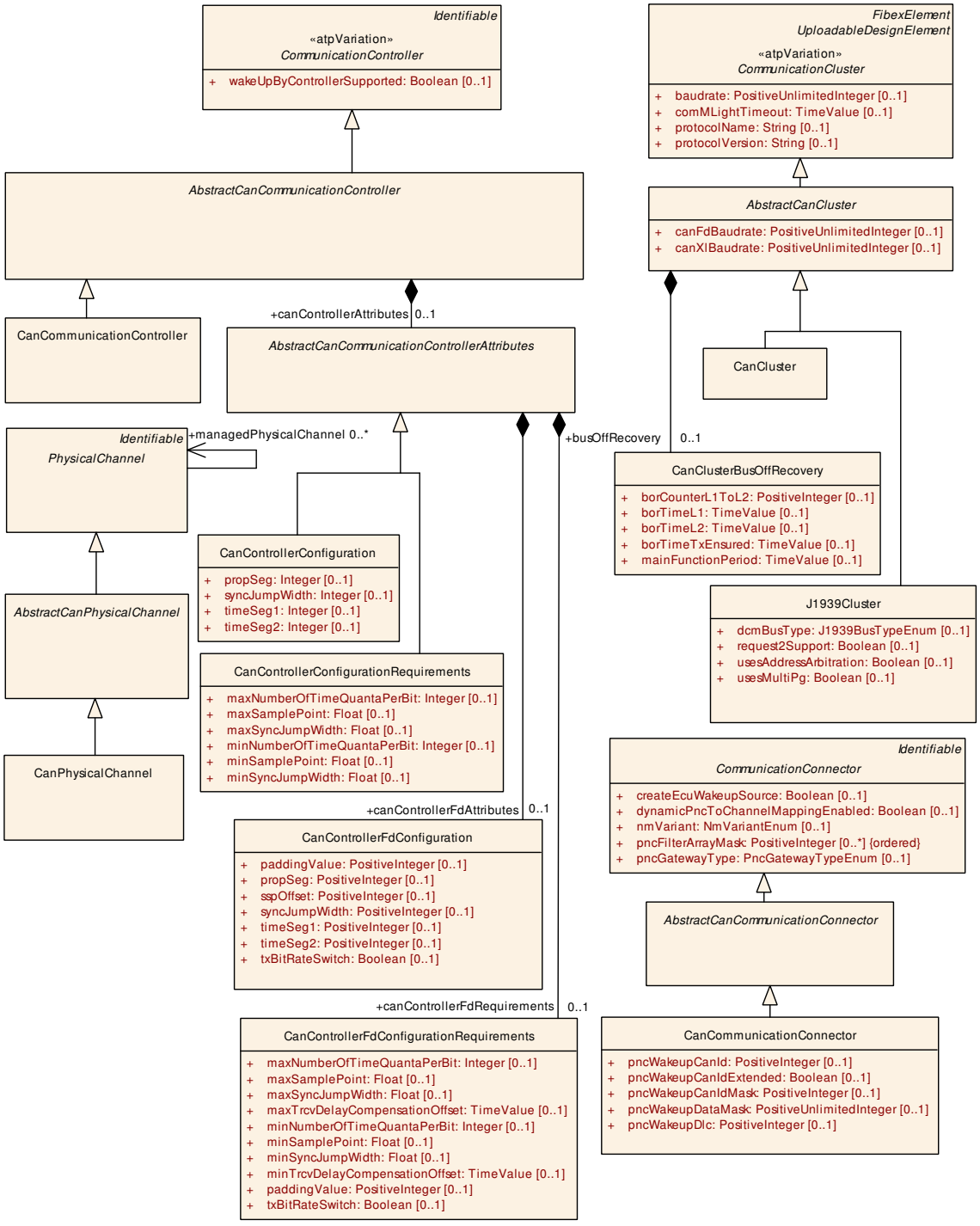

Figure 3.9: CAN bus elements (Fibex4Can_Topology)

3.3.1.1 CAN Cluster

CanCluster specifies the existence of a CAN cluster in the system's physical topology. It contains additional CAN-specific cluster-wide attributes. The common CAN attributes are collected in the AbstractCanCluster class.

Class	«atpVariation» ***AbstractCanCluster*** (abstract)			
Note	Abstract class that is used to collect the common J1939 and CAN Cluster attributes.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Subclasses	CanCluster, J1939Cluster			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
busOffRecovery	CanClusterBusOff Recovery	0..1	aggr	CAN bus off monitoring / recovery at system level. **Tags:** atp.Splitkey=busOffRecovery
canFdBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the controller in bits/s.
canXlBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the CAN XL controller in bits/s.

Table 3.8: AbstractCanCluster

Class	«atpVariation» **CanCluster**			
Note	CAN bus specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *AbstractCanCluster*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.9: CanCluster

Class	**CanClusterBusOffRecovery**			
Note	This element contains the attributes that are used to configure the CAN bus off monitoring / recovery at system level.			
Base	*ARObject*			
Aggregated by	*AbstractCanCluster*.busOffRecovery			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
borCounterL1ToL2	PositiveInteger	0..1	attr	This threshold defines the count of bus-offs after which the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).
borTimeL1	TimeValue	0..1	attr	This attribute defines the duration of the bus-off recovery time in level 1 (short recovery time) in seconds.
borTimeL2	TimeValue	0..1	attr	This attribute defines the duration of the bus-off recovery time in level 2 (long recovery time) in seconds.
borTimeTxEnsured	TimeValue	0..1	attr	This attribute defines the duration of the bus-off event check in seconds.
mainFunctionPeriod	TimeValue	0..1	attr	This attribute defines the cycle time of the function Can SM_MainFunction in seconds.

Table 3.10: CanClusterBusOffRecovery

3.3.1.2 CAN Communication Controller

CanCommunicationController is a specialization of the abstract CommunicationController class. It contains the specific CAN controller attributes needed for configuring the CAN stack in an ECU connected to a certain CAN cluster. The common CAN attributes are collected in the AbstractCanCommunicationController class. It is possible to specify the CAN Controller configuration parameters as exact values or as requirements that have to be respected by the ECU developer. Therefore the two elements CanControllerConfiguration and CanControllerConfigurationRequirements were created.

Class	«atpVariation» **CanCommunicationController**			
Note	CAN bus specific communication port attributes.			
Base	*ARObject*, *AbstractCanCommunicationController*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 3.11: CanCommunicationController

Class	«atpVariation» ***AbstractCanCommunicationController*** (abstract)			
Note	Abstract class that is used to collect the common CAN Controller attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanCommunicationController			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
canController Attributes	AbstractCan Communication ControllerAttributes	0..1	aggr	CAN Bit Timing configuration **Tags:** atp.Splitkey=canControllerAttributes

Table 3.12: AbstractCanCommunicationController

Class	***AbstractCanCommunicationControllerAttributes*** (abstract)			
Note	For the configuration of the CanController parameters two different approaches can be used: 1. Providing exact values which are taken by the ECU developer (CanControllerConfiguration). 2. Providing ranges of values which are taken as requirements and have to be respected by the ECU developer (CanControllerConfigurationRequirements).			
Base	*ARObject*			
Subclasses	CanControllerConfiguration, CanControllerConfigurationRequirements			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
canControllerFd Attributes	CanControllerFd Configuration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanFD frame. If this element exists the controller supports CanFD frames and the ECU developer shall take these values for the configuration of the CanFD controller. **Tags:** atp.Splitkey=canControllerFdAttributes

▽

△

Class	***AbstractCanCommunicationControllerAttributes*** (abstract)			
canControllerFd Requirements	CanControllerFd Configuration Requirements	0..1	aggr	Additional CanFD ranges of the bit timing related configuration of a CanFD controller. If this element exists the controller supports CanFD frames and the ECU developer shall take these ranges as requirements for the configuration of the CanFD controller. **Tags:** atp.Splitkey=canControllerFdRequirements
canControllerXl Attributes	CanControllerXl Configuration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanXL frame. If this element exists the controller supports CanXL frames and the ECU developer shall take these values for the configuration of the CanXL controller. **Tags:** atp.Splitkey=canControllerXlAttributes
canControllerXl Requirements	CanControllerXl Configuration Requirements	0..1	aggr	Additional CanXL ranges of the bit timing related configuration of a CanXL controller. If this element exists the controller supports CanXL frames and the ECU developer shall take these ranges as requirements for the configuration of the CanXL controller. **Tags:** atp.Splitkey=canControllerXlRequirements

Table 3.13: AbstractCanCommunicationControllerAttributes

Class	**CanControllerConfiguration**			
Note	This element is used for the specification of the exact CAN Bit Timing configuration parameter values.			
Base	*ARObject*, *AbstractCanCommunicationControllerAttributes*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes, CanXlProps.canConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
propSeg	Integer	0..1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	Integer	0..1	attr	The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
timeSeg1	Integer	0..1	attr	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
timeSeg2	Integer	0..1	attr	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2

Table 3.14: CanControllerConfiguration

Class	**CanControllerConfigurationRequirements**			
Note	This element allows the specification of ranges for the CAN Bit Timing configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	*ARObject*, *AbstractCanCommunicationControllerAttributes*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.

▽

△

Class	**CanControllerConfigurationRequirements**			
maxSyncJump Width	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJump Width	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Table 3.15: CanControllerConfigurationRequirements

3.3.1.2.1 CAN FD

[TPS_SYST_01154] CAN Controller support of CAN FD frames ⌈The bit timing configuration of CAN controllers for CAN FD frames is supported by the `CanControllerFdConfiguration` element that is aggregated by `AbstractCanCommunicationControllerAttributes`.⌋

[constr_3095] canControllerFdAttributes and canControllerFdRequirements are mutually exclusive

Imposition time: CP: `IT_SysDesc`

⌈The existence of `canControllerFdAttributes` and `canControllerFdRequirements` is mutually exclusive.⌋

[constr_3518] Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Imposition time: CP: `IT_SysDesc`

⌈The value given for `CanControllerFdConfiguration.paddingValue` and `CanControllerFdConfigurationRequirements.paddingValue` shall be in the range from 0 to 255.⌋

Class	**CanControllerFdConfiguration**			
Note	Bit timing related configuration of a CAN controller for payload and CRC of a CAN FD frame.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdAttributes, CanXlProps.canFdConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**CanControllerFdConfiguration**			
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.
propSeg	PositiveInteger	0..1	attr	Specifies propagation delay in time quantas.
sspOffset	PositiveInteger	0..1	attr	Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
syncJumpWidth	PositiveInteger	0..1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	0..1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	0..1	attr	Specifies phase segment 2 in time quantas.
txBitRateSwitch	Boolean	0..1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table 3.16: CanControllerFdConfiguration

Class	**CanControllerFdConfigurationRequirements**			
Note	This element allows the specification of ranges for the CanFD bit timing configuration parameters. These ranges are taken as requirements and shall be respected by the ECU developer.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdRequirements			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.

▽

△

Class	**CanControllerFdConfigurationRequirements**			
minSyncJump Width	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.
txBitRateSwitch	Boolean	0..1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table 3.17: CanControllerFdConfigurationRequirements

3.3.1.2.2 CAN XL

CAN XL - as continued development of CAN FD - is able to carry large data per frame (up to 2048 bytes payload). The type of payload contained in a CAN XL frame is specified by the “SDU Type” field within the CAN XL frame header.

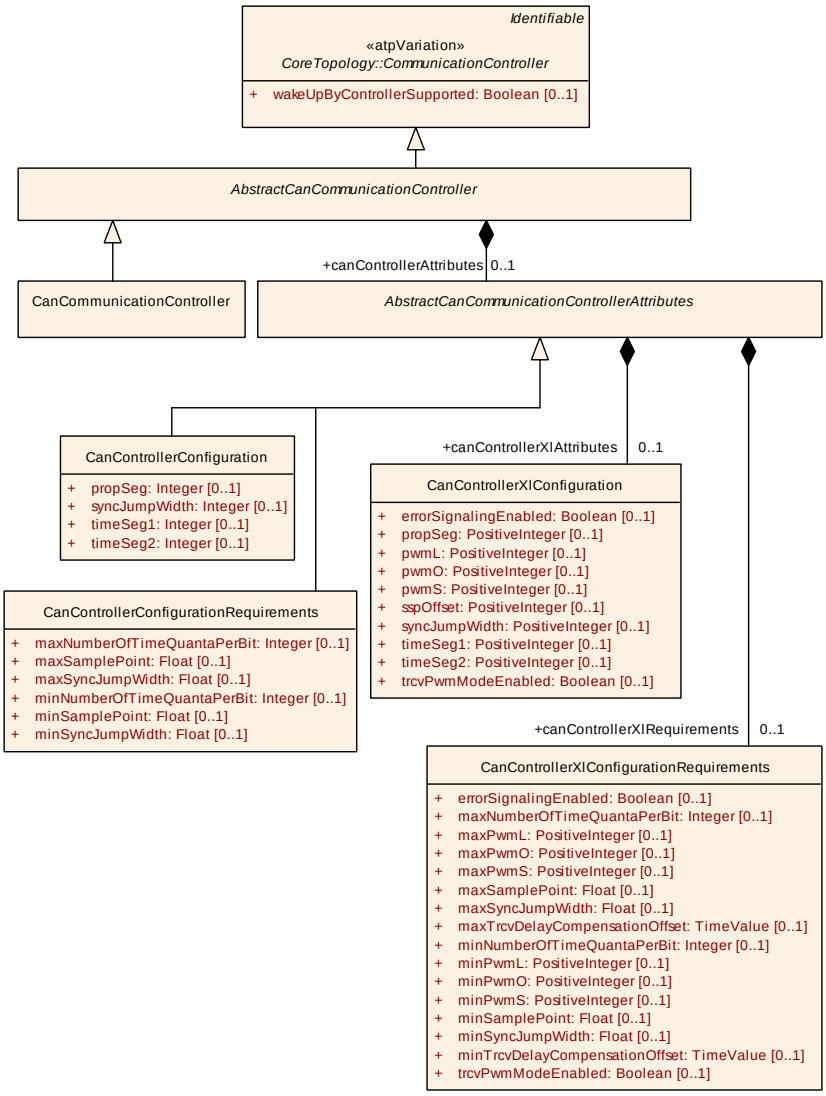

Figure 3.10: CAN XL bus elements

AUTOSAR CP supports the following SDU Type values (as specified by the *CiA* standardization - see Ch. 6.7.3.2) relating to the use cases of CAN XL communication:

1. **SDU Type 0x01h - Content based addressing:** This is equivalent to Classical CAN communication using the CAN XL specific PDU_N Identifier to denote the content of the CAN frame. For CAN XL, native payload data with up to 2048 bytes is transported in CAN XL frames.
2. **SDU Type 0x02h - Node addressing:** Using the “destination node address” field within the CAN XL frame header, a specific destination node that shall receive the CAN XL frame carrying native payload data is defined.
3. **SDU Type 0x03h - Classical CAN / CAN FD mapped tunneling:** A Classical CAN or CAN FD frame is tunneled inside a CAN XL frame.

4. **SDU Type 0x04h - IEEE 802.3 (Ethernet) tunneling:** An Ethernet frame is tunneled by a CAN XL frame without mapping of addressing information onto the CAN XL frame header.
5. **SDU Type 0x05h - IEEE 802.3 (Ethernet) mapped tunneling:** An Ethernet frame is tunneled by a CAN XL frame with partial mapping of the MAC destination address (4 specific bytes out of 6) and - if defined - the VLAN ID (lower 8 bits out of 12) onto fields of the CAN XL frame header.
6. **SDU Types 0xE0h..0xFEh - Manufacturer specific:** Manufacturer specific CAN XL frames for proprietary use cases.

Please note that for SDU Type 0x05h (Ethernet mapped tunneling), the mapping allows for hardware filtering of a subset of CAN XL frames carrying Ethernet frames which are not relevant for the receiving node. Since the full addressing information (i.e. full MAC address and full VLAN ID) is inside the Ethernet frame, the final reception filtering must be done by software processing of the tunneled Ethernet frame at the receiving node.

For all SDU Types described above, CAN XL requires the definition of `CanControllerXlConfiguration` or `CanControllerXlConfigurationRequirements` which contain physical layer related configuration parameters or requirements, respectively, as well as the definition of `CanXlFrameTriggeringProps` (aggregated by `CanFrameTriggering`) which contains, amongst others, the `sduType` attribute.

For the use cases involving Ethernet tunneling (i.e. SDU Types 0x04h and 0x05h), the `EthernetCommunicationController` is the connecting model element as it establishes - via the reference `EthernetCommunicationController.canXlConfig` - the link between the `CanCommunicationController` (which is CAN XL by aggregating a `CanControllerXlConfiguration`) and the `EthernetCommunicationController` which represents the lower end of the Ethernet and TCP/IP stack to be connected to the “CAN XL tunnel”. Further details on this tunneling configuration are given in Ch. 3.3.5.4.

Please note that the Ethernet and TCP/IP stack “on top” of `EthernetCommunicationController` need to be defined in the usual Ethernet and TCP/IP way (including `EthernetPhysicalChannel` but with the exception of network management - see Ch. 6.9.5.2). By the above mentioned linking to the associated CAN XL based `CanCommunicationController`, the tunneling is realized and the respective Ethernet frames are sent and received through the associated CAN XL physical channel.

[TPS_SYST_03074] CAN Controller support of CAN XL frames ⌈The specific configuration of CAN controllers for CAN XL frames is supported by the `CanControllerXlConfiguration` and `CanControllerXlConfigurationRequirements` elements that are aggregated by `AbstractCanCommunicationControllerAttributes`.⌋

Please note that the CAN XL specific baudrate is configured by the attribute canXlBaudrate of the abstract element AbstractCanCluster.

[constr_3695] canControllerXlAttributes and canControllerXlRequirements are mutually exclusive

Imposition time: CP: IT_SysDesc

⌈The existence of canControllerXlAttributes and canControllerXlRequirements is mutually exclusive.⌋

[constr_3696] Mandatory attributes of CanControllerXlConfiguration

Imposition time: CP: IT_SysDesc

⌈A CanControllerConfiguration configuring a CAN XL controller shall aggregate CanControllerXlConfiguration with the following attributes defined:

- errorSignalingEnabled
- propSeg
- syncJumpWidth
- timeSeg1
- timeSeg2
- trcvPwmModeEnabled

⌋

[constr_3697] Latest existence time of CanControllerXlConfiguration and CanControllerXlConfigurationRequirements

Imposition time: CP: IT_SysDesc

⌈In case that a CAN XL controller is configured, then either CanControllerXlConfiguration or CanControllerXlConfigurationRequirements shall exist within their aggregating class CanControllerConfiguration or CanControllerConfigurationRequirements.⌋

[constr_3698] Value of errorSignalingEnabled

Imposition time: CP: IT_SysDesc

⌈The attribute errorSignalingEnabled shall be set to FALSE if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3699] Existence of pwmL

Imposition time: CP: IT_SysDesc

⌈The attribute pwmL shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3700] Existence of pwmO

Imposition time: CP: IT_SysDesc

⌈The attribute pwmO shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

[constr_3701] Existence of pwmS

Imposition time: CP: IT_SysDesc

⌈The attribute pwmS shall be defined if trcvPwmModeEnabled is set to TRUE.⌋

Class	**CanControllerXlConfiguration**			
Note	This meta-class represents the CAN XL-specific controller attributes.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlAttributes, CanXlProps.canXlConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
propSeg	PositiveInteger	0..1	attr	Specifies propagation delay in time quantas.
pwmL	PositiveInteger	0..1	attr	Specifies the PWM long phase length.
pwmO	PositiveInteger	0..1	attr	Specifies the PWM time offset.
pwmS	PositiveInteger	0..1	attr	Specifies the PWM short phase length.
sspOffset	PositiveInteger	0..1	attr	Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
syncJumpWidth	PositiveInteger	0..1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	0..1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	0..1	attr	Specifies phase segment 2 in time quantas.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table 3.18: CanControllerXlConfiguration

Class	**CanControllerXlConfigurationRequirements**			
Note	This element allows the specification of ranges for the CAN XL configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlRequirements, CanXlProps.canXlConfig Reqs			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	CanControllerXlConfigurationRequirements			
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxPwmL	PositiveInteger	0..1	attr	Specifies the maximum PWM long phase length.
maxPwmO	PositiveInteger	0..1	attr	Specifies the maximum PWM time offset.
maxPwmS	PositiveInteger	0..1	attr	Specifies the maximum PWM short phase length.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minPwmL	PositiveInteger	0..1	attr	Specifies the minimum PWM long phase length.
minPwmO	PositiveInteger	0..1	attr	Specifies the minimum PWM time offset.
minPwmS	PositiveInteger	0..1	attr	Specifies the minimum PWM short phase length.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJump Width	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table 3.19: CanControllerXlConfigurationRequirements

For the use cases of tunneling Ethernet frames through CAN XL, specific CAN XL related configurations need to be defined for `EthernetCommunicationController` and `EthernetCommunicationConnector`. Please refer to Ch. 3.3.5.4 and Ch. 3.3.5.5 for the respective configuration details.

3.3.1.3 CAN Physical Channel

CanPhysicalChannel is a specialization of the abstract PhysicalChannel class. It contains the specific CAN PhysicalChannel attributes. The common CAN attributes are collected in the AbstractCanPhysicalChannel class.

Class	***AbstractCanPhysicalChannel*** (abstract)			
Note	Abstract class that is used to collect the common CAN PhysicalChannel attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Subclasses	CanPhysicalChannel			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.20: AbstractCanPhysicalChannel

Class	**CanPhysicalChannel**			
Note	CAN bus specific physical channel attributes.			
Base	*ARObject*, *AbstractCanPhysicalChannel*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.21: CanPhysicalChannel

[constr_3003] Number of CAN channels

Imposition time: CP: IT_SysDesc

⌈CAN clusters shall aggregate exactly one PhysicalChannel.⌋

3.3.1.4 CAN Communication Connector

CanCommunicationConnector is a specialization of the abstract CommunicationConnector class. It contains the specific CAN CommunicationConnector attributes. The common CAN attributes are collected in the AbstractCanCommunicationConnector class.

Class	***AbstractCanCommunicationConnector*** (abstract)			
Note	Abstract class that is used to collect the common CAN CommunicationConnector attributes.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanCommunicationConnector			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.22: AbstractCanCommunicationConnector

Class	CanCommunicationConnector			
Note	CAN bus specific communication connector attributes.			
Base	*ARObject*, *AbstractCanCommunicationConnector*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pncWakeupCanId	PositiveInteger	0..1	attr	CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupCanIdExtended	Boolean	0..1	attr	Defines whether pncWakeupCanId and pncWakeupCanIdMask shall be interpreted as extended or standard CAN ID.
pncWakeupCanIdMask	PositiveInteger	0..1	attr	Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDlc	PositiveInteger	0..1	attr	Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.

Table 3.23: CanCommunicationConnector

3.3.2 SAE J1939

Modeling of J1939 Communication Clusters is supported in the System Template with the `J1939Cluster` element that is derived from `AbstractCanCluster` (see figure 3.9).

Class	«atpVariation» **J1939Cluster**			
Note	J1939 specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *AbstractCanCluster*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dcmBusType	J1939BusTypeEnum	0..1	attr	Identification of a network to which the broadcast of messages is started by the DM13 command message.
request2Support	Boolean	0..1	attr	Enables support for the Request2 PGN (RQST2).
usesAddressArbitration	Boolean	0..1	attr	Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes. True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not care for contending address claims.

▽

△

Class	«atpVariation» **J1939Cluster**			
usesMultiPg	Boolean	0..1	attr	This attribute shall only be configured for a CanCluster that supports CAN FD. If configured and set to False, the J1939 communication on this cluster is expected to adhere to J1939-21. Otherwise, the J1939 communication is expected to adhere to J1939-22. Note: The communication of the J1939 specific modules is derived taking this setting into account.

Table 3.24: J1939Cluster

Enumeration	**J1939BusTypeEnum**
Note	Supported networks for Start/Stop Broadcast by DM13 command message.
Aggregated by	J1939Cluster.dcmBusType
Literal	*Description*
iso_9141	ISO 9141 network **Tags:** atp.EnumerationLiteralIndex=13
j1587	SAE J1587 network **Tags:** atp.EnumerationLiteralIndex=11
j1850	SAE J1850 network **Tags:** atp.EnumerationLiteralIndex=14
j1922	SAE J1922 network **Tags:** atp.EnumerationLiteralIndex=12
j1939_1	SAE J1939 network #1, primary vehicle network **Tags:** atp.EnumerationLiteralIndex=0
j1939_10	SAE J1939 network #10 **Tags:** atp.EnumerationLiteralIndex=9
j1939_11	SAE J1939 network #11 **Tags:** atp.EnumerationLiteralIndex=10
j1939_2	SAE J1939 network #2 **Tags:** atp.EnumerationLiteralIndex=1
j1939_3	SAE J1939 network #3 **Tags:** atp.EnumerationLiteralIndex=2
j1939_4	SAE J1939 network #4 **Tags:** atp.EnumerationLiteralIndex=3
j1939_5	SAE J1939 network #5 **Tags:** atp.EnumerationLiteralIndex=4
j1939_6	SAE J1939 network #6 **Tags:** atp.EnumerationLiteralIndex=5
j1939_7	SAE J1939 network #7 **Tags:** atp.EnumerationLiteralIndex=6
j1939_8	SAE J1939 network #8 **Tags:** atp.EnumerationLiteralIndex=7
j1939_9	SAE J1939 network #9 **Tags:** atp.EnumerationLiteralIndex=8

▽

△

Enumeration	**J1939BusTypeEnum**
other	Manufacture Specified Port **Tags:** atp.EnumerationLiteralIndex=15
proprietary_1	Proprietary network #1 **Tags:** atp.EnumerationLiteralIndex=16
proprietary_2	Proprietary network #2 **Tags:** atp.EnumerationLiteralIndex=17

Table 3.25: J1939BusTypeEnum

To describe the communication on a J1939Cluster CanFrameTriggerings are used that are aggregated by a CanPhysicalChannel.

[constr_3050] J1939Cluster uses exactly one CanPhysicalChannel

Imposition time: CP: IT_SysDesc

⌈A J1939Cluster shall aggregate exactly one CanPhysicalChannel.⌋

Please note that AUTOSAR supports only the four mentioned bus types. Still, an implementation could e.g. support J1708 [13] by means of a complex driver and would then need to assign the corresponding bus type.

[constr_9403] J1939TpProtocolTypeEnum values fd_bam, fd_bam_cmdt, and fd_cmdt shall only be used on CAN FD clusters

Imposition time: CP: IT_SysDesc

⌈The literals fd_bam, fd_bam_cmdt, and fd_cmdt of a J1939TpConnection.tpProtocolType shall only be used if the J1939TpConnection references a J1939TpNode in the transmitter or receiver role that in turn references a CommunicationConnector that references a CommunicationController with aggregated CanControllerFdConfiguration or CanControllerFdConfigurationRequirements.⌋

[constr_9404] J1939Cluster.usesMultiPg attribute shall only be set to a value on CAN FD clusters

Imposition time: CP: IT_SysDesc

⌈The J1939Cluster.usesMultiPg attribute shall only be set to true if the J1939Cluster also defines a AbstractCanCluster.canFdBaudrate.⌋

3.3.2.1 J1939Node

J1939 is not restricted to mere communication protocols. It also specifies the communication of software functions (a.k.a. J1939 Controller Applications) and thus has a very dedicated view on the software of an automotive ECU. The approach taken by J1939 with respect to software is very similar to the way AUTOSAR specifies software-components.

However, J1939 uses a different terminology and associates such a software-component with a predefined function. In addition, every function in J1939 has a standardized id. This function id is distributed by the Controller Application to the network as part of the so-called "name" which is a unique identifier representing a Controller Application within the J1939 network management.

Controller Applications, to some extent, fulfill the role of a "virtual ECU" since they are visible as independent entities on a J1939 network. In terms of AUTOSAR modeling, the role of a "virtual ECU" for J1939 Controller Applications is fulfilled by the meta-class J1939Node.

In order to make use of the AUTOSAR modeling approach for J1939 it is very helpful to associate a standardized function id with a software-component during an early phase of a development project. This function id shall later be mapped to a J1939Node with the identical J1939Node.nodeName.function.

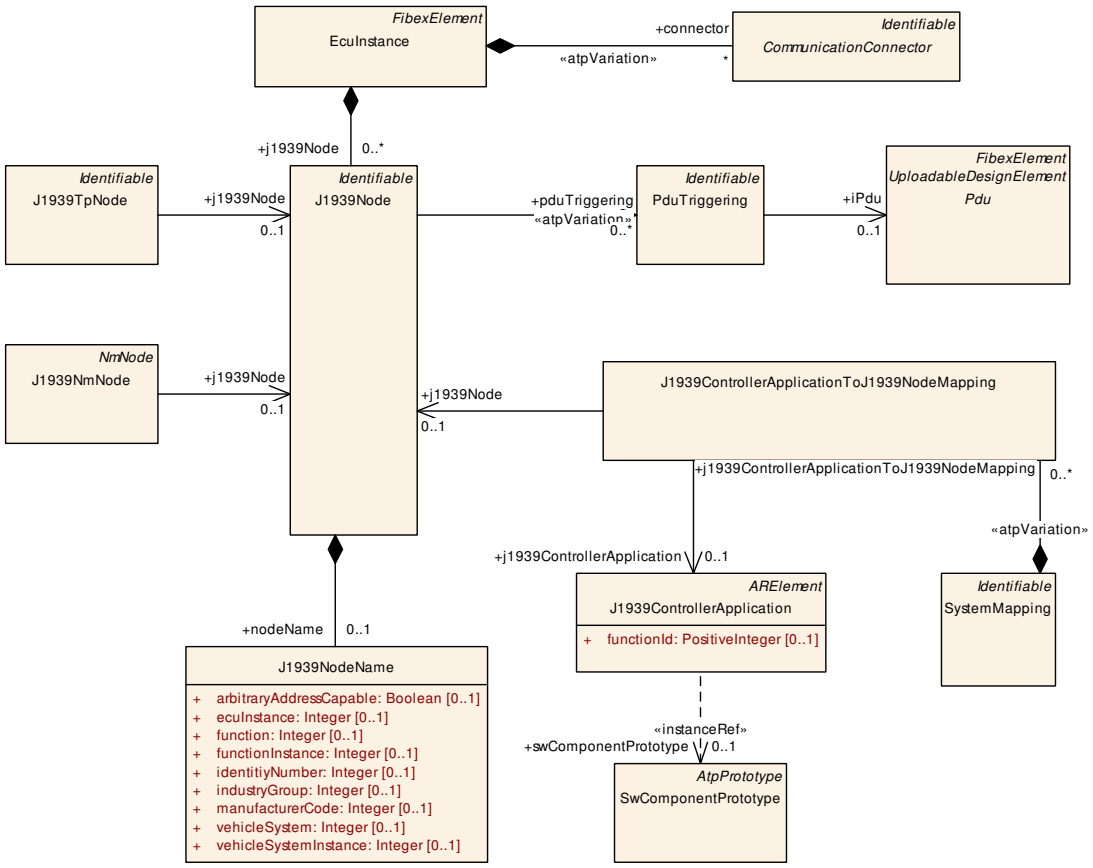

Figure 3.11: Modeling of J1939Node

[TPS_SYST_02435] Semantics of J1939Node ⌈A J1939Node represents a J1939ControllerApplication on an EcuInstance that allows the assignment of Pdus to the J1939ControllerApplication.⌋

[constr_9377] J1939Node.pduTriggering reference restriction

Imposition time: CP: IT_SysDesc

⌈A J1939Node shall not reference a PduTriggering in the role pduTriggering that in turn references an ISignalIPdu in the role iPdu.⌋

ISignalIPdus are indirectly assigned to a J1939ControllerApplication via the DataMapping of the SwComponentPrototype that is linked to the J1939ControllerApplication. The J1939Node.pduTriggering reference is used for the assignment of Pdus other than ISignalIPdus to the representative of the J1939ControllerApplication.

[constr_9378] J1939Nodes on the same EcuInstance shall not share PduTriggerings

Imposition time: CP: IT_SysDesc

⌈A PduTriggering shall not be referenced by several J1939Nodes that are aggregated by the same EcuInstance in the role pduTriggering.⌋

Class	**J1939Node**			
Note	J1939Node defined on the aggregating EcuInstance that is associated with J1939ControllerApplication via a J1939ControllerApplicationTo1939NodeMapping. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.j1939Node			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nodeName	J1939NodeName	0..1	aggr	Collectzion of attributes to configure the NodeName. **Tags:** atp.Splitkey=nodeName
pduTriggering	PduTriggering	*	ref	Reference to Pdus that are related to the J1939Node and in turn to a J1939ControllerApplication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.26: J1939Node

Class	**J1939NodeName**			
Note	This element contains attributes to configure the J1939NmNode NAME. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	J1939Node.nodeName			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arbitrary Address Capable	Boolean	0..1	attr	Arbitrary Address Capable field of the NAME of this node.
ecuInstance	Integer	0..1	attr	ECU Instance field of the NAME of this node.
function	Integer	0..1	attr	Function field of the NAME of this node.

▽

△

Class	J1939NodeName			
function Instance	Integer	0..1	attr	Function Instance field of the NAME of this node.
identitiyNumber	Integer	0..1	attr	Identity Number field of the NAME of this node.
industryGroup	Integer	0..1	attr	Industry Group field of the NAME of this node.
manufacturer Code	Integer	0..1	attr	Manufacturer Code field of the NAME of this node.
vehicleSystem	Integer	0..1	attr	Vehicle System field of the NAME of this node.
vehicleSystem Instance	Integer	0..1	attr	Vehicle System Instance field of the NAME of this node.

Table 3.27: J1939NodeName

[constr_3102] Restriction on usage of `J1939NodeName` attributes

Imposition time: CP: `IT_SysDesc`

⌈`EcuInstance`s that are referenced by the same `CanPhysicalChannel` of a `J1939Cluster` via `commConnector` shall not aggregate two `J1939Node`s with identical `J1939NodeName` attributes.⌋

In other words `J1939Node`s with identical `J1939NodeName` attributes shall not be used on the same `J1939Cluster`.

[constr_3103] Range of `ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `ecuInstance` range from 0 to 7.⌋

[constr_3104] Range of `function`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `function` range from 0 to 255.⌋

[constr_3105] Range of `functionInstance`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `functionInstance` range from 0 to 31.⌋

[constr_3106] Range of `identitiyNumber`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `identitiyNumber` range from 0 to 2097151.⌋

[constr_3107] Range of `industryGroup`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `industryGroup` range from 0 to 7.⌋

[constr_3108] Range of `manufacturerCode`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `manufacturerCode` range from 0 to 2047.⌋

[constr_3109] Range of `vehicleSystem`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `vehicleSystem` range from 0 to 127.⌋

[constr_3110] Range of `vehicleSystemInstance`

Imposition time: CP: `IT_SysDesc`

⌈The allowed values of `vehicleSystemInstance` range from 0 to 15.⌋

[constr_9167] Existence of `J1939NodeName.arbitraryAddressCapable`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `arbitraryAddressCapable` shall exist.⌋

[constr_9168] Existence of `J1939NodeName.ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `ecuInstance` shall exist.⌋

[constr_9169] Existence of `J1939NodeName.function`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `function` shall exist.⌋

[constr_9170] Existence of `J1939NodeName.functionInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `functionInstance` shall exist.⌋

[constr_9171] Existence of `J1939NodeName.identitiyNumber`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `identitiyNumber` shall exist.⌋

[constr_9172] Existence of `J1939NodeName.industryGroup`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `industryGroup` shall exist.⌋

[constr_9173] Existence of `J1939NodeName.manufacturerCode`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `manufacturerCode` shall exist.⌋

[constr_9174] Existence of `J1939NodeName.vehicleSystem`

Imposition time: CP: `IT_SysDesc`

⌈For each `J1939NodeName`, the attribute `vehicleSystem` shall exist.⌋

[constr_9175] Existence of J1939NodeName.vehicleSystemInstance

Imposition time: CP: IT_SysDesc

⌈For each J1939NodeName, the attribute vehicleSystemInstance shall exist.⌋

Class	**J1939ControllerApplicationToJ1939NodeMapping**			
Note	This meta-class represents the ability to map a J1939ControllerApplication to a J1939Node. Note that this is similar but not identical to the mapping of SwComponentPrototypes to EcuInstances; for J1939 the semantics of an EcuInstance itself is basically replaced by a J1939Node. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	SystemMapping.j1939ControllerApplicationToJ1939NodeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
j1939Controller Application	J1939Controller Application	0..1	ref	Reference to the J1939 Controller Application that is mapped to the referenced J1939Node. **Tags:** atp.Splitkey=j1939ControllerApplication
j1939Node	J1939Node	0..1	ref	J1939Node that is the target of the J1939Controller ApplicationTo1939NodeMapping. **Tags:** atp.Splitkey=j1939Node

Table 3.28: J1939ControllerApplicationToJ1939NodeMapping

Class	**J1939ControllerApplication**			
Note	This element represents a J1939 controller application. **Tags:** atp.recommendedPackage=J1939ControllerApplications This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
functionId	PositiveInteger	0..1	attr	This attribute represents the numerical function id of the J1939 controller application.
swComponent Prototype	SwComponent Prototype	0..1	iref	This represents the SwComponentPrototype (which is typically typed by a CompositionSwComponentType) that corresponds to the J1939ControllerApplication. **Tags:** atp.Splitkey=swComponentPrototype, swComponentPrototype.contextComposition, swComponentPrototype.contextComponent, swComponentPrototype.targetComponent **InstanceRef implemented by:** ComponentInSystemInstanceRef

Table 3.29: J1939ControllerApplication

[constr_9379] Consistent mapping of software-component to J1939Node

Imposition time: CP: IT_SysDesc

⌈The value of attribute J1939Node.nodeName.function of a J1939Node referenced by J1939ControllerApplicationToJ1939NodeMapping in the role j1939Node shall be identical to the value of J1939ControllerApplication.functionId.⌋

[constr_9380] Consistent mapping of J1939ControllerApplication to EcuInstance

Imposition time: CP: IT_SysDesc

⌈A SwComponentPrototype that is referenced by a J1939ControllerApplication mapped to a specific J1939Node shall only be mapped to an EcuInstance that in turn owns the same J1939Node.⌋

[constr_5493] Existence of J1939ControllerApplication.functionId

Imposition time: CP: IT_SysDesc

⌈For each J1939ControllerApplication, the attribute functionId shall exist.⌋

[constr_9381] J1939Node shall be referenced by a single J1939ControllerApplicationToJ1939NodeMapping

Imposition time: CP: IT_SysDesc

⌈At most one J1939ControllerApplicationToJ1939NodeMapping shall reference a J1939Node in the role J1939ControllerApplicationToJ1939NodeMapping.j1939Node.⌋

3.3.3 FlexRay

Modeling of FlexRay clusters is supported in the System Template by the means of four specialized meta-model classes: `FlexrayCluster`, `FlexrayCommunicationConnector`, `FlexrayPhysicalChannel`, `FlexrayCommunicationController` (Figure 3.12).

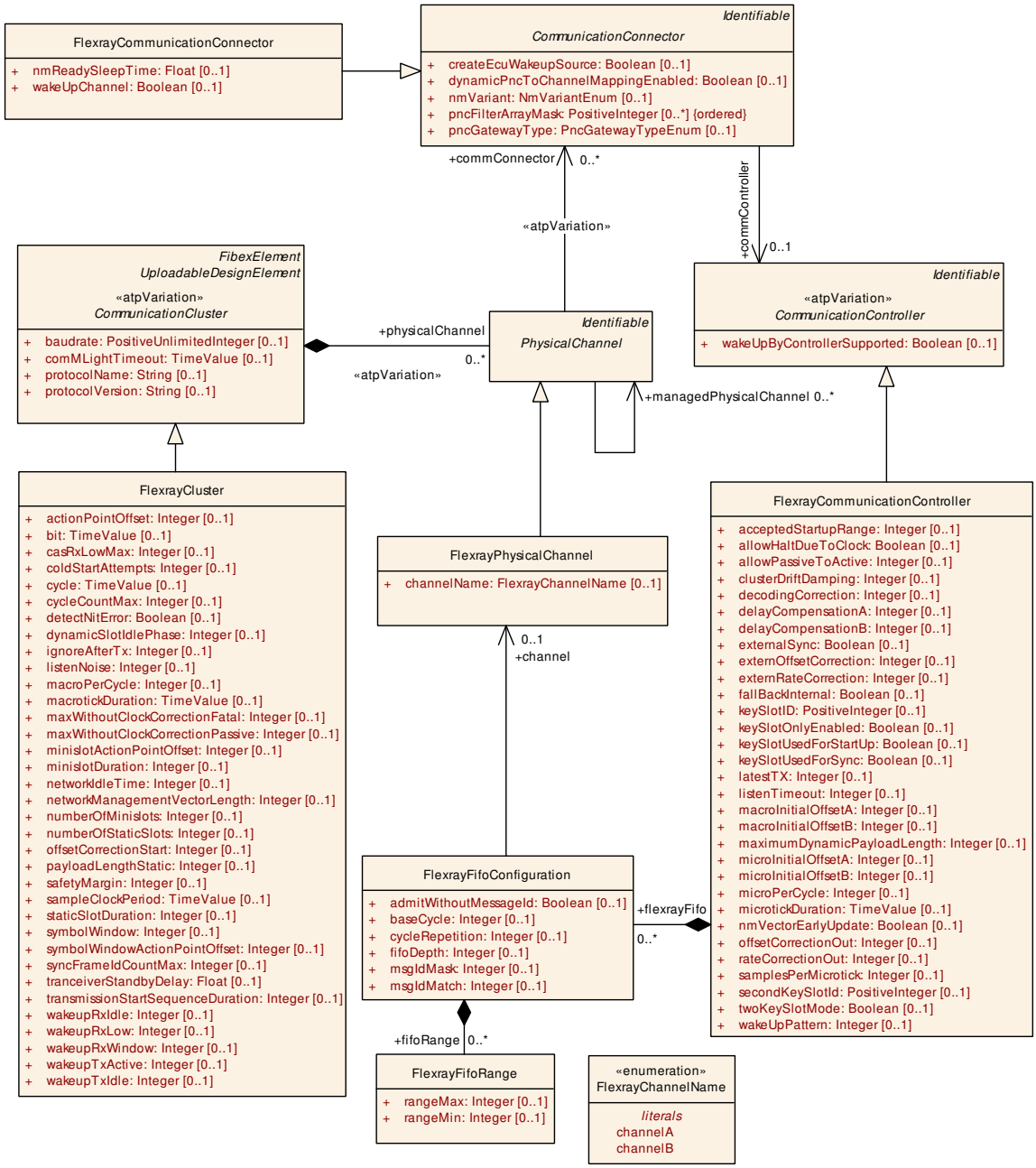

Figure 3.12: FlexRay cluster elements (Fibex4FlexRay_Topology)

3.3.3.1 FlexRay Cluster

`FlexrayCluster` specifies the existence of a FlexRay cluster in the system's physical topology. It contains additional FlexRay-specific cluster-wide attributes.

Class	«atpVariation» **FlexrayCluster**			
Note	FlexRay specific attributes to the physicalCluster **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
actionPoint Offset	Integer	0..1	attr	The offset of the action point in networks
bit	TimeValue	0..1	attr	Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPer Bit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLowMax	Integer	0..1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStart Attempts	Integer	0..1	attr	The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization
cycle	TimeValue	0..1	attr	Length of the cycle. Unit: seconds
cycleCountMax	Integer	0..1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
detectNitError	Boolean	0..1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSlotIdle Phase	Integer	0..1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfterTx	Integer	0..1	attr	Duration for which the bitstrobing is paused after transmission [gdBit].
listenNoise	Integer	0..1	attr	Upper limit for the start up and wake up listen timeout in the presence of noise. Expressed as a multiple of the cluster constant pdListenTimeout. Unit microticks
macroPerCycle	Integer	0..1	attr	The number of macroticks in a communication cycle
macrotick Duration	TimeValue	0..1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithout ClockCorrection Fatal	Integer	0..1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state.
maxWithout ClockCorrection Passive	Integer	0..1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state.
minislotAction PointOffset	Integer	0..1	attr	The Offset of the action point within a minislot. Unit: macroticks
minislotDuration	Integer	0..1	attr	The duration of a minislot (dynamic segment). Unit: macroticks.
networkIdle Time	Integer	0..1	attr	The duration of the network idle time in macroticks

△

Class	«atpVariation» **FlexrayCluster**			
network Management VectorLength	Integer	0..1	attr	Length of the Network Management vector in a cluster [bytes]
numberOf Minislots	Integer	0..1	attr	Number of Minislots in the dynamic segment.
numberOfStatic Slots	Integer	0..1	attr	The number of static slots in the static segment.
offsetCorrection Start	Integer	0..1	attr	Start of the offset correction phase within the Network Idle Time (NIT), expressed as the number of macroticks from the start of cycle. Unit: macroticks
payloadLength Static	Integer	0..1	attr	Globally configured payload length of a static frame. Unit: 16-bit WORDS.
safetyMargin	Integer	0..1	attr	Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has be resynchronized.
sampleClock Period	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlot Duration	Integer	0..1	attr	The duration of a slot in the static segment. Unit: macroticks
symbolWindow	Integer	0..1	attr	The duration of the symbol window. Unit: macroticks
symbolWindow ActionPoint Offset	Integer	0..1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameId CountMax	Integer	0..1	attr	Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.
tranceiver StandbyDelay	Float	0..1	attr	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full Flex Ray cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value. Not specifying a value or a value of 0 shall imply that the timer is not used.
transmission StartSequence Duration	Integer	0..1	attr	Number of bits in the Transmission Start Sequence [gd Bits].
wakeupRxIdle	Integer	0..1	attr	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit:bit Duration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.
wakeupRxLow	Integer	0..1	attr	Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit:bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow.
wakeupRx Window	Integer	0..1	attr	The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow.
wakeupTxActive	Integer	0..1	attr	Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP. Unit:bitDuration
wakeupTxIdle	Integer	0..1	attr	Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gDbit

Table 3.30: FlexrayCluster

[constr_5405] Existence of actionPointOffset

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute actionPointOffset shall exist.⌋

[constr_5406] Existence of bit

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute bit shall exist.⌋

[constr_5407] Existence of casRxLowMax

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster the attribute casRxLowMax shall exist.⌋

[constr_5408] Existence of coldStartAttempts

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute coldStartAttempts shall exist.⌋

[constr_5409] Existence of cycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute cycle shall exist.⌋

[constr_5410] Existence of cycleCountMax

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute cycleCountMax shall exist.⌋

[constr_5412] Existence of dynamicSlotIdlePhase

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute dynamicSlotIdlePhase shall exist.⌋

[constr_5414] Existence of listenNoise

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute listenNoise shall exist.⌋

[constr_5415] Existence of macroPerCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute macroPerCycle shall exist.⌋

[constr_5416] Existence of macrotickDuration

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute macrotickDuration shall exist.⌋

[constr_5417] Existence of maxWithoutClockCorrectionFatal

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute maxWithoutClockCorrectionFatal shall exist.⌋

[constr_5418] Existence of maxWithoutClockCorrectionPassive

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute maxWithoutClockCorrection-Passive shall exist.⌋

[constr_5419] Existence of minislotActionPointOffset

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute minislotActionPointOffset shall exist.⌋

[constr_5420] Existence of minislotDuration

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute minislotDuration shall exist.⌋

[constr_5421] Existence of networkIdleTime

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute networkIdleTime shall exist.⌋

[constr_5422] Existence of networkManagementVectorLength

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute networkManagementVectorLength shall exist.⌋

[constr_5423] Existence of numberOfMinislots

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute numberOfMinislots shall exist.⌋

[constr_5424] Existence of numberOfStaticSlots

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute numberOfStaticSlots shall exist.⌋

[constr_5425] Existence of offsetCorrectionStart

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute offsetCorrectionStart shall exist.⌋

[constr_5426] Existence of payloadLengthStatic

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute payloadLengthStatic shall exist.⌋

[constr_5428] Existence of staticSlotDuration

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute staticSlotDuration shall exist.⌋

[constr_5429] Existence of symbolWindow

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute symbolWindow shall exist.⌋

[constr_5431] Existence of syncFrameIdCountMax

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute syncFrameIdCountMax shall exist.⌋

[constr_5432] Existence of transmissionStartSequenceDuration

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute transmissionStartSequence-Duration shall exist.⌋

[constr_5433] Existence of wakeupRxIdle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute wakeupRxIdle shall exist.⌋

[constr_5434] Existence of wakeupRxLow

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute wakeupRxLow shall exist.⌋

[constr_5435] Existence of wakeupRxWindow

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute wakeupRxWindow shall exist.⌋

[constr_5436] Existence of wakeupTxActive

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute wakeupTxActive shall exist.⌋

[constr_5437] Existence of wakeupTxIdle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayCluster, the attribute wakeupTxIdle shall exist.⌋

[constr_5438] Existence of `sampleClockPeriod`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, the attribute `sampleClockPeriod` shall exist.⌋

3.3.3.2 FlexRay Communication Controller

`FlexrayCommunicationController` is a specialization of the `CommunicationController` class. It contains the specific FlexRay controller attributes needed for configuring the FlexRay stack in an ECU connected to a certain FlexRay cluster.

Class	«atpVariation» **FlexrayCommunicationController**			
Note	FlexRay bus specific communication port attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accepted StartupRange	Integer	0..1	attr	Expanded range of measured clock deviation allowed for startup frames during integration. Unit:microtick
allowHaltDueTo Clock	Boolean	0..1	attr	Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the Communication Controller is allowed to transition to POC:halt. If set to false, the Communication Controller will not transition to the POC:halt state but will enter or remain in the normal POC (passive State).
allowPassiveTo Active	Integer	0..1	attr	Number of consecutive even/odd cycle pairs that shall have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to 0, the Communication Controller is not allowed to transition from POC:norm
clusterDrift Damping	Integer	0..1	attr	The cluster drift damping factor used in clock synchronization rate correction in microticks
decoding Correction	Integer	0..1	attr	Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)
delay CompensationA	Integer	0..1	attr	Value used to compensate for reception delays on channel A Unit: Microticks. This optional parameter shall only be filled out if channel A is used.
delay CompensationB	Integer	0..1	attr	Value used to compensate for reception delays on channel B. Unit: Microticks. This optional parameter shall only be filled out if channel B is used.
externalSync	Boolean	0..1	attr	Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TT-E Time Triggered External Sync cluster) or locally synchronized.
externOffset Correction	Integer	0..1	attr	Fixed amount added or subtracted to the calculated offset correction term to facilitate external offset correction, expressed in node-local microticks.
externRate Correction	Integer	0..1	attr	Fixed amount added or subtracted to the calculated rate correction term to facilitate external rate correction, expressed in node-local microticks.

▽

△

Class	«atpVariation» **FlexrayCommunicationController**			
fallBackInternal	Boolean	0..1	attr	Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).
flexrayFifo	FlexrayFifo Configuration	*	aggr	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO. **Tags:** atp.Splitkey=flexrayFifo
keySlotID	PositiveInteger	0..1	attr	ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedForStartUp, keySlotUsedForSync, or keySlot OnlyEnabled are set to true the key slot value is mandatory.
keySlotOnly Enabled	Boolean	0..1	attr	Flag indicating whether or not the node shall enter key slot only mode following startup.
keySlotUsedFor StartUp	Boolean	0..1	attr	Flag indicating whether the Key Slot is used to transmit a startup frame.
keySlotUsedFor Sync	Boolean	0..1	attr	Flag indicating whether the Key Slot is used to transmit a sync frame.
latestTX	Integer	0..1	attr	The number of the last minislot in which a transmission can start in the dynamic segment for the respective node
listenTimeout	Integer	0..1	attr	Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks
macroInitial OffsetA	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel A is used.
macroInitial OffsetB	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel B is used.
maximum Dynamic PayloadLength	Integer	0..1	attr	Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.
microInitial OffsetA	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel A is used.
microInitial OffsetB	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel B is used.
microPerCycle	Integer	0..1	attr	The nominal number of microticks in a communication cycle
microtick Duration	TimeValue	0..1	attr	Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds

▽

△

Class	«atpVariation» **FlexrayCommunicationController**			
nmVectorEarly Update	Boolean	0..1	attr	Flag indicating when the update of the Network Management Vector in the CHI shall take place. If set to false, the update shall take place after the NIT. If set to true, the update shall take place after the end of the static segment.
offsetCorrection Out	Integer	0..1	attr	Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)
rateCorrection Out	Integer	0..1	attr	Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrection Out) Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift.
samplesPer Microtick	Integer	0..1	attr	Number of samples per microtick
secondKeySlot Id	PositiveInteger	0..1	attr	ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.
twoKeySlot Mode	Boolean	0..1	attr	Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.
wakeUpPattern	Integer	0..1	attr	Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster

Table 3.31: FlexrayCommunicationController

Class	**FlexrayFifoConfiguration**			
Note	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.			
Base	*ARObject*			
Aggregated by	FlexrayCommunicationController.flexrayFifo			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
admitWithout MessageId	Boolean	0..1	attr	Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.
baseCycle	Integer	0..1	attr	FIFO cycle counter acceptance criteria.
channel	FlexrayPhysicalChannel	0..1	ref	Fifo channel admittance criteria. **Tags:** atp.Splitkey=channel
cycleRepetition	Integer	0..1	attr	FIFO cycle counter acceptance criteria.
fifoDepth	Integer	0..1	attr	FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.
fifoRange	FlexrayFifoRange	*	aggr	FIFO Frame Id range acceptance criteria. **Tags:** atp.Splitkey=fifoRange
msgIdMask	Integer	0..1	attr	FIFO message identifier acceptance criteria (Mask filter).
msgIdMatch	Integer	0..1	attr	FIFO message identifier acceptance criteria (Match filter).

Table 3.32: FlexrayFifoConfiguration

Class	**FlexrayFifoRange**			
Note	FIFO Frame Id range acceptance criteria.			
Base	*ARObject*			
Aggregated by	FlexrayFifoConfiguration.fifoRange			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
rangeMax	Integer	0..1	attr	Max Range.
rangeMin	Integer	0..1	attr	Min Range.

Table 3.33: FlexrayFifoRange

[constr_5439] Existence of `admitWithoutMessageId`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `admitWithoutMessageId` shall exist.⌋

[constr_5440] Existence of `baseCycle`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `baseCycle` shall exist.⌋

[constr_5441] Existence of `cycleRepetition`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `cycleRepetition` shall exist.⌋

[constr_5442] Existence of `fifoDepth`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `fifoDepth` shall exist.⌋

[constr_5443] Existence of `msgIdMask`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `msgIdMask` shall exist.⌋

[constr_5444] Existence of `msgIdMatch`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoConfiguration`, the attribute `msgIdMatch` shall exist.⌋

[constr_5445] Existence of `fifoRange`

Imposition time: CP: `IT_SysDesc`

⌈Each `FlexrayFifoConfiguration` shall aggregate at least two `FlexrayFifoRange`s in the role `fifoRange`.⌋

[constr_5446] Existence of `rangeMax`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoRange`, the attribute `rangeMax` shall exist.⌋

[constr_5447] Existence of `rangeMin`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayFifoRange`, the attribute `rangeMin` shall exist.⌋

3.3.3.3 FlexRay Communication Connector

FlexrayCommunicationConnector adds the FlexRay specific attributes to the CommunicationConnector.

Class	FlexrayCommunicationConnector			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmReadySleep Time	Float	0..1	attr	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.
wakeUp Channel	Boolean	0..1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table 3.34: FlexrayCommunicationConnector

[constr_3508] Value of nmReadySleepTime

Imposition time: CP: IT_SysDesc

⌈The nmReadySleepTime value shall be a multiple of cycle * nmRepetitionCycle.⌋

3.3.3.4 FlexRay Physical Channel

FlexrayPhysicalChannel adds the FlexRay specific attributes to the PhysicalChannel.

Class	FlexrayPhysicalChannel			
Note	FlexRay specific attributes to the physicalChannel			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
channelName	FlexrayChannelName	0..1	attr	Name of the channel (Channel A or Channel B).

Table 3.35: FlexrayPhysicalChannel

Enumeration	FlexrayChannelName
Note	Name of the channel.
Aggregated by	FlexrayPhysicalChannel.channelName
Literal	**Description**
channelA	Channel A **Tags:** atp.EnumerationLiteralIndex=0
channelB	Channel B **Tags:** atp.EnumerationLiteralIndex=1

Table 3.36: FlexrayChannelName

[constr_3018] Number of FlexRay channels

Imposition time: CP: IT_SysDesc

⌈A FlexrayCluster shall use either one FlexrayPhysicalChannel with channelName set to either channelA or channelB or else two FlexrayPhysicalChannels with one channelName channelA and one channelName channelB.⌋

[constr_5448] Existence of channelName

Imposition time: CP: IT_SysDesc

⌈For each FlexrayPhysicalChannel, the attribute channelName shall exist.⌋

3.3.4 LIN

A LinCluster consists of exactly one master node connected to several slave nodes. The master is responsible for providing the frame headers on the bus according to a predefined schedule, whereas the slaves send or receive the actual frame information ([9]).

[TPS_SYST_01012] Different Properties of LinMaster and LinSlave ⌈In the System Template the different properties of master and slave nodes are handled by deriving the LIN-specific subclasses LinMaster and LinSlave as specializations of LinCommunicationController.⌋

AUTOSAR supports the stand-alone definition of both LIN masters and LIN slaves.

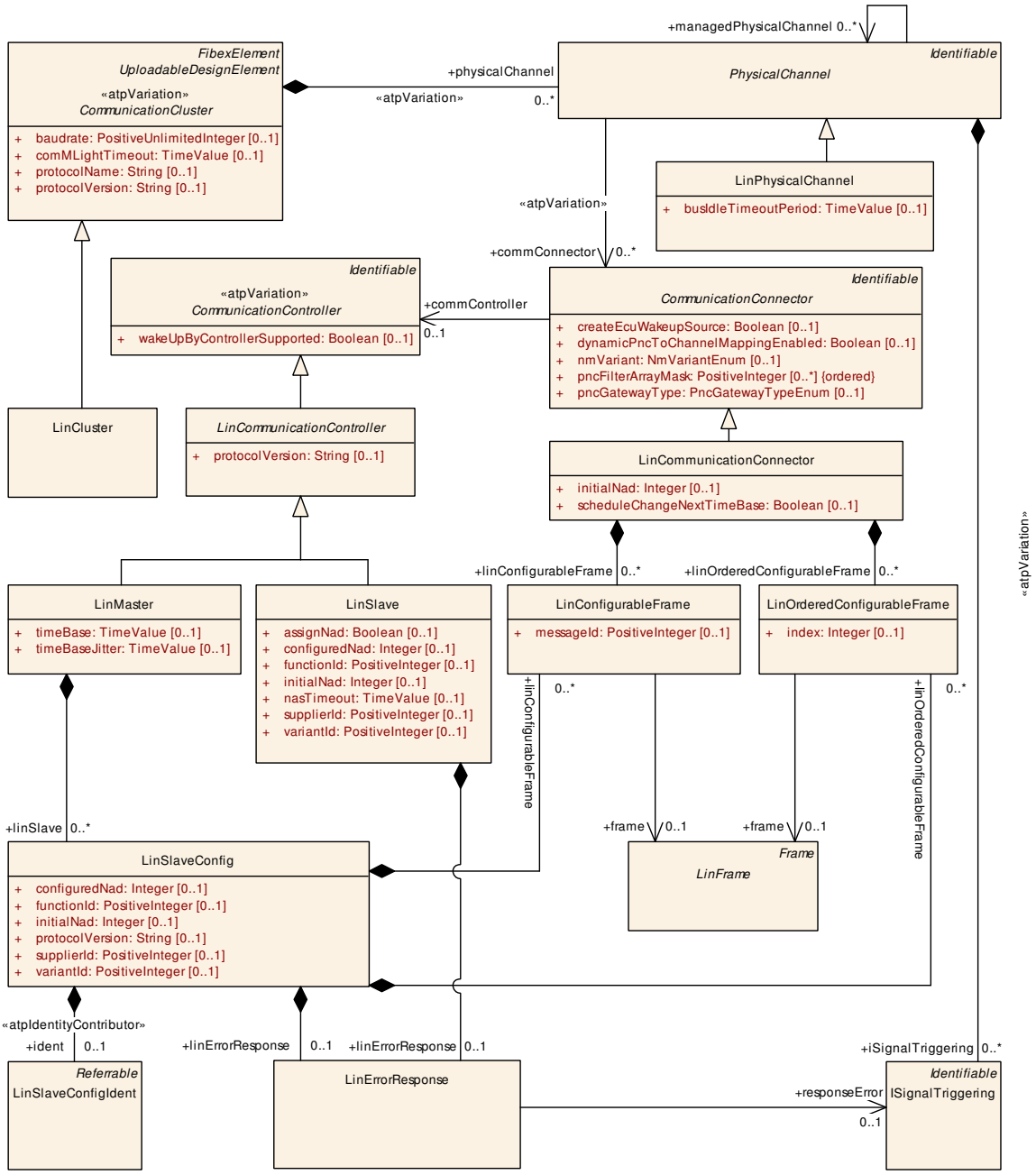

Figure 3.13: Specialized `LinCommunicationController` attributes (Fibex4Lin_Topology)

[constr_5252] `LinSlaveConfig.protocolVersion` shall exist

Imposition time: CP: `IT_SysDesc`

⌈The attribute `LinSlaveConfig.protocolVersion` shall exist.⌋

3.3.4.1 LIN Cluster

`LinCluster` specifies the existence of a LIN cluster in the system's physical topology.

Class	«atpVariation» **LinCluster**			
Note	LIN specific attributes **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 3.37: LinCluster

3.3.4.2 LIN Communication Controller

`LinCommunicationController` is a specialization of the `CommunicationController` class. It is an abstract class, to be further specialized by `LinMaster` and `LinSlave`.

Class	«atpVariation» ***LinCommunicationController*** (abstract)			
Note	LIN bus specific communication controller attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	LinMaster, LinSlave			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
protocolVersion	String	0..1	attr	Version specifier for a communication protocol.

Table 3.38: LinCommunicationController

[TPS_SYST_02257] Standardized values of `LinCommunicationController.protocolVersion` and `LinSlaveConfig.protocolVersion` ⌈The following values of attributes `LinCommunicationController.protocolVersion` and `LinSlaveConfig.protocolVersion` are standardized by AUTOSAR:

- LIN13
- LIN20
- LIN21
- LIN22
- ISO17987

⌋

[constr_5449] `LinCommunicationController.protocolVersion` shall exist

Imposition time: CP: `IT_SysDesc`

⌈The attribute `LinCommunicationController.protocolVersion` shall exist.⌋

3.3.4.3 LIN Master

LinMaster describes the existence of a LIN master task in a LIN topology node. As such it contains the attributes specific to a LIN master task.

Class	«atpVariation» **LinMaster**			
Note	Describing the properties of the refering ecu as a LIN master. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *LinCommunicationController*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
linSlave	LinSlaveConfig	*	aggr	LinSlaves that are handled by the LinMaster. **Tags:** atp.Splitkey=linSlave, linSlave.ident.shortName
timeBase	TimeValue	0..1	attr	Time base is mandatory for the master. It is not used for slaves. LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time." The time base shall be specified AUTOSAR conform in seconds.
timeBaseJitter	TimeValue	0..1	attr	The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves. LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)." The jitter shall be specified AUTOSAR conform in seconds.

Table 3.39: LinMaster

Class	**LinSlaveConfig**			
Note	Node attributes of LIN slaves that are handled by the LinMaster. In the System Description LIN slaves may be described in the context of the Lin Master. In an ECU Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the Lin Master. The values of attributes of LinSlaveConfig and the corresponding LinSlave shall be identical (if both are defined in a System Description). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinMaster.linSlave			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
configuredNad	Integer	0..1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	0..1	attr	LIN function ID.
ident	LinSlaveConfigIdent	0..1	aggr	This adds the ability to become referrable to LinSlave Config. **Stereotypes:** atpIdentityContributor
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.

▽

△

Class	**LinSlaveConfig**			
linConfigurable Frame	LinConfigurableFrame	*	aggr	List of all frames that are processed by the slave node **Tags:** atp.Splitkey=linConfigurableFrame
linError Response	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames. **Tags:** atp.Splitkey=linErrorResponse
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	aggr	List of all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.1 Assign-Frame-PID-Range command. **Tags:** atp.Splitkey=linOrderedConfigurableFrame
protocolVersion	String	0..1	attr	Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different.
supplierId	PositiveInteger	0..1	attr	LIN Supplier ID.
variantId	PositiveInteger	0..1	attr	Specifies the Variant ID.

Table 3.40: LinSlaveConfig

Class	**LinSlaveConfigIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable Lin SlaveConfig. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Referrable*			
Aggregated by	LinSlaveConfig.ident			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 3.41: LinSlaveConfigIdent

[constr_3219] The mutual existence of `LinSlave`s in the `LinMaster` EcuExtract

Imposition time: CP: `IT_EcuExt`

⌈`LinSlave`s shall not be part of the EcuExtract of the corresponding `LinMaster`.⌋

[constr_1655] The mutual existence of `LinMaster`s in the `LinSlave` EcuExtract

Imposition time: CP: `IT_EcuExt`

⌈A `LinMaster` shall not be part of the EcuExtract of a corresponding `LinSlave`.⌋

[TPS_SYST_02101] Usage of `LinSlaveConfig` in Ecu Extract ⌈In order to configure `LinMaster` in a `System` with `category` `ECU_EXTRACT` the `LinSlaveConfig` aggregated by the `LinMaster` shall be used.⌋

Please note that, in concordance with [TPS_SYST_02101], even if the `LinSlave` **can** be modeled independently of the `LinMaster` it still makes sense that the configuration of the `LinMaster` **positively contains the aggregation of the `LinSlaveConfig`** in the role `linSlave`.

In other words, the configuration of a `LinMaster` is **not affected** by the question of whether or not the `LinSlave` is explicitly modeled.

This statement is valid for both the existence of the LinMaster in a System of category SYSTEM_DESCRIPTION or in a System of category ECU_EXTRACT.

The actual correspondence between a Lin slave described by means of the LinSlaveConfig and the actual model of the LinSlave shall be determined by identifying pairs of LinSlaveConfig and LinSlave with an identical set of the attributes that are equally named in both meta-classes. This rule does not apply for the shortName.

Another relevant condition for finding pairs of corresponding LinSlaveConfig and LinSlave is obviously that the LinMaster that aggregates the LinSlaveConfig shall be connected to the same LinCluster to which the corresponding LinSlave is connected.

Of course, this condition can only be checked in the context of a System of category SYSTEM_DESCRIPTION or perhaps SYSTEM_EXTRACT.

3.3.4.4 LIN Slave

AUTOSAR supports the definition of a stand-alone LIN slave[1]. In other words, it is possible to define an ECU Extract that contains the modeling of a LIN slave independently of the modeling of the LIN master.

That said, the ability to define properties of the LIN slave in the context of the LIN master in the form of LinMaster.linSlave still exists and can be used where applicable.

[TPS_SYST_05018] Semantics of meta-class LinSlave ⌈Meta-class LinSlave describes the existence of a LIN slave task in a LIN topology node. It describes the attributes of a single LIN slave node.⌋

[TPS_SYST_05019] Semantics of LinErrorResponse.responseError ⌈Each Lin slave has the ability to set an error bit in the response part of one specific LinUnconditionalFrame in the event of errors occurring on frame level. The error bit is modeled by means of a reference to an ISignalTriggering in the role LinErrorResponse.responseError.⌋

Please note that because the response error bit applies for frame errors the responsibility for setting the response error bit lies exclusively at the LinIf.

In the event of such an error, the LinIf on the Lin slave Ecu calls Com_SendSignal() to set the value of the response error bit if applicable[2] and thus

[1] In former versions of this specification document the properties of a LIN slave could only be defined in the context of the corresponding LIN master

[2] In principle, the LinIf on the Lin slave Ecu could directly patch the value of the response error bit into a Tx Pdu before sending but in this case the LinIf would have to pick the correct Pdu and patch it accordingly. In other words, the LinIf would replicate a certain amount of Com functionality. the usage of an ISignalTriggering significantly simplifies the implementation of the LinIf.

the system model needs to foresee the existence of an ISignalTriggering for this purpose.

Aside: on the Lin master, typically a piece of application software picks up the received error bit and uses it to e.g. increment a counter for debouncing purposes. If the counter exceeds a certain value in e.g. a given time interval the Lin master has to assume that a serious problem exists in the communication with this specific slave and react accordingly.

In terms of modeling, this means that if formally modeled application software on the Lin master exists that processes the error response bit (the receiving side of the error response bit) then a DataMapping to a SystemSignal that carries the response error bit needs to be defined.

It is important to understand that application software on a Lin slave positively has no business of setting the value of the response error bit. In other words: on the sending side (i.e. the Lin slave), the application software is not affected and therefore there shall not be a DataMapping on the sending side. This relation motivates the existence of [constr_1656].

[constr_1656] No application-level write access to LinErrorResponse.responseError on Lin slave

Imposition time: CP: IT_SysDesc

⌈The SystemSignal referenced in the role systemSignal by the ISignal referenced by the ISignalTriggering that in turn is referenced in the role LinErrorResponse.responseError shall not be referenced by a DataMapping that allows for writing to the SystemSignal.⌋

Class	«atpVariation» **LinSlave**			
Note	Describing the properties of the referring ecu as a LIN slave. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *LinCommunicationController*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignNad	Boolean	0..1	attr	This attribute has the ability to control whether the node configuration command 'Assign NAD' is supported.
configuredNad	Integer	0..1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	0..1	attr	LIN function ID
initialNad	Integer	0..1	attr	This attribute represents the initial NAD.
linError Response	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames. **Tags:** atp.Splitkey=linErrorResponse
nasTimeout	TimeValue	0..1	attr	Value of the N_AS timeout. Unit: seconds.
supplierId	PositiveInteger	0..1	attr	LIN Supplier ID
variantId	PositiveInteger	0..1	attr	Specifies the Variant ID

Table 3.42: LinSlave

Class	**LinErrorResponse**			
Note	Each slave node shall publish a one bit signal, named response_error, to the master node in one of its transmitted unconditional frames. The response_error signal shall be set whenever a frame (except for event triggered frame responses) that is transmitted or received by the slave node contains an error in the frame response. The response_error signal shall be cleared when the unconditional frame containing the response_error signal is successfully transmitted. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinSlave.linErrorResponse, LinSlaveConfig.linErrorResponse			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
responseError	ISignalTriggering	0..1	ref	This ISignal shall be taken to transport the responseError bit. **Tags:** atp.Splitkey=responseError

Table 3.43: LinErrorResponse

3.3.4.5 LIN Communication Connector

LinCommunicationConnector is a specialization of the CommunicationConnector class. The LinCommunicationConnector element contains lists of frames processed by the slave node.

[constr_3029] Assign-Frame command usage

Imposition time: CP: IT_SysDesc

⌈For the LIN 2.0 Assign-Frame command the LinConfigurableFrame list shall be used. For the LIN 2.1 Assign-Frame-PID-Range command the LinOrderedConfigurableFrame list shall be used.⌋

[constr_5030] Uniqueness of LinOrderedConfigurableFrame.index

Imposition time: CP: IT_SysDesc

⌈LinOrderedConfigurableFrame.index shall always be set and be unique in the context of the aggregating LinCommunicationConnector.⌋

[constr_5450] Existence of index

Imposition time: CP: IT_SysDesc

⌈For each LinOrderedConfigurableFrame, the attribute shall index shall exist.⌋

[constr_5451] Existence of LinOrderedConfigurableFrame.frame reference

Imposition time: CP: IT_SysDesc

⌈For each LinOrderedConfigurableFrame, the reference to LinFrame in the role frame shall exist.⌋

[constr_5452] Existence of LinConfigurableFrame.frame reference

Imposition time: CP: IT_SysDesc

⌈For each LinConfigurableFrame, the reference to LinFrame in the role frame shall exist.⌋

Class	LinCommunicationConnector			
Note	LIN bus specific communication connector attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurable Frame	LinConfigurableFrame	*	aggr	LinConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.0 Assign-Frame command. **Tags:** atp.Splitkey=linConfigurableFrame
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	aggr	LinOrderedConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.1 Assign-Frame-PID-Range command. **Tags:** atp.Splitkey=linOrderedConfigurableFrame
schedule ChangeNext TimeBase	Boolean	0..1	attr	This attribute defines the point in time where a schedule table switch is performed. If this attribute is set to false or not present, the schedule table shall be switched after the current entry of the active schedule table is ended. If this attribute is enabled, the schedule table shall be switched when message transmission or reception within an entry has been completed, ensured by status checks for transmission and reception.

Table 3.44: LinCommunicationConnector

Class	LinConfigurableFrame			
Note	Assignment of messageIds to Frames. This element shall be used for the LIN 2.0 Assign-Frame command. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinCommunicationConnector.linConfigurableFrame, LinSlaveConfig.linConfigurableFrame			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frame	LinFrame	0..1	ref	Reference to a Frame that is processed by the slave node. **Tags:** atp.Splitkey=frame
messageId	PositiveInteger	0..1	attr	MessageId for the referenced frame

Table 3.45: LinConfigurableFrame

Class	LinOrderedConfigurableFrame			
Note	With the assignment of the index to a frame a mapping of Pids to Frames is possible. This element shall be used for the LIN 2.1 Assign-Frame-PID-Range command. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	LinCommunicationConnector.linOrderedConfigurableFrame, LinSlaveConfig.linOrderedConfigurable Frame			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**LinOrderedConfigurableFrame**			
frame	LinFrame	0..1	ref	Reference to a Frame that is processed by the slave node. **Tags:** atp.Splitkey=frame
index	Integer	0..1	attr	This attribute is used to order the elements and allows an assignment of Pids to ConfigurableFrames that are defined in the slave.

Table 3.46: LinOrderedConfigurableFrame

3.3.4.6 LIN Physical Channel

`LinPhysicalChannel` is a specialization of the `PhysicalChannel` class. It contains additional Lin-specific `PhysicalChannel` attributes.

Class	**LinPhysicalChannel**			
Note	LIN specific attributes to the physicalChannel This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
busIdleTimeout Period	TimeValue	0..1	attr	This attribute shall be used to set an idle timeout period for the enclosing LinPhysicalChannel.
scheduleTable	LinScheduleTable	*	aggr	Schedule tables organize the timings of the frames for LIN. atpVariation: If the transmitted frames are variable, the corresponding ScheduleTables shall be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=scheduleTable, scheduleTable.shortName, scheduleTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.47: LinPhysicalChannel

[constr_3015] Number of LIN channels

Imposition time: CP: `IT_SysDesc`

⌈LIN clusters shall aggregate exactly one `LinPhysicalChannel`.⌋

3.3.5 Ethernet

The EthernetCluster represents an Ethernet network which may consist of several ECUs connected.

An essential aspect of modern Ethernet is the possibility to introduce Ethernet switches in order to partition the EthernetCluster into segments which are used for point-to-point communication between the respective partners. It is possible to define the behavior of such Ethernet switches, this is described in chapter 3.3.5.6.

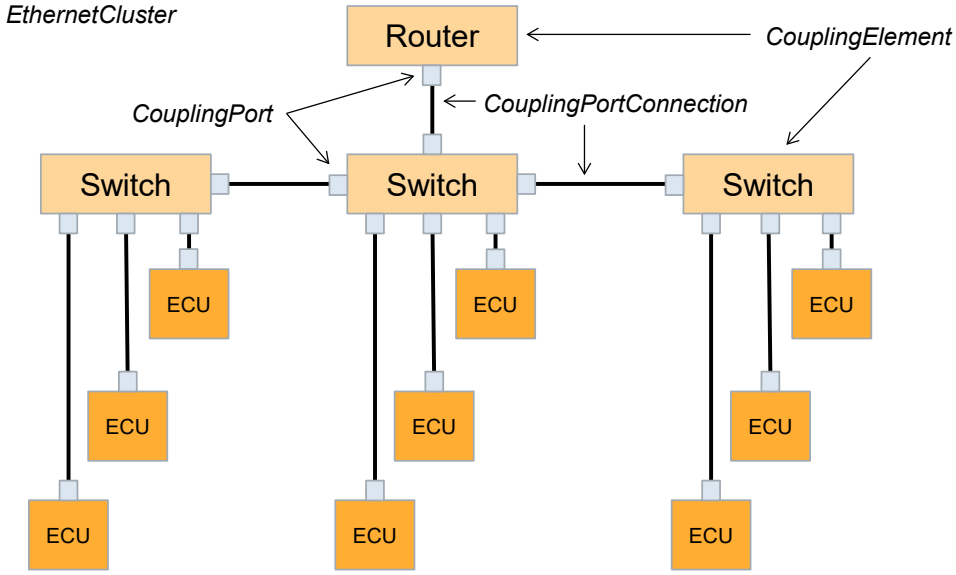

Figure 3.14: Example of an EthernetCluster

Figure 3.14 illustrates an example of an EthernetCluster. In this figure the focus is on the *Link Layer* and represents the wiring of ECUs, their communication connectors, switches, hubs, routers, and how these elements are connected electrically.

To describe the Ethernet at the data link- and physical layer the following System Template meta-model classes are used: EthernetCluster, EthernetCommunicationController, EthernetCommunicationConnector, EthernetPhysicalChannel, CouplingElement, CouplingPort and CouplingPortConnection (see Figure 3.15).

AUTOSAR supports the wake-up and sleep mechanism complying with the Open Alliance TC10 specification (OA TC10, see [14]), which is used for a switched Ethernet network. The details are described in chapter 3.3.5.8.

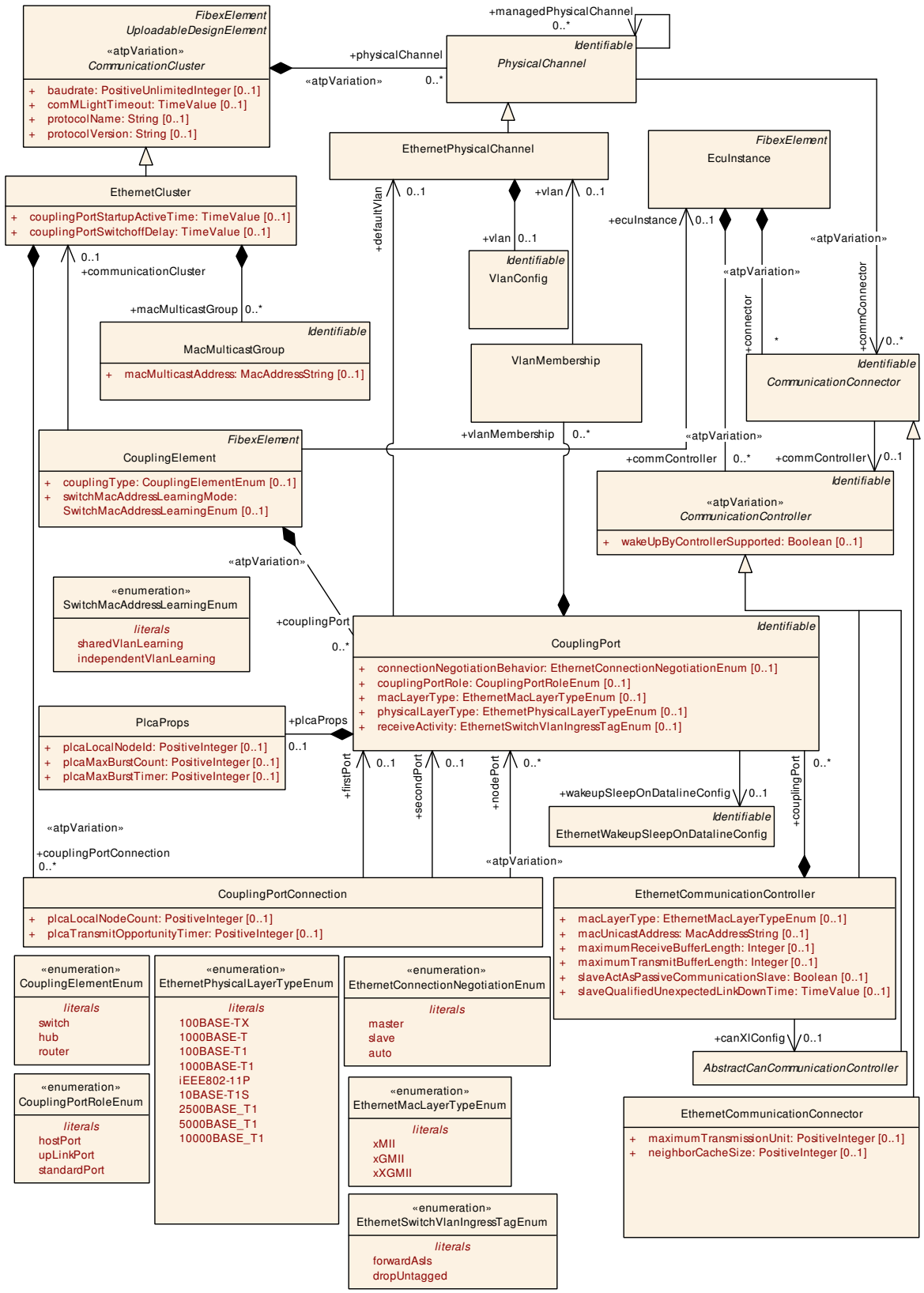

Figure 3.15: Ethernet topology elements (Fibex4Ethernet_Topology)

[constr_5251] `CouplingPort.connectionNegotiationBehavior` shall exist

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `CouplingPort.connectionNegotiationBehavior` shall be defined.⌋

3.3.5.1 Ethernet Cluster

Each `EthernetCluster` may have globally defined `MacMulticastGroup`s. `MacMulticastGroup`s have a `macMulticastAddress` (for example 01:00:5E:7F:FF:FF). One sender can handle many receivers simultaneously, if the receivers have all the same `macMulticastAddress`.

[constr_3047] Uniqueness of `macMulticastAddress`es

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `macMulticastAddress` shall be unique in a particular `EthernetCluster`.⌋

[constr_5453] Existence of `macMulticastAddress`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `MacMulticastGroup`, the attribute `macMulticastAddress` shall exist.⌋

For details on `CouplingPort` specific attributes of `EthernetCluster` in relation with Partial Networks please refer to chapter 5.5.1.1.

Class	«atpVariation» **EthernetCluster**			
Note	Ethernet-specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort Connection	CouplingPort Connection	*	aggr	Specification of connections between CouplingElements and EcuInstances. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPortConnection, couplingPort Connection.variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingPort StartupActive Time	TimeValue	0..1	attr	The attribute specifies the time in second a coupling port is switched on to enable the host ECU (ECU that maintains an Ethernet switch) to listen to the network for potential network management requests.
couplingPort SwitchoffDelay	TimeValue	0..1	attr	Switch off delay for CouplingPorts in seconds. It denotes the delay of switching off couplingPorts after the request to switch off a couplingPort was issued. (e.g. switch off of Ethernet switch ports).

▽

△

Class	«atpVariation» **EthernetCluster**			
macMulticast Group	MacMulticastGroup	*	aggr	MacMulticastGroup that is defined for the Subnet (EthernetCluster). **Tags:** atp.Splitkey=macMulticastGroup, macMulticast Group.shortName

Table 3.48: EthernetCluster

Class	**MacMulticastGroup**			
Note	Per EthernetCluster globally defined MacMulticastGroup. One sender can handle many receivers simultaneously if the receivers have all the same macMulticastAddress. The addresses need to be unique for the particular EthernetCluster.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetCluster.macMulticastGroup			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
macMulticast Address	MacAddressString	0..1	attr	A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.

Table 3.49: MacMulticastGroup

[TPS_SYST_02362] Relevance of attribute `EthernetCluster.baudrate` ⌈The value of the attribute `baudrate` in the context of an `EthernetCluster` has no meaning and shall be ignored.⌋

The communication speed is defined by the attribute `CouplingPort.physicalLayerType` of the involved `CouplingPort`s.

3.3.5.2 Ethernet Physical Channel

The `EthernetPhysicalChannel` represents a VLAN. VLANs (IEEE 802.1q) divide physical Ethernet networks in logical subnets. Their realization requires switches with VLAN support. VLANs are defined on a switch on a port-by-port basis.

The term `EthernetPhysicalChannel` may be misleading because it actually does *not* defined the physical (electrical) attributes of the communication but the `EthernetPhysicalChannel` defines the VLANs as *logical* broadcast domains in which the communication partners can interact.

Regardless whether the Ethernet communication uses tagged [TPS_SYST_01095] or untagged [TPS_SYST_01096] VLANs all communication needs to be defined within respective `EthernetPhysicalChannel`s as defined in chapter 6.1.

[TPS_SYST_01095] tagged VLANs ⌈In the System Description a VLAN is represented by an `EthernetPhysicalChannel` and is identified by its `vlanIdentifier`.⌋

[TPS_SYST_01096] **untagged VLANs** ⌈If the `VlanConfig` and the `vlanIdentifier` are not defined for an `EthernetPhysicalChannel` than the channel is called "untagged".⌋

Every `Frame` that is sent over a "tagged" VLAN is tagged with a VLAN Tag. With this tag every receiving switch has the information about the VLAN that the `Frame` belongs to. The VLAN Tag that is attached to a `Frame` contains the user priority for the `Frame` that is described with the `defaultPriority` and the `vlanIdentifier`.

Class	**EthernetPhysicalChannel**			
Note	The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comMLight Timeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine.
network Endpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan. **Tags:** atp.Splitkey=networkEndpoint, network Endpoint.shortName
soAdConfig	SoAdConfig	0..1	aggr	SoAd Configuration for one specific Physical Channel. **Tags:** atp.Splitkey=soAdConfig This Attribute is only used by the AUTOSAR Classic Platform.
vlan	VlanConfig	0..1	aggr	VLAN Configuration. **Tags:** atp.Splitkey=vlan, vlan.shortName

Table 3.50: EthernetPhysicalChannel

[constr_3333] **Standardized values for the attribute `category` of meta-class `EthernetPhysicalChannel`**

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The following values of the attribute `category` of meta-class `EthernetPhysicalChannel` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetPhysicalChannel` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetPhysicalChannel` in case of a wireless ethernet connection

⌋

[TPS_SYST_02159] **Default value for the attribute `category` of meta-class `EthernetPhysicalChannel`** ⌈The default value for the `category` of an `EthernetPhysicalChannel` shall be *WIRED*.⌋

[constr_3334] Allowed references between `EthernetPhysicalChannel` and `EthernetCommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈An `EthernetPhysicalChannel` is only allowed to reference `EthernetCommunicationConnector`s in the role `commConnector` that have the same `category` value as the referencing `EthernetPhysicalChannel`.⌋

[constr_3365] `EthernetPhysicalChannel`s with different `category` values are not allowed within an `EthernetCluster`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A mix of `EthernetPhysicalChannel`s with different `category` values within an `EthernetCluster` is currently not supported by AUTOSAR.⌋

[constr_3336] `EthernetPhysicalChannel.soAdConfig` in case of WIRELESS `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈If `EthernetPhysicalChannel` has the `category` *WIRELESS* then the `EthernetPhysicalChannel` shall not aggregate the `SoAdConfig`.⌋

[TPS_SYST_01086] Number of Ethernet channels ⌈Each `EthernetCluster` may aggregate up to 4096 `EthernetPhysicalChannel`s.⌋

Class	**VlanConfig**			
Note	VLAN Configuration attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetPhysicalChannel.vlan			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
vlanIdentifier	PositiveInteger	0..1	attr	A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.

Table 3.51: VlanConfig

[constr_3048] Range of `vlanIdentifier`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The allowed values of `vlanIdentifier` range from 0 to 4095.⌋

[constr_5454] Existence of `vlanIdentifier`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanConfig`, the attribute `vlanIdentifier` shall exist.⌋

3.3.5.2.1 VLAN Priority

The Priority is a 3-bit field which refers to the IEEE 802.1Q priority. It indicates the frame priority level. Values are from 0 (best effort) to 7 (highest); 1 represents the

lowest priority. These values can be used to prioritize different classes of traffic (voice, video, data, etc.). The priority is contained in the Ethernet Header together with the vlanIdentifier.

The defaultPriority in the context of a CouplingPort can be overwritten on different levels (higher numbers in the list indicate higher overwriting priority):

1. NetworkEndpoint ([TPS_SYST_01088])
2. ApplicationEndpoint ([TPS_SYST_01089])
3. ProvidedServiceInstance ([TPS_SYST_01108]) or ConsumedEvent-Group ([TPS_SYST_02014])

If a priority on an ApplicationEndpoint is defined the priorities in the Network-Endpoint and the defaultPriority in the VlanMembership would be ignored.

The following table shows a CouplingPort with defaultPriority equal to 0. In the first row no other priorities are defined, so the effective priority is 0. In the second row a priority of 4 is defined on the NetworkEndpoint, so the effective priority is 4. In the third row a priority of 2 is defined on the ApplicationEndpoint, so the effective priority is 2. In the last row a priority of 1 is defined on the Provided-ServiceInstance, so the effective priority is 1.

Physical Port (defaultPriority of CouplingPort)	NetworkEndpoint.priority (e.g. IP Address)	Application-Endpoint.priority (e.g. Tcp Port)	ProvidedService-Instance.priority	Effective priority
0	-	-	-	**0**
0	**4**	-	-	**4**
0	4	**2**	-	**2**
0	4	2	**1**	**1**

Table 3.52: VLAN Priority Example

3.3.5.3 Ethernet Coupling Elements and Coupling Ports

A CouplingElement is used to connect EcuInstances via CouplingPorts to EthernetPhysicalChannels (VLANs) that are defined within an Ethernet-Cluster.

CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement references the EthernetCluster and contains a collection of available CouplingPorts. The couplingType identifies the CouplingElement as a switch, hub or router.

Class	CouplingElement			
Note	A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. Coupling Elements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device. **Tags:** atp.recommendedPackage=CouplingElements			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Cluster	EthernetCluster	0..1	ref	This relationship defines to which cluster the Coupling Element belongs. **Tags:** atp.Splitkey=communicationCluster
coupling ElementDetails	CouplingElement AbstractDetails	0..1	aggr	Definition of details for this specific CouplingElement. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingElementDetails, couplingElement Details.shortName, couplingElementDetails.variation Point.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild xml.namePlural=COUPLING-ELEMENT-DETAILS This Attribute is only used by the AUTOSAR Classic Platform.
couplingPort	CouplingPort	*	aggr	Hardware Port of the CouplingElement that is used to connect this CouplingPort to EcuInstances or other CouplingElements. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPort, couplingPort.shortName, couplingPort.variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingType	CouplingElementEnum	0..1	attr	Describes the coupling type of this CouplingElement.
ecuInstance	EcuInstance	0..1	ref	Optional reference to the ECU where the Coupling Element is located. **Tags:** atp.Splitkey=ecuInstance
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of a Coupling Element. **Tags:** atp.Splitkey=firewallRule
switchMac Address LearningMode	SwitchMacAddress LearningEnum	0..1	attr	Defines the MAC address learning mode of the Ethernet switch.

Table 3.53: CouplingElement

Enumeration	CouplingElementEnum
Note	Identifies the Coupling type.
Aggregated by	CouplingElement.couplingType
Literal	**Description**
hub	A device that is used to connect segments of a LAN. In Hubs frames are "broadcasted" to every one of its ports. **Tags:** atp.EnumerationLiteralIndex=0



△

Enumeration	**CouplingElementEnum**
router	A device that routes frames between different networks. **Tags:** atp.EnumerationLiteralIndex=1
switch	A device that filters and forwards frames between different LAN segments. **Tags:** atp.EnumerationLiteralIndex=2

Table 3.54: CouplingElementEnum

[constr_5455] Existence of couplingType

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each CouplingElement, the attribute couplingType shall exist.⌋

[constr_5456] Existence of communicationCluster

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each CouplingElement, the reference communicationCluster shall exist.⌋

[constr_3062] The EcuInstance that is referenced from a specific CouplingElement shall be connected to the same EthernetCluster as the specific CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The EcuInstance referenced from a specific CouplingElement in the role ecuInstance shall be connected via the CommunicationConnector and a EthernetPhysicalChannel that refers the CommunicationConnector to the EthernetCluster referenced by the specific CouplingElement in the role communicationCluster.⌋

Class	**CouplingPort**			
Note	A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElement.couplingPort, EthernetCommunicationController.couplingPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connection Negotiation Behavior	EthernetConnection NegotiationEnum	0..1	attr	Specifies the connection negotiation of the CouplingPort.
couplingPort Details	CouplingPortDetails	0..1	aggr	Defines more details of a CouplingPort in case a more specific configuration is required. **Tags:** atp.Splitkey=couplingPortDetails
couplingPort Role	CouplingPortRoleEnum	0..1	attr	Defines the role this CouplingPort takes in the context of the CouplingElement.

▽

△

Class	**CouplingPort**			
defaultVlan	EthernetPhysical Channel	0..1	ref	The vLanIdentifier of the referenced VLAN is the Default-PVID (port VLAN ID). A Port VLAN ID is a default VLAN ID that is assigned to an access CouplingPort to designate the VLAN segment to which this port is connected. Also, if a CouplingPort has not been configured with any VLAN memberships, the virtual switch's Port VLAN ID (pvid) becomes the default VLAN ID for the ports connection. This identifier/tag is added for incoming untagged messages at the port (ingress tagging). For outgoing messages with this identifier, the tag is removed at the port (egress untagging, depending on the Vlan Membership.sendActivity). **Tags:** atp.Splitkey=defaultVlan
macAddress VlanAssignment	MacAddressVlan Membership	*	aggr	Statically defines the assignment of MAC-Multicast-Addresses, optionally together with VLANs, to this CouplingPort. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=macAddressVlanAssignment, macAddress VlanAssignment.shortName, macAddressVlan Assignment.variationPoint.shortLabel vh.latestBindingTime=postBuild
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the CouplingPort.
macSecProps	MacSecProps	*	aggr	Properties to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:** atp.Splitkey=macSecProps atp.Status=candidate
physicalLayer Type	EthernetPhysicalLayer TypeEnum	0..1	attr	Specifies the physical layer type of the CouplingPort.
plcaProps	PlcaProps	0..1	aggr	Optional properties for configuration of PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the Coupling Port (PHY). **Tags:** atp.Splitkey=plcaProps
pncMapping	PncMappingIdent	*	ref	Reference to the partial networks this CouplingPort participates in. **Tags:** atp.Splitkey=pncMapping
receiveActivity	EthernetSwitchVlan IngressTagEnum	0..1	attr	Defines the handling of frames at the ingress port.
vlan Membership	VlanMembership	*	aggr	Messages of VLANs that are defined here can be communicated via the CouplingPort. **Tags:** atp.Splitkey=vlanMembership
wakeupSleep OnDataline Config	EthernetWakeupSleep OnDatalineConfig	0..1	ref	Optional reference to EthernetWakeupSleepOnDataline Config. **Tags:** atp.Splitkey=wakeupSleepOnDatalineConfig

Table 3.55: CouplingPort

[constr_3726] Upper multiplicity of aggregation in the role `CouplingPort.macSecProps`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈In the context of `CouplingPort`, the aggregation in the role `macSecProps` shall exist at most once.⌋

Frame preemption is a features specified in IEEE Std 802.1Q-2022 [15] standard.

[TPS_SYST_03127] Support for `CouplingPortDetails.framePreemptionSupport` ⌈The attribute `CouplingPortDetails.framePreemptionSupport` defines whether frame preemption according to IEEE Std 802.1Q-2022 [15] shall be supported by the `CouplingPort` owning this `CouplingPortDetails`.⌋

Frame preemption is operating on a peer-to-peer communication connection level. Both communication partners need to have the same configuration for the frame preemption support.

[constr_3782] Consistent `framePreemptionSupport` setting in the scope of one `CouplingPortConnection`

Imposition time: CP: `IT_SysDesc`

⌈For each `CouplingPortConnection` the value of `CouplingPort.couplingPortDetails.framePreemptionSupport` shall be identical for both, `CouplingPortConnection.firstPort` and `CouplingPortConnection.secondPort`.⌋

Enumeration	**EthernetConnectionNegotiationEnum**
Note	Specifies connection negotiation types of Ethernet transceiver links.
Aggregated by	CouplingPort.connectionNegotiationBehavior
Literal	***Description***
auto	Automatic Negotiation **Tags:** atp.EnumerationLiteralIndex=0
master	Master **Tags:** atp.EnumerationLiteralIndex=1
slave	Slave **Tags:** atp.EnumerationLiteralIndex=2

Table 3.56: EthernetConnectionNegotiationEnum

Enumeration	**EthernetMacLayerTypeEnum**
Note	Specifies MAC (Media Access Control) Layer types.
Aggregated by	CouplingPort.macLayerType, EthernetCommunicationController.macLayerType
Literal	***Description***

▽

△

Enumeration	**EthernetMacLayerTypeEnum**
xGMII	Mac layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII) **Tags:** atp.EnumerationLiteralIndex=1 xml.name=XG-MII
xMII	Mac layer interface (data) bandwith class 100Mbit/s and 10Mbit/s (e.g. RMII, RvMII, SMII, RvMII) **Tags:** atp.EnumerationLiteralIndex=0 xml.name=X-MII
xXGMII	Mac layer interface (data) bandwith class 10Gbit/s **Tags:** atp.EnumerationLiteralIndex=2 xml.name=XXG-MII

Table 3.57: EthernetMacLayerTypeEnum

Enumeration	**EthernetPhysicalLayerTypeEnum**
Note	Specifies physical layer types of Ethernet transceiver links.
Aggregated by	CouplingPort.physicalLayerType
Literal	***Description***
_10000BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 10Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=13 xml.name=10000BASE-T1
_1000BASE_T	Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs. **Tags:** atp.EnumerationLiteralIndex=6 xml.name=1000BASE-T
_1000BASE_T1	Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=8 xml.name=1000BASE-T1
_100BASE_T1	Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100 BASE-T1 is the IEEE Standardized version of BroadRReach. **Tags:** atp.EnumerationLiteralIndex=7 xml.name=100BASE-T1
_100BASE_TX	Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs. **Tags:** atp.EnumerationLiteralIndex=5 xml.name=100BASE-TX
_10BASE_T1S	Physical layer interface 10BASE-T1S (10Mbit/s, 2 pairs). Used for automotive. **Tags:** atp.EnumerationLiteralIndex=10 atp.Status=draft xml.name=10BASE-T1S
_2500BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 2.5Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=11 xml.name=2500BASE-T1

▽

△

Enumeration	**EthernetPhysicalLayerTypeEnum**
_5000BASE_T1	Ethernet Standard (IEEE 802.3ch) to support 5Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=12 xml.name=5000BASE-T1
iEEE802_11P	Ethernet Standard (IEEE 802.11p) to support wireless communication in vehicular environments. **Tags:** atp.EnumerationLiteralIndex=9 xml.name=IEEE802-11P

Table 3.58: EthernetPhysicalLayerTypeEnum

Enumeration	**EthernetSwitchVlanIngressTagEnum**
Note	Defines the possible tagging behavior at an ingress port.
Aggregated by	CouplingPort.receiveActivity
Literal	*Description*
dropUntagged	Drop if untagged. **Tags:** atp.EnumerationLiteralIndex=1
forwardAsIs	Forward with the same VLAN as received. Also untagged frames will be forwarded as untagged. **Tags:** atp.EnumerationLiteralIndex=0

Table 3.59: EthernetSwitchVlanIngressTagEnum

Class	**VlanMembership**			
Note	Static logical channel or VLAN binding to a switch-port. The reference to an EthernetPhysicalChannel without a VLAN defined represents the handling of untagged frames.			
Base	*ARObject*			
Aggregated by	CouplingPort.vlanMembership			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
defaultPriority	PositiveInteger	0..1	attr	Standard output-priority outgoing Frames will be tagged with. Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed. In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.
dhcpAddress Assignment	DhcpServer Configuration	0..1	aggr	Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddress Assignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server. **Tags:** atp.Splitkey=dhcpAddressAssignment
sendActivity	EthernetSwitchVlan EgressTaggingEnum	0..1	attr	Attribute denotes whether a VLAN tagged ethernet frame will be 1. sent with its VLAN tag (sentTagged) 2. sent without a VLAN tag (sentUntagged) 3. will be dropped at this port (notSent or VLAN not member of this list)

▽

△

Class	VlanMembership			
vlan	EthernetPhysical Channel	0..1	ref	References a channel that represents a VLAN or an untagged channel. **Tags:** atp.Splitkey=vlan

Table 3.60: VlanMembership

[constr_5457] Existence of `defaultPriority`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanMembership`, the attribute `defaultPriority` shall exist.⌋

[constr_5458] Existence of `vlan`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `VlanMembership`, the reference `vlan` shall exist.⌋

Class	CouplingPortConnection			
Note	Connection between two CouplingPorts (firstPort and secondPort) or between a collection of Ports that are all referenced by the portCollection reference.			
Base	*ARObject*			
Aggregated by	EthernetCluster.couplingPortConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstPort	CouplingPort	0..1	ref	Reference to the first CouplingPort that is connected via the CouplingPortConnection. **Tags:** atp.Splitkey=firstPort
nodePort	CouplingPort	*	ref	Reference to a number of CouplingPorts that are connected via the CouplingPortConnection. This reference shall be used to describe a 10BASE-T1S topology architecture where several CouplingPorts of EthernetCommunicationControllers are connected via one CouplingPortConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nodePort, nodePort.couplingPort, node Port.variationPoint.shortLabel vh.latestBindingTime=postBuild
plcaLocalNode Count	PositiveInteger	0..1	attr	Defines the number of communication participants in case 10BASE-T1S and the nodePort reference is used.
plcaTransmit Opportunity Timer	PositiveInteger	0..1	attr	Timer for the transmission in bit time to evaluate if a Transmission Opportunity is yield or not.
secondPort	CouplingPort	0..1	ref	Reference to the second CouplingPort that is connected via the CouplingPortConnection. **Tags:** atp.Splitkey=secondPort

Table 3.61: CouplingPortConnection

`CouplingPort`s are hardware ports of `CouplingElement`s and `EcuInstance`s. Connections between `CouplingPort`s are realized through `CouplingPort-Connection`s.

Optionally the CouplingPort of a CouplingElement may also have one or several VlanMemberships, a defaultVlan reference and a reference to a MacMulticastGroup.

[constr_3521] defaultVlan and vlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort refers to an EthernetPhysicalChannel in the role defaultVlan the CouplingPort shall also have a vlanMembership defined. This VlanMembership shall point to the same EthernetPhysicalChannel in the role vlan as the defaultVlan.⌋

[constr_3133] physicalLayerType of connected CouplingPorts

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The physicalLayerType of two CouplingPorts which are connected via a CouplingPortConnection shall be equal.⌋

[constr_3134] The connection of two CouplingPorts with connectionNegotiationBehavior set to master is forbidden

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The connectionNegotiationBehavior of two CouplingPorts which are connected via a CouplingPortConnection shall not be both set to master.⌋

[constr_3135] The connection of two CouplingPorts with connectionNegotiationBehavior set to slave is forbidden

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The connectionNegotiationBehavior of two CouplingPorts which are connected via a CouplingPortConnection shall not be both set to slave.⌋

[TPS_SYST_01097] Assignment of CouplingPorts to a VLAN ⌈CouplingPorts of CouplingElements can be assigned to VLANs (EthernetPhysicalChannels) with the vlanMembership aggregation.⌋

[TPS_SYST_01098] Assignment of CouplingPorts to an "untagged" VLAN ⌈A CouplingPort may be assigned to several VLANs, but only one of those assignments can be "untagged".⌋

[constr_3534] EthernetPhysicalChannel shall only be referenced by one VlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈An EthernetPhysicalChannel shall only be referenced by one VlanMembership in the role VlanMembership.vlan in the scope of one CouplingPort.⌋

Figure 3.16 shows a CouplingElement with two CouplingPorts.

In this example Port 0 is assigned to three VLANs and one “untagged” EthernetPhysicalChannel. VLAN3 is marked as the defaultVlan. With the combination of the defaultVlan and the VlanMembership to the “untagged” EthernetPhysicalChannel the Frames that are transmitted over Port 0 on VLAN3 are “untagged” on the wire in both directions (Tx and Rx). The switch adds the tag for incoming untagged messages at the port (ingress tagging) and for outgoing messages the tag is removed at the port (egress untagging).

Port 1 is assigned to three VLANs. But the VlanMembership to the “untagged” EthernetPhysicalChannel is not defined here. For this reason, Frames that are transmitted over Port 1 on VLAN3 are “tagged”.

If a defaultVlan is defined for a CouplingPort but the defaultVlan is not referenced by the VlanMembership then “untagged” Frames can be received via the CouplingPort. But a response can not be send back.

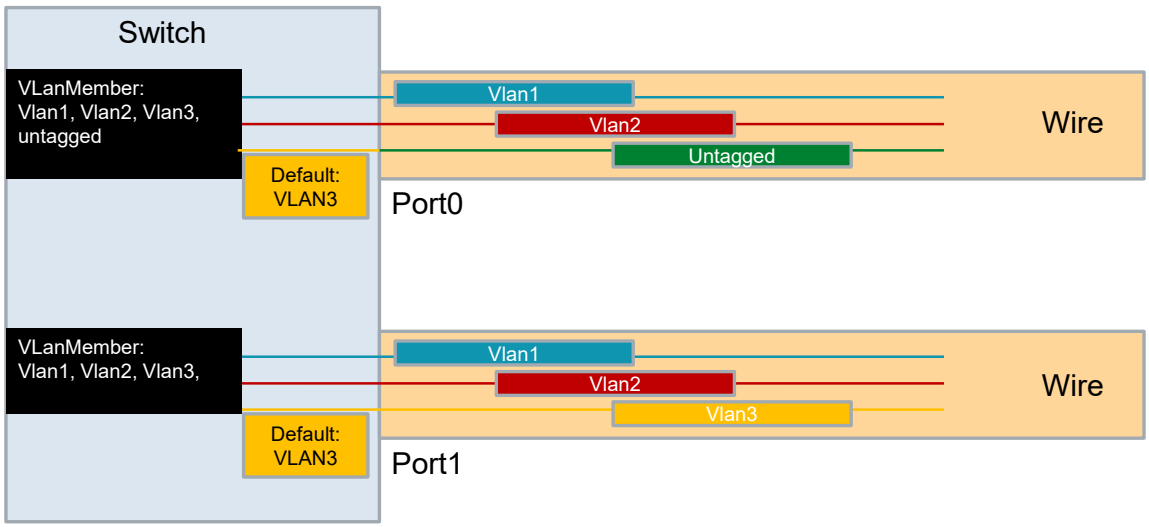

Figure 3.16: Default Vlan Example

3.3.5.4 Ethernet Communication Controller

EthernetCommunicationController is a specialization of the CommunicationController class. It contains the specific Ethernet controller attributes needed for configuring an EcuInstance connected to a certain Ethernet cluster.

Class	«atpVariation» **EthernetCommunicationController**			
Note	Ethernet specific communication port attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	«atpVariation» **EthernetCommunicationController**			
canXlConfig	AbstractCan Communication Controller	0..1	ref	If the Ethernet frames handled by this Ethernet CommunicationController are to be tunneled through CAN XL, then this reference shall refer to the Abstract CanCommunicationController that aggregates the Can ControllerXlConfiguration of the physical CAN XL channel to be used for tunneling. **Tags:** atp.Splitkey=canXlConfig
couplingPort	CouplingPort	*	aggr	Optional CouplingPort that can be used to connect the ECU to a CouplingElement (e.g. a switch). **Tags:** atp.Splitkey=couplingPort, couplingPort.shortName
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the ethernet controller.
macUnicast Address	MacAddressString	0..1	attr	Media Access Control address (MAC address) that uniquely identifies each EthernetCommunication Controller in the network.
maximum ReceiveBuffer Length	Integer	0..1	attr	Determines the maximum receive buffer length (frame length) in bytes.
maximum TransmitBuffer Length	Integer	0..1	attr	Determines the maximum transmit buffer length (frame length) in bytes.
slaveActAs Passive Communication Slave	Boolean	0..1	attr	This attribute specifies if the EcuInstance is acting as a passive communication slave on the connected Physical Channel. This is used for EthernetCommunication Controllers that use Ethernet hardware which supports wake-up and sleep on the network (e.g. Open Alliance TC10 compliant Ethernet hardware).
slaveQualified UnexpectedLink DownTime	TimeValue	0..1	attr	This attribute specifies time when an unexpected link down is evaluated as link down and indicated to the AUTOSAR communication stack.

Table 3.62: EthernetCommunicationController

[constr_3535] `EthernetCommunicationController` shall aggregate at most one `CouplingPort`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈An `EthernetCommunicationController` is allowed to aggregate at most one `CouplingPort`.⌋

[constr_3332] Standardized values for the attribute `category` of meta-class `EthernetCommunicationController`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The following values of the attribute `category` of meta-class `EthernetCommunicationController` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationController` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationController` in case of a wireless ethernet connection
- CAN_XL: This represents the tunneling of Ethernet frames handled by the `EthernetCommunicationController` through CAN XL.

⌋

[TPS_SYST_02158] Default value for the attribute `category` of meta-class `EthernetCommunicationController` ⌈The default value for the `category` of an `EthernetCommunicationController` shall be *WIRED*.⌋

The `EthernetCommunicationController` has the additional information of a `macUnicastAddress`. This is a globally unique MAC-address for the `CommunicationController`.

[constr_3702] Relevant attributes of EthernetCommunicationController for CAN XL

Imposition time: CP: `IT_SysDesc`

⌈If the category of `EthernetCommunicationController` is equal to `CAN_XL`, then only the following attributes of this meta-class are relevant:

- `macLayerType`
- `macUnicastAddress`

⌋

[constr_3703] Reference to CanControllerXlConfiguration in case of category CAN_XL

Imposition time: CP: `IT_SysDesc`

⌈If the category of `EthernetCommunicationController` is equal to `CAN_XL`, then the reference `canXlConfig` of `EthernetCommunicationController` shall refer to the `CanCommunicationController` aggregating the `CanController-Configuration` which in turn aggregates the `CanControllerXlConfiguration` that is used for tunneling of the Ethernet frames associated with the aforementioned `EthernetCommunicationController`.⌋

3.3.5.5 Ethernet Communication Connector

`EthernetCommunicationConnector` adds the Ethernet specific attributes to the `CommunicationConnector`.

Class	**EthernetCommunicationConnector**			
Note	Ethernet specific attributes to the CommunicationConnector.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	EthernetCommunicationConnector			
ethIpProps	EthIpProps	0..1	ref	EcuInstance specific IP attributes. **Tags:** atp.Splitkey=ethIpProps This Attribute is only used by the AUTOSAR Classic Platform.
maximum Transmission Unit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
neighborCache Size	PositiveInteger	0..1	attr	This attribute specifies the size of neighbor cache or ARP table in units of entries.

Table 3.63: EthernetCommunicationConnector

[constr_3331] Standardized values for the attribute `category` of meta-class `EthernetCommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The following values of the attribute `category` of meta-class `EthernetCommunicationConnector` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationConnector` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationConnector` in case of a wireless ethernet connection
- CAN_XL: This represents the tunneling of Ethernet frames handled by the `EthernetCommunicationConnector` through CAN XL.

⌋

[TPS_SYST_02157] Default value for the attribute `category` of meta-class `EthernetCommunicationConnector` ⌈The default value for the `category` of an `EthernetCommunicationConnector` shall be *WIRED*.⌋

[constr_3335] Allowed references between `EthernetCommunicationConnector` and `EthernetCommunicationController`

Imposition time: CP: `IT_SysDesc`

⌈An `EthernetCommunicationConnector` is only allowed to reference an `EthernetCommunicationController` in the role `commController` that has the same `category` value as the referencing `EthernetCommunicationConnector`.⌋

3.3.5.6 Ethernet Switch Driver

Ethernet networks in an automotive environment consist basically of ECUs with a single port PHY and switch ECUs with several ports. Different to consumer networks, where switches are typically stand-alone devices, switches in automotive networks may be integrated and connected to a CPU via MII and other interfaces. The configuration of these switches does influence the communication behavior within the network.

3.3.5.6.1 Ethernet switch port structure

In order to describe switched Ethernet networks it is essential to describe some parts of an Ethernet switch. Examples are scheduling and forwarding mechanisms within a switch as well as the switch structure within its ports.

As shown in figure 3.17, the switch consists of a certain number of ports. Each port has its own set of egress FIFOs in which the incoming packets are buffered. How the messages in the FIFOs will be forwarded depends mainly on the shaping and port scheduling mechanisms. Thus, the parametrization of the egress port influences the latency of messages within the network.

Please note that the egress port structures in figure 3.17 are meant as an example. Other structures with different FIFO numbers are possible as well.

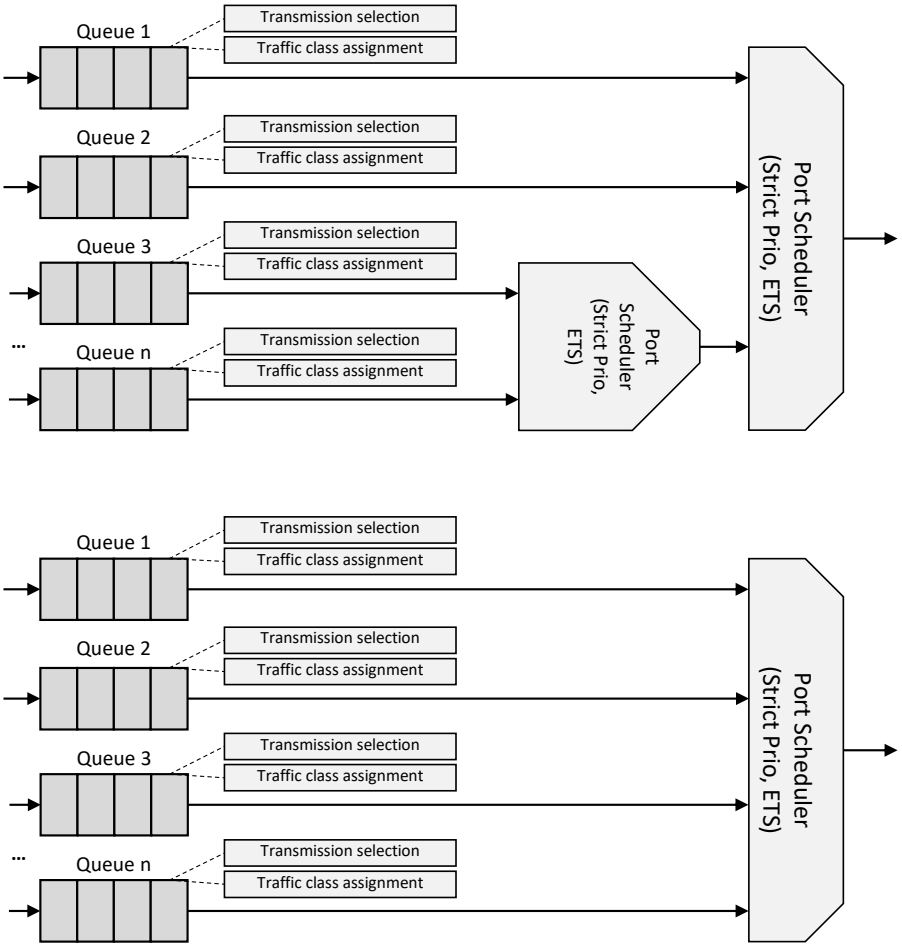

Figure 3.17: Example egress switch port configurations

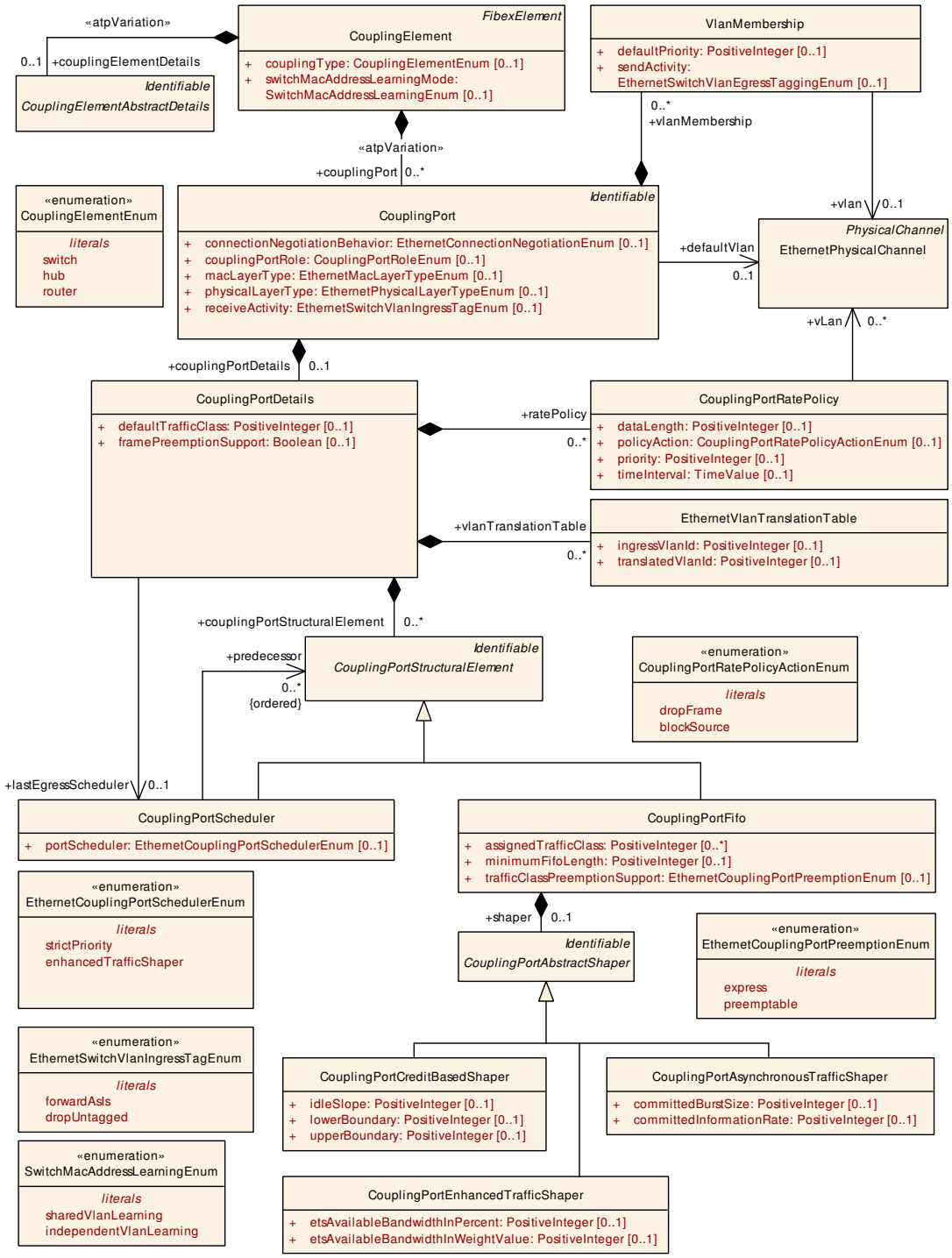

Figure 3.18: Egress switch port structure

The structural description of an Ethernet switch is based on the already existing `CouplingElement` in the System Template. Each `CouplingElement` can already have a set of `CouplingPort`s.

In case a detailed Switch configuration is required, there is the configuration option to add to the `CouplingPort` a `CouplingPortDetails` element which encapsulates the structural description of one switch port.

The elements which one switch port consists of are (egress side):

- CouplingPortFifo
- CouplingPortScheduler.

The model allows to collect the egress parts of one switch port in the CouplingPortDetails.couplingPortStructuralElements.

[TPS_SYST_03006] Ethernet switch egress port setup ⌈Two setups can be defined at an egress port of a switch:

- The switch port has only one Fifo:
  - the CouplingPortFifo element is aggregated at the CouplingPortDetails.couplingPortStructuralElements
  - no CouplingPortDetails.lastEgressScheduler is defined.
- The switch port has at least one scheduler
  - the various switch port elements are all aggregated at the CouplingPortDetails.couplingPortStructuralElements
  - the CouplingPortScheduler which is the last scheduler in a chain of structural elements is additionally referenced in the role CouplingPortDetails.lastEgressScheduler

⌋

The modeling approach is based on a predecessor chain model where the chain is started by the last scheduler in the switch port and defines where the input to this scheduler comes from. The input to a scheduler can come from several predecessor elements which might be

- another CouplingPortScheduler
- a CouplingPortFifo.

Frame preemption is a features specified in IEEE Std 802.1Q-2022 [15] standard. Also for CouplingPorts used in the context of an Ethernet switch the specification of [TPS_SYST_03127] and [constr_3782] applies.

Additionally a CouplingPortFifo (i.e. queue) has the possibility to define whether frame preemption is enabled for that specific CouplingPortFifo using the attribute CouplingPortFifo.trafficClassPreemptionSupport. Since the CouplingPortFifo is associated with exactly one traffic class as defined by assignedTrafficClass, the trafficClassPreemptionSupport is implicitly associated with exactly one traffic class as well.

[constr_3783] Definition of CouplingPortFifo.trafficClassPreemption-Support only in context of an Ethernet switch

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort is aggregated by a CouplingElement with CouplingElement.couplingType equal to CouplingElementEnum.switch and CouplingPort.couplingPortDetails.framePreemptionSupport is set to true, then the attribute CouplingPortFifo.trafficClassPreemptionSupport shall be defined.⌋

[TPS_SYST_03128] Traffic class specific enabling of egress frame preemption ⌈The attribute CouplingPortFifo.trafficClassPreemptionSupport defines whether frames handled by this CouplingPortFifo shall be

- EthernetCouplingPortPreemptionEnum.preemptable or have the setting
- EthernetCouplingPortPreemptionEnum.express, which makes then non preemptable

according to IEEE Std 802.1Q-2022 [15].⌋

Enumeration	**EthernetCouplingPortPreemptionEnum**
Note	Defines whether frames assigned to the traffic class associated with this CouplingPortFifo may be preempted or not.
Aggregated by	CouplingPortFifo.trafficClassPreemptionSupport
Literal	***Description***
express	Frames assigned to the traffic class associated with this CouplingPortFifo will never be preempted. **Tags:** atp.EnumerationLiteralIndex=0
preemptable	Frames assigned to the traffic class associated with this CouplingPortFifo may be preempted. **Tags:** atp.EnumerationLiteralIndex=1

Table 3.64: EthernetCouplingPortPreemptionEnum

[TPS_SYST_03007] Ethernet port scheduler algorithm ⌈The scheduler performs a prioritization of the frames based on the algorithm defined in the CouplingPort-Scheduler.portScheduler.⌋

[TPS_SYST_03008] Ethernet port scheduler priority ⌈The first element in CouplingPortScheduler.predecessor has the highest priority.Therefore, it is important to have the predecessor definition of the scheduler ordered.⌋

A CouplingPortFifo defines the way how that FIFO is processed via the CouplingPortFifo.shaper definition.

[constr_3784] Applicable `CouplingPortFifo` as `predecessor` for `portScheduler` = `enhancedTrafficShaper`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPortScheduler` with `CouplingPortScheduler.portScheduler` equals to `EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper` shall only refer to `CouplingPortFifo` in the role `CouplingPortScheduler.predecessor` where the `CouplingPortFifo` has a `CouplingPortFifo.shaper` of kind `CouplingPortEnhancedTrafficShaper`.⌋

[constr_3785] Exclusive definition of `etsAvailableBandwidthInPercent` or `etsAvailableBandwidthInWeightValue`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPortEnhancedTrafficShaper` shall either define an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent` or an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue` value, but not both.⌋

[constr_3786] Consistent usage of either `etsAvailableBandwidthInPercent` or `etsAvailableBandwidthInWeightValue` for `portScheduler` = `enhancedTrafficShaper`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈All the `CouplingPortFifo` referenced by the same `CouplingPortScheduler` with `CouplingPortScheduler.portScheduler` equals to `EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper` (according to [constr_3784]) shall define in their `CouplingPortEnhancedTrafficShaper` the same kind of value. All shall use either an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent` or an `CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue` value.⌋

Class	**CouplingPortRatePolicy**			
Note	Defines a rate policy on a CouplingPort. **Tags:** atp.Status=obsolete			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.ratePolicy			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	Amount of data in bytes (excluding header information) that can be received to define the rate policy. **Tags:** atp.Status=obsolete
policyAction	CouplingPortRatePolicyActionEnum	0..1	attr	Defines the action to be performed when this rate policy is violated. **Tags:** atp.Status=obsolete

▽

△

Class	**CouplingPortRatePolicy**			
priority	PositiveInteger	0..1	attr	Defines the priority which this rate policy shall be limited on. If no priority is given this rate policy is not considering priority. **Tags:** atp.Status=obsolete
timeInterval	TimeValue	0..1	attr	Time interval used to define the base of the rate policy. **Tags:** atp.Status=obsolete
vLan	EthernetPhysical Channel	*	ref	Defines the VLANs this rate policy shall be limited on. If no VLAN is given this rate policy is not considering VLAN tags. **Tags:** atp.Splitkey=vLan atp.Status=obsolete

Table 3.65: CouplingPortRatePolicy

Enumeration	**CouplingPortRatePolicyActionEnum**
Note	Defines the action to be performed when a rate policy is violated. **Tags:** atp.Status=obsolete
Aggregated by	CouplingPortRatePolicy.policyAction
Literal	***Description***
blockSource	If the rate policy is violated the CouplingPort this CouplingPortRatePolicy is defined on shall block all frames from the MAC-Address the violation was caused by. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=obsolete
dropFrame	If the rate policy is violated the frame shall be dropped. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=obsolete

Table 3.66: CouplingPortRatePolicyActionEnum

3.3.5.6.1.1 Ethernet switch Credit-Based Shaper (CBS)

The functionality of the credit-based shaper (CBS) (`CouplingPortCreditBasedShaper`) is configured using the attribute `idleSlope`.

[TPS_SYST_03009] Ethernet switch `CouplingPortCreditBasedShaper`

Status: DRAFT

⌈The `idleSlope` is defined in the IEEE802.1Qav standard [15] as a parameter for an increase of credit in bits per second. The `idleSlope` can never exceed the maximal transmit rate of a port, e.g. 100MBits for BroadR-Reach and 1GBits for RTPGE. The `idleSlope` determines the maximum fraction of the port transmit rate that is available for the queue associated with the shaper: bandwidthFraction = idleSlope/portTransmitRate.⌋

Lower and upper boundaries of the CBS credit are not configurable. If the credit counter of an implementation should over- or under-run its upper or lower limit at runtime, the

resulting behavior is not specified by AUTOSAR or by IEEE and remains implementation specific.

3.3.5.6.1.2 Ethernet switch Asynchronous Traffic Shaper (ATS)

The functionality of the asynchronous traffic shaper (ATS) is defined for a `CouplingPortFifo` using the `CouplingPortAsynchronousTrafficShaper`. Further configuration at the level of `SwitchStreamIdentification` is contributing to the ATS behavior.

Class	**CouplingPortDetails**			
Note	Defines details of a CouplingPort. May be used to configure the structures of a switch.			
Base	*ARObject*			
Aggregated by	CouplingPort.couplingPortDetails			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort Structural Element	CouplingPortStructural Element	*	aggr	Collects all the structural parts at which a CouplingPort may be configurable. **Tags:** atp.Splitkey=couplingPortStructuralElement, couplingPortStructuralElement.shortName
defaultTraffic Class	PositiveInteger	0..1	attr	Defines the default traffic class for this CouplingPort.
ethernetPriority Regeneration	EthernetPriority Regeneration	0..8	aggr	Defines a priority regeneration where the ingress priority is replaced by regenerated priority. **Tags:** atp.Splitkey=ethernetPriorityRegeneration, ethernetPriorityRegeneration.shortName
ethernetTraffic Class Assignment	CouplingPortTraffic ClassAssignment	*	aggr	Defines the priority to traffic class assignment. **Tags:** atp.Splitkey=ethernetTrafficClassAssignment, ethernetTrafficClassAssignment.shortName
frame Preemption Support	Boolean	0..1	attr	Defines whether frames handled by this CouplingPort may be preemted.
globalTime Props	GlobalTimeCoupling PortProps	0..1	aggr	Specifies properties for the usage of the CouplingPort in the scope of Global Time Sync. **Tags:** atp.Splitkey=globalTimeProps
lastEgress Scheduler	CouplingPortScheduler	0..1	ref	Defines which CouplingPortScheduler is the last in the egress port structure. **Tags:** atp.Splitkey=lastEgressScheduler
ratePolicy	CouplingPortRatePolicy	*	aggr	Rate policies to be applied for this CouplingPort. **Tags:** atp.Splitkey=ratePolicy atp.Status=obsolete
vlanTranslation Table	EthernetVlanTranslation Table	*	aggr	Definition of entries that define the ingress Vlan translation between IngressVlanID and TranslatedVlanID. **Tags:** atp.Splitkey=vlanTranslationTable

Table 3.67: CouplingPortDetails

Class	**EthernetVlanTranslationTable**			
Note	This element defines one ingress Vlan translation entry in which the IngressVlanID from the incoming frame is replaced by the TranslatedVlanID.			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.vlanTranslationTable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ingressVlanId	PositiveInteger	0..1	attr	Incoming VlanID from received frame
translatedVlanId	PositiveInteger	0..1	attr	Mapped VlanID after ingress Vlan translation

Table 3.68: EthernetVlanTranslationTable

[constr_9346] Existence of EthernetVlanTranslationTable.translatedVlanId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthernetVlanTranslationTable, the attribute translatedVlanId shall exist.⌋

[constr_9347] Range of EthernetVlanTranslationTable.ingressVlanId and EthernetVlanTranslationTable.translatedVlanId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of ingressVlanId and translatedVlanId shall be in the range 0..4095.⌋

[TPS_SYST_02418] Definition of wildcard EthernetVlanTranslationTable

⌈If the EthernetVlanTranslationTable defines only the attribute translatedVlanId without the ingressVlanId attribute then a wildcard entry is configured by this EthernetVlanTranslationTable and all Vlan Ids not matching any configured ingressVlanId value are mapped to the translatedVlanId.⌋

[constr_9348] EthernetVlanTranslationTable.translatedVlanId and vlanMembership

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort defines an EthernetVlanTranslationTable via the CouplingPortDetails then the CouplingPort shall have a vlanMembership defined that references an EthernetPhysicalChannel that has the same vlanIdentifier value defined as the translatedVlanId value in the EthernetVlanTranslationTable.⌋

Class	***CouplingPortStructuralElement*** (abstract)			
Note	General class to define structural elements a CouplingPort may consist of.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CouplingPortFifo, CouplingPortScheduler			
Aggregated by	CouplingPortDetails.couplingPortStructuralElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.69: CouplingPortStructuralElement

Class	**CouplingPortScheduler**			
Note	Defines a scheduler for the CouplingPort egress structure.			
Base	*ARObject*, *CouplingPortStructuralElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortDetails.couplingPortStructuralElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
portScheduler	EthernetCouplingPort SchedulerEnum	0..1	attr	Defines the schedule algorithm to be used.
predecessor (ordered)	CouplingPortStructural Element	*	ref	Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.

Table 3.70: CouplingPortScheduler

Enumeration	**EthernetCouplingPortSchedulerEnum**
Note	Defines the schedule algorithm to be used.
Aggregated by	CouplingPortScheduler.portScheduler
Literal	***Description***
enhancedTraffic Shaper	Scheduler used for enhanced traffic shaping (e.g. weighted round robin) **Tags:** atp.EnumerationLiteralIndex=3
strictPriority	Schedule algorithm "strict priority" **Tags:** atp.EnumerationLiteralIndex=1

Table 3.71: EthernetCouplingPortSchedulerEnum

Class	**CouplingPortFifo**			
Note	Defines a FIFO for the CouplingPort egress structure.			
Base	*ARObject*, *CouplingPortStructuralElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortDetails.couplingPortStructuralElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignedTraffic Class	PositiveInteger	*	attr	Defines a set of Traffic Classes which shall be handled by this FIFO.
minimumFifo Length	PositiveInteger	0..1	attr	FIFO minimum length in Byte. An actual configuration/ hardware may use a bigger value.
shaper	CouplingPortAbstract Shaper	0..1	aggr	Definition of the shaper to be used for the processing of this FIFO. **Tags:** atp.Splitkey=shaper, shaper.shortName atp.Status=candidate
trafficClass Preemption Support	EthernetCouplingPort PreemptionEnum	0..1	attr	Defines whether frames assigned to the traffic class associated with this CouplingPortFifo may be preempted or not.

Table 3.72: CouplingPortFifo

Class	***CouplingPortAbstractShaper*** (abstract)
Note	Abstract class for the definition of coupling port shapers. **Tags:** atp.Status=candidate
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	**CouplingPortAbstractShaper** (abstract)			
Subclasses	CouplingPortAsynchronousTrafficShaper, CouplingPortCreditBasedShaper, CouplingPortEnhanced TrafficShaper			
Aggregated by	CouplingPortFifo.shaper			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 3.73: CouplingPortAbstractShaper

Class	**CouplingPortCreditBasedShaper**			
Note	Defines a Credit Based Shaper (CBS) for the CouplingPort egress structure. **Tags:** atp.Status=candidate			
Base	*ARObject*, *CouplingPortAbstractShaper*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortFifo.shaper			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
idleSlope	PositiveInteger	0..1	attr	Defines the increase of credit in bits per second for the CBS shaper. **Tags:** atp.Status=candidate
lowerBoundary	PositiveInteger	0..1	attr	Defines the lower boundary of credit for the CBS shaper. **Tags:** atp.Status=obsolete
upperBoundary	PositiveInteger	0..1	attr	Defines the upper boundary of credit for the CBS shaper. **Tags:** atp.Status=obsolete

Table 3.74: CouplingPortCreditBasedShaper

Class	**CouplingPortAsynchronousTrafficShaper**			
Note	Defines an Asynchronous Traffic Shaper (ATS) for the CouplingPort egress structure. **Tags:** atp.Status=candidate			
Base	*ARObject*, *CouplingPortAbstractShaper*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortFifo.shaper			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
committedBurst Size	PositiveInteger	0..1	attr	Maximum token capacity of the token bucket in bit. **Tags:** atp.Status=obsolete
committed InformationRate	PositiveInteger	0..1	attr	Defines the rate at which the token bucket is refilled with tokens in bit per second. **Tags:** atp.Status=obsolete
trafficShaper Group	SwitchAsynchronous TrafficShaperGroup Entry	0..1	ref	Reference to the Traffic Shaper Group this Asynchronous Traffic Shaper is part of. **Tags:** atp.Splitkey=trafficShaperGroup atp.Status=obsolete This Attribute is only used by the AUTOSAR Classic Platform.

Table 3.75: CouplingPortAsynchronousTrafficShaper

Class	**CouplingPortEnhancedTrafficShaper**			
Note	Defines a scheduler used for enhanced traffic shaping (e.g. weighted round robin). **Tags:** atp.Status=candidate			
Base	*ARObject*, *CouplingPortAbstractShaper*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPortFifo.shaper			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
etsAvailable BandwidthIn Percent	PositiveInteger	0..1	attr	Defines the available bandwidth in percent of an enhanced transmission selection algorithm (ETS). **Tags:** atp.Status=candidate
etsAvailable BandwidthIn WeightValue	PositiveInteger	0..1	attr	Defines the available bandwidth as weight value of an enhanced transmission selection algorithm (ETS). **Tags:** atp.Status=candidate

Table 3.76: CouplingPortEnhancedTrafficShaper

3.3.5.6.2 Ethernet packet forwarding

Besides the modeling of egress ports, it is necessary to specify how incoming packets are forwarded to the egress ports. For this purpose, different assignment policies of packets to egress port FIFOs are implemented in switches.

As an example, the Ethernet priority field can be evaluated and remapped into an regenerated priority: Within the VLAN-tag, the PCP-field (priority code point) is a parameter which can be modified at an ingress port of an Ethernet switch. For this purpose a priority regeneration table can be defined.

The `CouplingPortDetails.ethernetPriorityRegeneration` is optional in case the feature of priority regeneration is not be used.

[TPS_SYST_03003] Ethernet priority regeneration ⌈The `CouplingPortDetails.ethernetPriorityRegeneration` specifies which `ingressPriority` is mapped to which `regeneratedPriority`.⌋

[constr_3515] Fully filled `EthernetPriorityRegeneration` table

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈In case the `CouplingPortDetails.ethernetPriorityRegeneration` is defined it shall contain exactly 8 elements of `EthernetPriorityRegeneration`, one for each value of `ingressPriority` (0-7).⌋

The (potentially remapped) Ethernet priority field can be evaluated and mapped to a traffic class. Such a traffic class is again mapped to an egress FIFO. Other header information of the Ethernet frame can be also used for the assignment of Ethernet frames to egress FIFOs. For the mapping to a certain traffic class, the following tables are necessary.

PORT-based Mapping	Traffic Class
Port2, Port3, Port4	7
Port1	6
–	5
–	4
–	3
–	2
–	1
–	0

Table 3.77: Port to Traffic Class mapping

PCP-based Mapping	Traffic Class
Prio 0	7
Prio 1	6
Prio 2-7	5
–	4
–	3
–	2
–	1
–	0

Table 3.78: PCP-field to Traffic Class mapping

While the first table shows the mapping of ingress-ports to traffic classes, the second table shows the priority-based mapping which can be defined per ingress port. Both tables are in conflict with each other, i.e. it has to be decided which mapping is applied.

Also the mapping of a traffic class to a FIFO shall be done on a per port basis. An example is shown in the following table.

Traffic Class	FIFO (if 4 FIFOs available)
7	3
6	2
0-5	1
–	0

Table 3.79: Traffic Class to FIFO mapping

[TPS_SYST_03010] Ethernet switch packet to traffic class assignment ⌈The aggregation `ethernetTrafficClassAssignment` defines a PCP-field (priority) to traffic class mapping. A set of `priority`s is mapped to one `trafficClass`.⌋

The priority based mapping of [TPS_SYST_03010] does not have to be complete. There may not be a `CouplingPortTrafficClassAssignment` defined for a set of priorities, or there may not even be any `CouplingPortTrafficClassAssignment` defined. For the case that a `CouplingPortTrafficClassAssignment` for a dedicated priority is not given by any `CouplingPortTrafficClassAssignment` at

a CouplingPortDetails, the defaultTrafficClass takes effect as defined in [TPS_SYST_03129].

[constr_5049] Ethernet switch packet to traffic class assignment restriction

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈Every CouplingPortTrafficClassAssignment shall have at least one priority attribute defined.⌋

[constr_3787] Existence of CouplingPortTrafficClassAssignment.trafficClass

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈Every CouplingPortTrafficClassAssignment shall have a trafficClass attribute defined.⌋

[TPS_SYST_03011] Ethernet switch traffic class to FIFO assignment ⌈The traffic class is used to assign a frame to a switch egress FIFO (queue). The CouplingPortFifo.assignedTrafficClass defines which traffic class this FIFO (queue) is assigned. Only frames with that assignedTrafficClass value will be assigned to this FIFO.⌋

[constr_3788] Existence of CouplingPortFifo.assignedTrafficClass

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each CouplingPortFifo, exactly one value of attribute CouplingPortFifo.assignedTrafficClass shall exist.⌋

Note that CouplingPortFifo.assignedTrafficClass is modelled with an upper multiplicity of * due to backward compatibility reasons. [constr_3788] allows only one value.

[constr_3789] Allowed values for CouplingPortFifo.assignedTrafficClass

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The allowed values for CouplingPortFifo.assignedTrafficClass are 0..65535.⌋

[TPS_SYST_03129] Definition of a defaultTrafficClass per CouplingPort ⌈The attribute CouplingPortDetails.defaultTrafficClass defines a default traffic class for the CouplingPort which owns the CouplingPortDetails. The defaultTrafficClass is applied when a frame is processed at the CouplingPort and there is no CouplingPortTrafficClassAssignment.priority defined for the frame's priority in the CouplingPortDetails.ethernetTrafficClassAssignment.⌋

[constr_3790] Existence of CouplingPortDetails.defaultTrafficClass

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each CouplingPortDetails, the attribute CouplingPortDetails.defaultTrafficClass shall exist.⌋

[constr_3791] Allowed values for CouplingPortDetails.defaultTrafficClass

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The allowed values for CouplingPortDetails.defaultTrafficClass are 0..65535.⌋

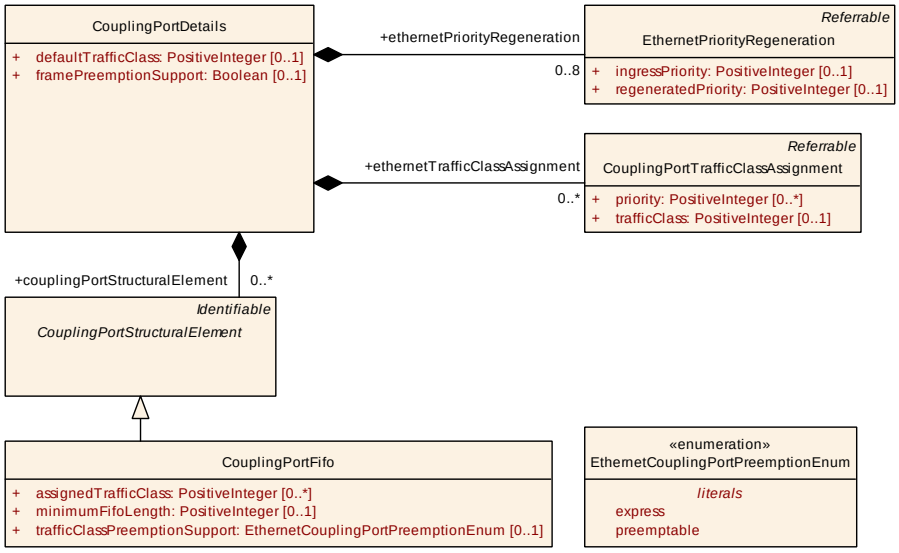

Figure 3.19: Ethernet Priority Regeneration and Ethernet Traffic Class Assignment

Class	**EthernetPriorityRegeneration**			
Note	Defines a priority regeneration where the ingressPriority is replaced by regeneratedPriority. The ethernetPriorityRegeneration is optional in case no priority regeneration shall be performed. In case a ethernetPriorityRegeneration is defined it shall have 8 mappings, one for each priority.			
Base	*ARObject*, *Referrable*			
Aggregated by	CouplingPortDetails.ethernetPriorityRegeneration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ingressPriority	PositiveInteger	0..1	attr	Message priority of the incoming message. range: 0-7
regenerated Priority	PositiveInteger	0..1	attr	Regenerated message priority. range: 0-7

Table 3.80: EthernetPriorityRegeneration

[constr_5462] Existence of ingressPriority

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthernetPriorityRegeneration, the attribute ingressPriority shall exist.⌋

[constr_5463] Existence of `regeneratedPriority`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `EthernetPriorityRegeneration`, the attribute `regeneratedPriority` shall exist.⌋

Class	**CouplingPortTrafficClassAssignment**			
Note	Defines the assignment of Traffic Class to a frame.			
Base	*ARObject*, *Referrable*			
Aggregated by	CouplingPortDetails.ethernetTrafficClassAssignment			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
priority	PositiveInteger	*	attr	Defines a priority which is mapped onto a Traffic Class.
trafficClass	PositiveInteger	0..1	attr	Defines the Traffic Class which is assigned.

Table 3.81: CouplingPortTrafficClassAssignment

[constr_5464] Existence of `trafficClass`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `CouplingPortTrafficClassAssignment`, the attribute `trafficClass` shall exist.⌋

3.3.5.6.3 Ethernet VLAN Configuration

For each VLAN identifier a table is necessary which stores at which egress port the corresponding VLAN is tagged or untagged. For an 8-port switch, this table could look like the following example where T stands for tagging and U for untagging:

	Port number							
VLAN-ID	1	2	3	4	5	6	7	8
1	T	T	–	U	–	–	–	T
2	T	U	–	T	–	–	–	T
...								
4094								

Table 3.82: VLAN Forwarding table

Incoming packets which contain a VLAN-ID of e.g. 1 can be forwarded to the ports 1, 2, 4, and 8. At ports 1, 2, and 8 these packets will be transmitted with the VLAN tag and at port 4 the tag will be removed. If a broadcast message with e.g. VLAN-ID 2 will be received at port 2 it will be forwarded to port 1, 4, and 8. The other ports 3, 5, 6, and 7 are not in the same VLAN. Thus, the packet will not be forwarded to these egress ports. The table considers only messages which contain a VLAN-ID within the switch.

`CouplingPort.vlanMembership` defines specific attributes to the behavior a packet with a specific VLAN-ID shall have on this `CouplingPort`.

[TPS_SYST_03004] VLAN specific sending behavior ⌈The `VlanMembership.sendActivity` defines for a `CouplingPort` and VLAN the sending behavior:

- `sentTagged`: packet is sent at this `CouplingPort` with the defined VLAN-ID
- `sentUntagged`: packet is sent at this `CouplingPort` but the VLAN-ID is removed before sending
- `notSent`: packet is not sent at this `CouplingPort`

⌋

Another table specifies a port-based modification of the VLAN-ID or an insertion of the VLAN-ID into the Ethernet message:

Port number	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
VLAN-Id	2	–	–	6	–	–	–	–

Table 3.83: Ingress VLAN Modification/Insertion Table

In this example, all incoming messages at port one will get the VLAN-Id 2 no matter whether they already had one before. At port 4, all incoming messages will get a 6 as their VLAN-Id. At the remaining ports, no VLAN-Ids will be inserted and an existing VLAN-Id in the Ethernet-message will remain without modification.

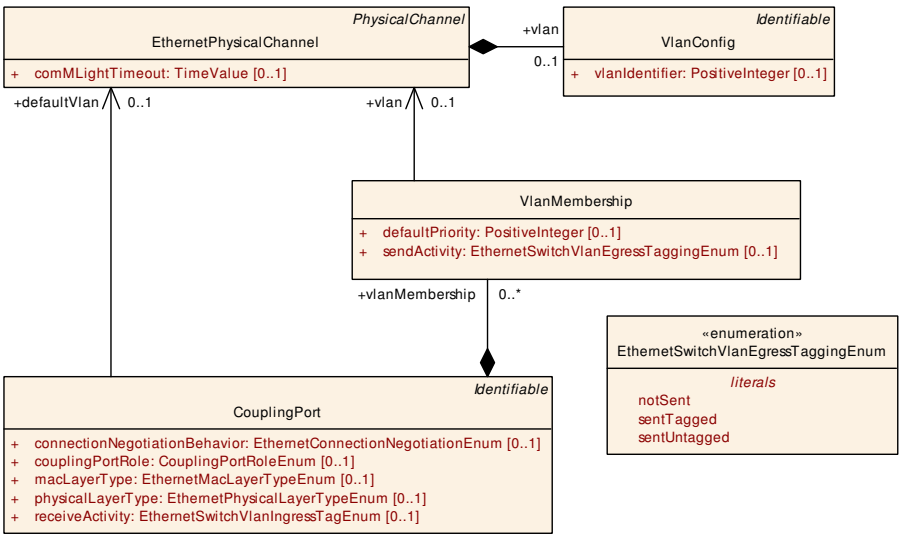

Figure 3.20: VLAN Modification

Enumeration	**EthernetSwitchVlanEgressTaggingEnum**
Note	Defines the VLAN tag sending behavior.
Aggregated by	VlanMembership.sendActivity
Literal	***Description***

▽

△

Enumeration	**EthernetSwitchVlanEgressTaggingEnum**
notSent	will not be sent **Tags:** atp.EnumerationLiteralIndex=0
sentTagged	sent with its VLAN tag **Tags:** atp.EnumerationLiteralIndex=1
sentUntagged	sent without a VLAN tag **Tags:** atp.EnumerationLiteralIndex=2

Table 3.84: EthernetSwitchVlanEgressTaggingEnum

Ethernet switches (according to [15]) support two kinds of MAC address learning approaches, defined in `CouplingElement.switchMacAddressLearningMode`

- SVL (shared VLAN learning)
- IVL (independent VLAN learning).

[TPS_SYST_03119] Definition of MAC address learning mode for an Ethernet switch ⌈The attribute `CouplingElement.switchMacAddressLearningMode` defines which MAC address learning mode an Ethernet switch uses. If not defined, then `SwitchMacAddressLearningEnum.sharedVlanLearning` is used.⌋

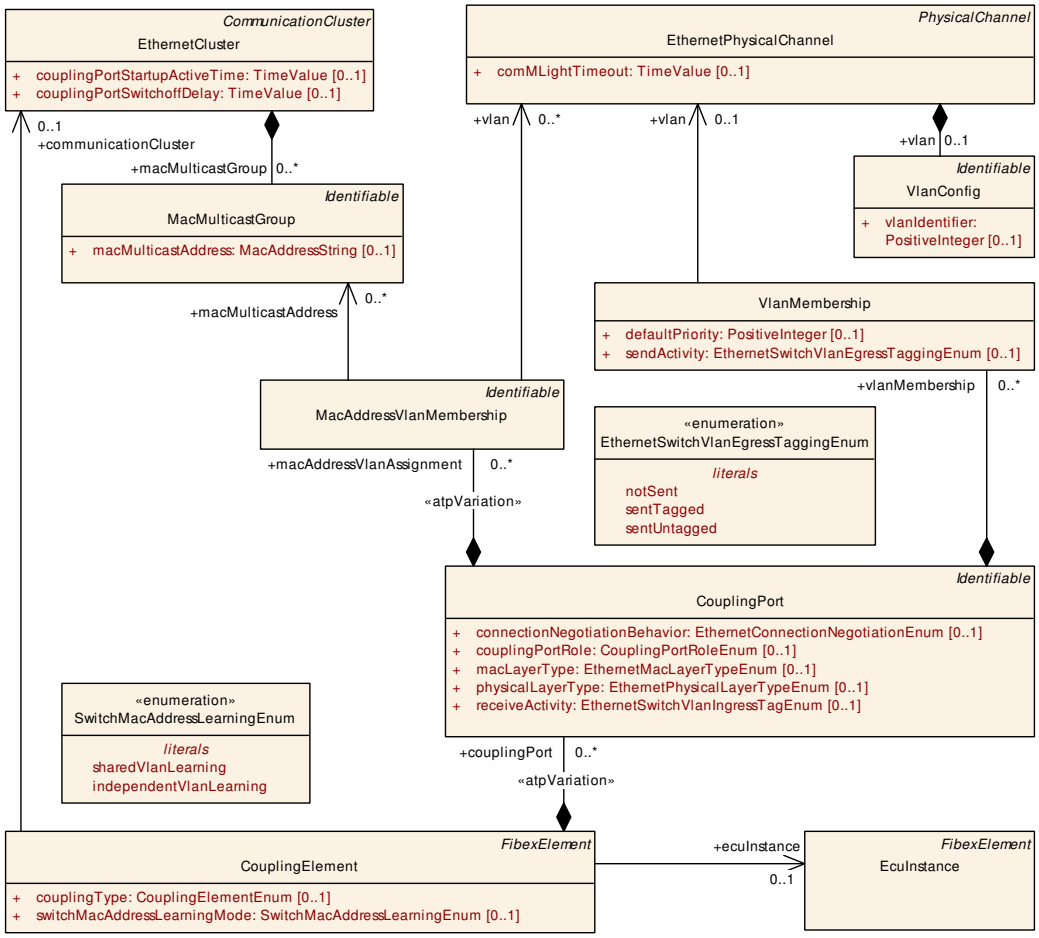

Figure 3.21: VLAN Configuration

Enumeration	**SwitchMacAddressLearningEnum**
Note	Defines the MAC address learning mode.
Aggregated by	CouplingElement.switchMacAddressLearningMode
Literal	***Description***
independentVlan Learning	Defines the Independent Vlan Learning (IVL) mode. **Tags:** atp.EnumerationLiteralIndex=1
sharedVlan Learning	Defines the Shared Vlan Learning (SVL) mode. **Tags:** atp.EnumerationLiteralIndex=0

Table 3.85: SwitchMacAddressLearningEnum

In order to pre-configure MAC multicast addresses associated with an Ethernet switch port configuration the CouplingPort.macAddressVlanAssignment is used. This assignment references one or more MacMulticastGroups in the role MacAddressVlanMembership.macMulticastAddress.

Additionally, if CouplingElement.switchMacAddressLearningMode = SwitchMacAddressLearningEnum.independentVlanLearning is defined, then the MacAddressVlanMembership is also able to reference one or more EthernetPhysicalChannels in the role MacAddressVlanMembership.vlan, stating that the referenced MacAddressVlanMembership.macMulticastAddress are to be considered in the referenced MacAddressVlanMembership.vlan only.

Class	**MacAddressVlanMembership**			
Note	Assigns a set of MAC-Multicast-Addresses which are addressable via the CouplingPort aggregating this MacAddressVlanMembership. Optionally also assigns a set of VLANs to this relation. This is a static pre-configuration and further addresses may be learned during runtime.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingPort.macAddressVlanAssignment			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
macMulticast Address	MacMulticastGroup	*	ref	Defines a set of macMulticastAddresses to be mapped to the CouplingPort. **Tags:** atp.Splitkey=macMulticastAddress
vlan	EthernetPhysical Channel	*	ref	Defines a set of VLANs the set of macMulticastAddress apply to. **Tags:** atp.Splitkey=vlan

Table 3.86: MacAddressVlanMembership

[TPS_SYST_03120] Definition of MAC multicast addresses in case of SwitchMacAddressLearningEnum.sharedVlanLearning ⌈If a CouplingPort is owned by a CouplingElement with switchMacAddressLearningMode = sharedVlanLearning (or switchMacAddressLearningMode not defined), then the CouplingPort.macAddressVlanAssignment defines a set of MacMulticastGroups which shall be pre-configured at that CouplingPort.⌋

[TPS_SYST_03121] Definition of MAC multicast addresses in combination with VLANs in case of `SwitchMacAddressLearningEnum.independentVlanLearning` ⌈If a `CouplingPort` is owned by a `CouplingElement` with `switchMacAddressLearningMode` = `independentVlanLearning`, then the `CouplingPort.macAddressVlanAssignment` defines a set of `MacMulticastGroup`s and `EthernetPhysicalChannel`s (representing VLANs) which shall be pre-configured at that `CouplingPort`.⌋

[constr_3764] Applicability of `CouplingPort.macAddressVlanAssignment`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The aggregation `CouplingPort.macAddressVlanAssignment` shall only exist if the `CouplingPort` is aggregated by a `CouplingElement` with `couplingType` = `CouplingElementEnum.switch`.⌋

[constr_3765] Applicability of `MacAddressVlanMembership.vlan`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The reference `MacAddressVlanMembership.vlan` shall only exist if the `CouplingPort` aggregating this `MacAddressVlanMembership` is aggregated by a `CouplingElement` with `switchMacAddressLearningMode` = `SwitchMacAddressLearningEnum.independentVlanLearning`.⌋

[constr_3766] Valid `MacAddressVlanMembership.vlan` target `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If an `EthernetPhysicalChannel` is referenced by a `CouplingPort.macAddressVlanAssignment.vlan`, then that `EthernetPhysicalChannel` shall also be referenced by the same `CouplingPort` via `CouplingPort.vlanMembership.vlan`⌋

3.3.5.6.4 Semi-static DHCP server configuration

The ECU which manages the Ethernet switch may run a semi-static DHCP server.

[TPS_SYST_03013] Semi-static DHCP server configuration ⌈In order to be able to assign always the same IP-address to a dedicated DHCP client, the DHCP server needs the information at which switch port the DHCP request with the specific MAC address has been received. With this switch port information the DHCP server will assign the IP-address according to the `VlanMembership.dhcpAddressAssignment`.

This allows the assignment of MAC addresses by the Tier 1 and assignment of IP addresses by the OEM. With this mechanism it is also possible to assign different IP addresses to several VLANs at the same port.⌋

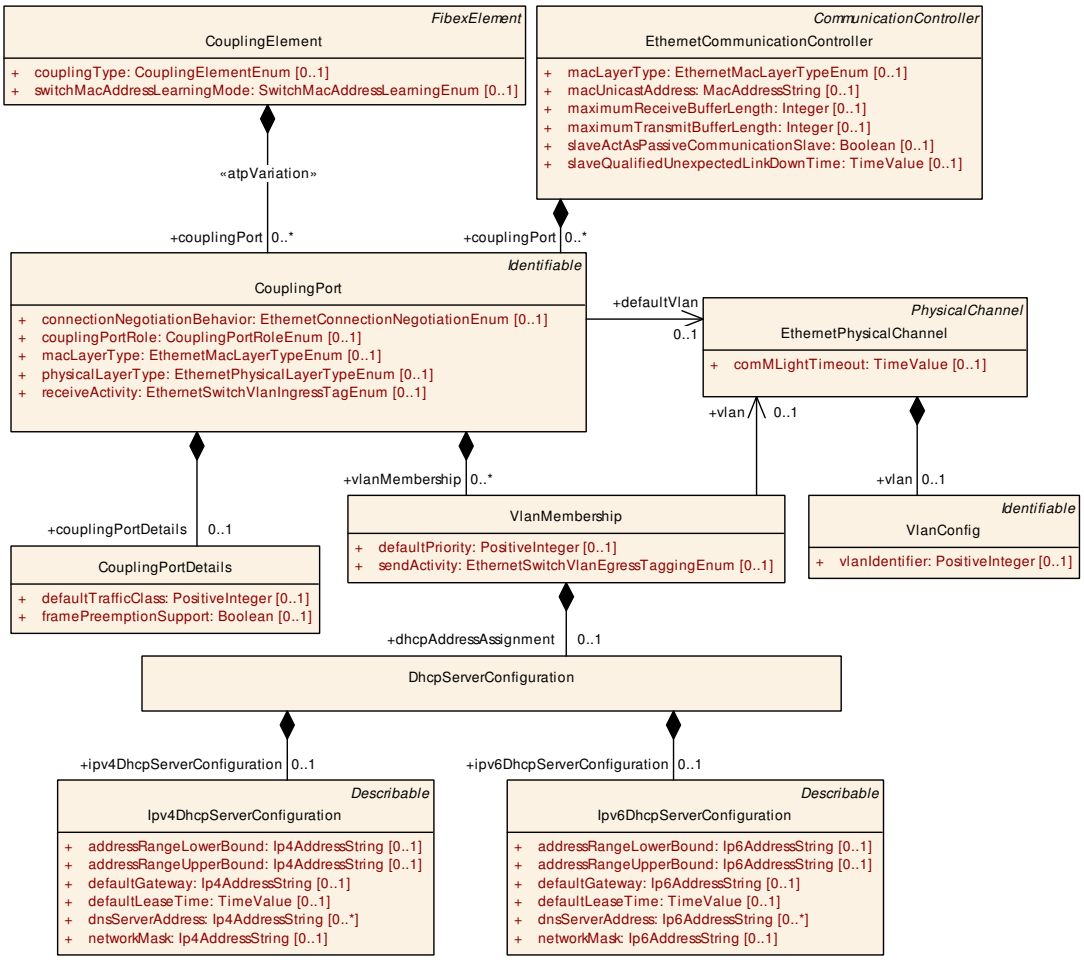

Figure 3.22: Semi-static DHCP configuration

Class	**DhcpServerConfiguration**			
Note	Defines the configuration of DHCP servers that are running on the network endpoint. It is possible that an Ipv4DhcpServer and an Ipv6DhcpServer run on the same Ecu.			
Base	*ARObject*			
Aggregated by	VlanMembership.dhcpAddressAssignment			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ipv4DhcpServer Configuration	Ipv4DhcpServer Configuration	0..1	aggr	Configuration of a IPv4 DHCP server that runs on the network endpoint. **Tags:** atp.Splitkey=ipv4DhcpServerConfiguration
ipv6DhcpServer Configuration	Ipv6DhcpServer Configuration	0..1	aggr	Configuration of a IPv6 DHCP server that runs on the network endpoint. **Tags:** atp.Splitkey=ipv6DhcpServerConfiguration

Table 3.87: DhcpServerConfiguration

Class	**Ipv4DhcpServerConfiguration**			
Note	Defines the configuration of a IPv4 DHCP server that runs on the network endpoint.			
Base	*ARObject*, *Describable*			
Aggregated by	DhcpServerConfiguration.ipv4DhcpServerConfiguration			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
addressRange LowerBound	Ip4AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv4 Address. Notation: 255.255.255.255.
addressRange UpperBound	Ip4AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. Pv4 Address. Notation: 255.255.255.255.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLease Time	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation 255.255.255.255 **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip4AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 3.88: Ipv4DhcpServerConfiguration

Class	**Ipv6DhcpServerConfiguration**			
Note	Defines the configuration of a IPv6 DHCP server that runs on the network endpoint.			
Base	*ARObject*, *Describable*			
Aggregated by	DhcpServerConfiguration.ipv6DhcpServerConfiguration			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
addressRange LowerBound	Ip6AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.
addressRange UpperBound	Ip6AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.
defaultGateway	Ip6AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLease Time	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation: FFFF:...:FFFF. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip6AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 3.89: Ipv6DhcpServerConfiguration

3.3.5.6.5 Stream Identification and per-stream filtering and policing

A stream identification and per-stream filtering and policing may be configured to be performed by a switch. For this purpose the `CouplingElement` may aggregate a `CouplingElement.couplingElementDetails`.

Note: the term "stream" is used in this chapter according to the definition in IEEE Std 802.1Q-2022 [15]. It is equivalent to the term "frame" (i.e. Ethernet frame).

The specialization for switch details is defined using `CouplingElementSwitch-Details`.

[constr_3762] Usage of CouplingElementSwitchDetails only on an Ethernet switch

Status: DRAFT
Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingElement aggregates a CouplingElementSwitchDetails in the role CouplingElement.couplingElementDetails, then that CouplingElement shall have the attribute couplingType set to the value CouplingElementEnum.switch.⌋

Class	***CouplingElementAbstractDetails*** (abstract)			
Note	Collection of specific details for the CouplingElement. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CouplingElementSwitchDetails			
Aggregated by	CouplingElement.couplingElementDetails			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.90: CouplingElementAbstractDetails

Class	**CouplingElementSwitchDetails**			
Note	Collection of specific details for the CouplingElement of couplingType switch. **Tags:** atp.Status=candidate atp.recommendedPackage=SwitchStreamIdentificationTables This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CouplingElementAbstractDetails*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElement.couplingElementDetails			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
atsInstance Entry	SwitchAtsInstanceEntry	*	aggr	Collection of ATS Instance Entries. **Tags:** atp.Splitkey=atsInstanceEntry, atsInstanceEntry.short Name atp.Status=candidate
flowMetering	SwitchFlowMetering Entry	*	aggr	Collection of Flow Metering Entries. **Tags:** atp.Splitkey=flowMetering, flowMetering.shortName atp.Status=candidate
streamFilter (ordered)	SwitchStreamFilterEntry	*	aggr	Collection of Stream Filter Entries. **Tags:** atp.Status=candidate
streamGate	SwitchStreamGateEntry	*	aggr	Collection of Stream Gate Entries. **Tags:** atp.Splitkey=streamGate, streamGate.shortName atp.Status=candidate
switchStream Identification (ordered)	SwitchStream Identification	*	aggr	Collection of switch stream identification entries. **Tags:** atp.Status=candidate

▽

△

Class	**CouplingElementSwitchDetails**			
trafficShaper Group	SwitchAsynchronous TrafficShaperGroup Entry	*	aggr	Collection of Traffic Shaper Groups. **Tags:** atp.Splitkey=trafficShaperGroup, trafficShaper Group.shortName atp.Status=candidate

Table 3.91: CouplingElementSwitchDetails

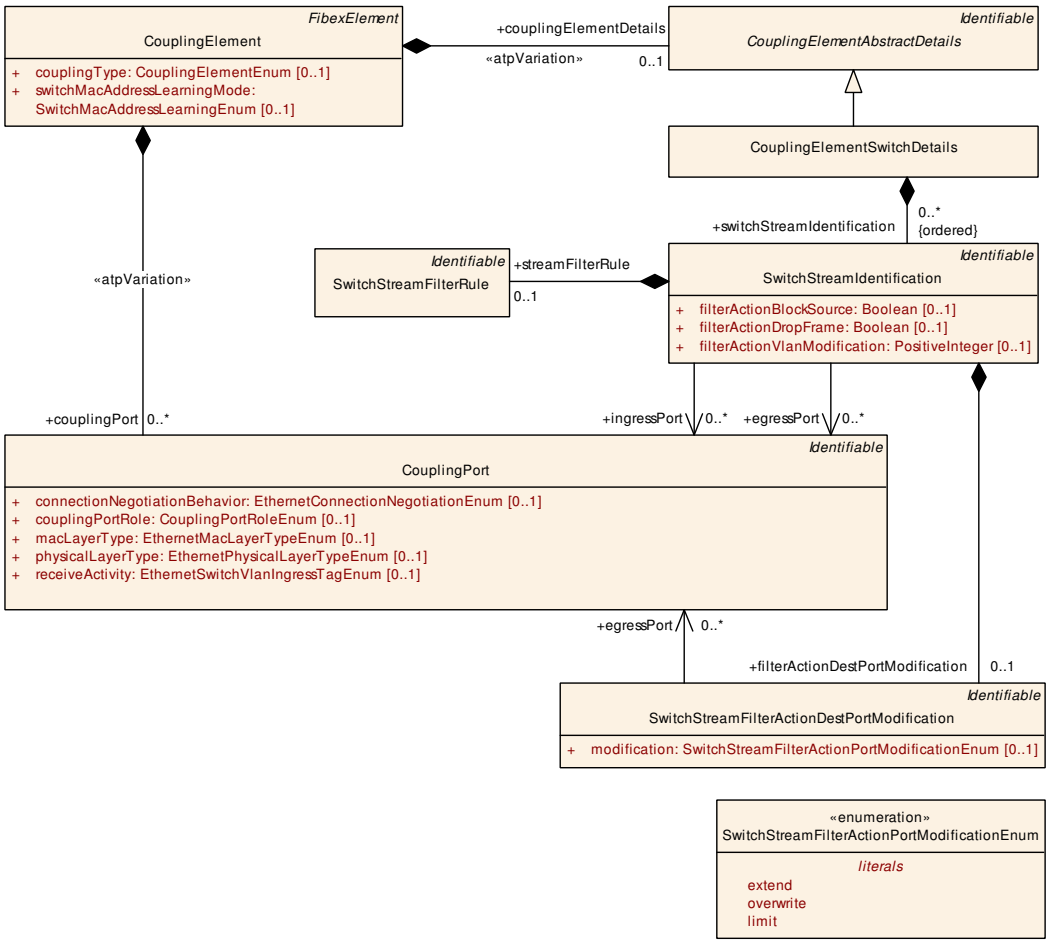

Figure 3.23: Ethernet `SwitchStreamIdentification`

The list of `CouplingElementSwitchDetails.switchStreamIdentification` elements is `ordered`, thus it is essential in which order the `SwitchStream-Identification` entries are defined. The earlier entries in the list are matched before later entries in the list. Thus, if a frame would match several `SwitchStream-Identification` entries, only the first match from the list is getting effective. Later matches are not considered.

The `SwitchStreamIdentification` applies to streams in the Ethernet switch. For the `SwitchStreamIdentification` one set of criteria is on which `CouplingPort` a frame has been received and to which `CouplingPort`(s) this frame will be forwarded to.

[TPS_SYST_03112] ingressPort definition for SwitchStreamIdentification

Status: DRAFT

⌈If a SwitchStreamIdentification references one or more CouplingPorts in the role ingressPort, then a frame is considered matching for this ingress check if that frame is received over any of the referenced CouplingPorts in the role ingressPort.

If a SwitchStreamIdentification references no CouplingPorts in the role ingressPort, then a frame is considered matching for this ingress check regardless over which CouplingPort that frame has been received.⌋

[TPS_SYST_03113] egressPort definition for SwitchStreamIdentification

Status: DRAFT

⌈If a SwitchStreamIdentification references one or more CouplingPorts in the role egressPort, then a frame is considered matching for this egress check if that frame is forwarded to all of the referenced CouplingPorts in the role egressPort.

If a SwitchStreamIdentification references no CouplingPorts in the role egressPort, then a frame is considered matching for this egress check regardless to which CouplingPorts that frame is forwarded.⌋

For further behavioral aspects refer to the SWS Switch Driver document [16].

Class	**SwitchStreamIdentification**			
Note	SwitchStreamIdentification **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElementSwitchDetails.switchStreamIdentification			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
egressPort	CouplingPort	*	ref	Reference to the CouplingPort to be taken into account as the egress role for this SwitchStreamIdentification. **Tags:** atp.Splitkey=egressPort atp.Status=candidate
filterActionBlock Source	Boolean	0..1	attr	Enables Blocking all frames from the MAC address. **Tags:** atp.Status=candidate
filterActionDest PortModification	SwitchStreamFilter ActionDestPort Modification	0..1	aggr	Defines the action to modify the destination port(s) determined by the frame forwarding process for an particular Ethernet frame. **Tags:** atp.Splitkey=filterActionDestPortModification, filterAction DestPortModification.shortName atp.Status=candidate
filterActionDrop Frame	Boolean	0..1	attr	Enables Drop Frame action. **Tags:** atp.Status=candidate

▽

△

Class	**SwitchStreamIdentification**			
filterActionVlan Modification	PositiveInteger	0..1	attr	Defines the action to modify the VLAN-ID within a VLAN tag of an Ethernet frame. **Tags:** atp.Status=candidate
ingressPort	CouplingPort	*	ref	Reference to the CouplingPort to be taken into account as the ingress role for this SwitchStreamIdentification. **Tags:** atp.Splitkey=ingressPort atp.Status=candidate
streamFilter Rule	SwitchStreamFilterRule	0..1	aggr	Definition of a stream filter rule for this SwitchStream Identification. **Tags:** atp.Splitkey=streamFilterRule, streamFilterRule.short Name atp.Status=candidate

Table 3.92: SwitchStreamIdentification

Another matching check is performed according to the definition of the `SwitchStreamFilterRule` defined at the `SwitchStreamIdentification.streamFilterRule`.

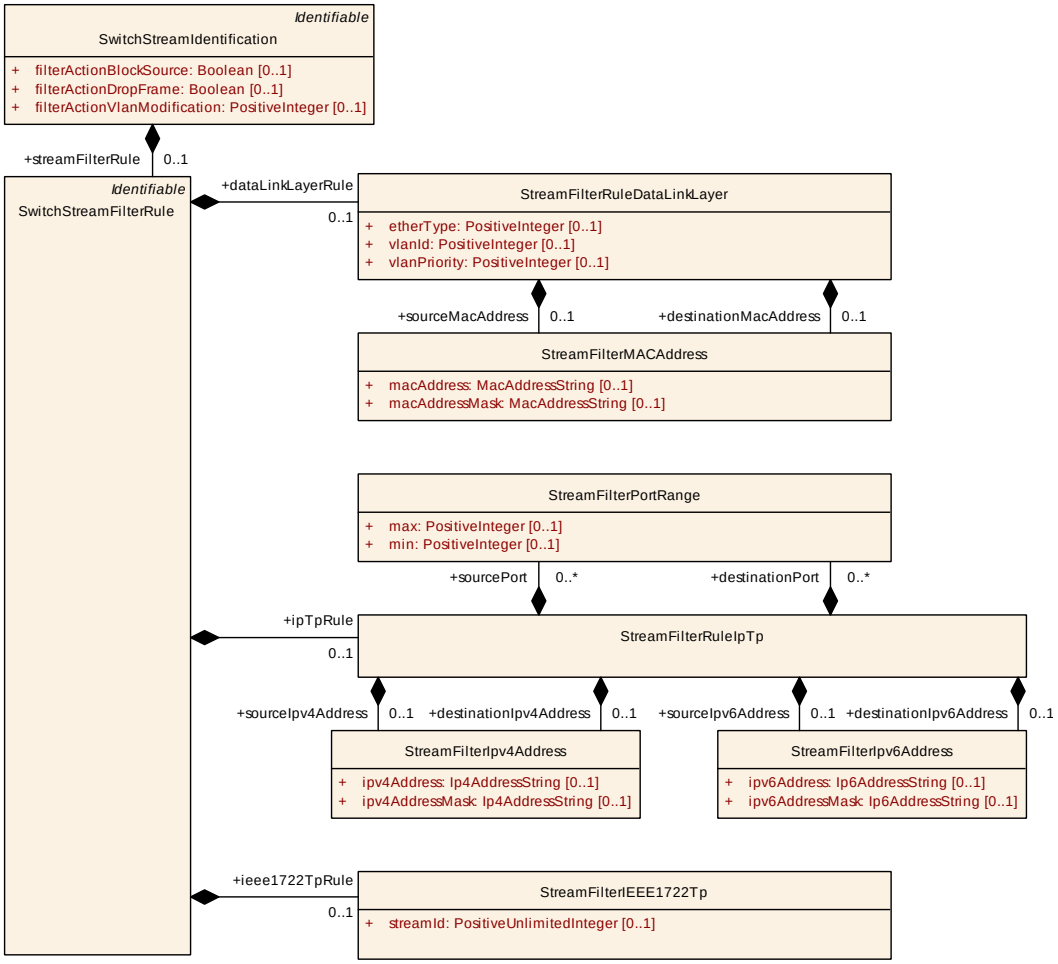

Figure 3.24: Ethernet `SwitchStreamFilterRule`

Class	**SwitchStreamFilterRule**			
Note	SwitchStreamIdentification **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SwitchStreamIdentification.streamFilterRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataLinkLayer Rule	StreamFilterRuleData LinkLayer	0..1	aggr	Definition of a filter rule on the data link layer. **Tags:** atp.Splitkey=dataLinkLayerRule atp.Status=candidate
ieee1722Tp Rule	StreamFilterIEEE1722 Tp	0..1	aggr	Definition of a filter rule for IEEE1722Tp. **Tags:** atp.Splitkey=ieee1722TpRule atp.Status=candidate
ipTpRule	StreamFilterRuleIpTp	0..1	aggr	Definition of a filter rule IP and TP. **Tags:** atp.Splitkey=ipTpRule atp.Status=candidate

Table 3.93: SwitchStreamFilterRule

Class	**StreamFilterRuleDataLinkLayer**			
Note	Configuration of filter rules on the DataLink layer **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	SwitchStreamFilterRule.dataLinkLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationMac Address	StreamFilter MACAddress	0..1	aggr	Filter to match packets with the destination MAC address/ mask. **Tags:** atp.Splitkey=destinationMacAddress atp.Status=candidate
etherType	PositiveInteger	0..1	attr	Filter to match packets based on the EtherType field in the Ethernet frame. **Tags:** atp.Status=candidate
sourceMac Address	StreamFilter MACAddress	0..1	aggr	Filter to match packets with the source MAC address/ mask. **Tags:** atp.Splitkey=sourceMacAddress atp.Status=candidate
vlanId	PositiveInteger	0..1	attr	Filter of packets with a VlanId. Note: for double tagged Ethernet frames the filter rule primitive will be applied to the C-Tag. **Tags:** atp.Status=candidate
vlanPriority	PositiveInteger	0..1	attr	Filter of packets with a Vlan priority. Note: for double tagged Ethernet frames the filter rule primitive will be applied to the C-Tag. **Tags:** atp.Status=candidate

Table 3.94: StreamFilterRuleDataLinkLayer

Class	**StreamFilterMACAddress**			
Note	Configuration of filter rules on the DataLink layer **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	StreamFilterRuleDataLinkLayer.destinationMacAddress, StreamFilterRuleDataLinkLayer.sourceMacAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
macAddress	MacAddressString	0..1	attr	Filter to match packets with the MAC address. **Tags:** atp.Status=candidate
macAddress Mask	MacAddressString	0..1	attr	Filter to match packets with the MAC address range. **Tags:** atp.Status=candidate

Table 3.95: StreamFilterMACAddress

Class	**StreamFilterRuleIpTp**			
Note	Configuration of filter rules for IP and TP. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	SwitchStreamFilterRule.ipTpRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationIpv4 Address	StreamFilterIpv4 Address	0..1	aggr	Filter to match packets with the destination IPv4 address range. **Tags:** atp.Splitkey=destinationIpv4Address atp.Status=candidate
destinationIpv6 Address	StreamFilterIpv6 Address	0..1	aggr	Filter to match packets with the destination IPv6 address range. **Tags:** atp.Splitkey=destinationIpv6Address atp.Status=candidate
destinationPort	StreamFilterPortRange	*	aggr	Filter to match packets with the set of destination UDP/TCP port ranges. **Tags:** atp.Splitkey=destinationPort atp.Status=candidate
sourceIpv4 Address	StreamFilterIpv4 Address	0..1	aggr	Filter to match packets with the source IPv4 address range. **Tags:** atp.Splitkey=sourceIpv4Address atp.Status=candidate
sourceIpv6 Address	StreamFilterIpv6 Address	0..1	aggr	Filter to match packets with the source IPv6 address range. **Tags:** atp.Splitkey=sourceIpv6Address atp.Status=candidate
sourcePort	StreamFilterPortRange	*	aggr	Filter to match packets with the set of source UDP/TCP port ranges. **Tags:** atp.Splitkey=sourcePort atp.Status=candidate

Table 3.96: StreamFilterRuleIpTp

Class	**StreamFilterIpv4Address**			
Note	IPv4 address range definition. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	StreamFilterRuleIpTp.destinationIpv4Address, StreamFilterRuleIpTp.sourceIpv4Address			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ipv4Address	Ip4AddressString	0..1	attr	Filter to match packets with the IPv4 address. **Tags:** atp.Status=candidate
ipv4Address Mask	Ip4AddressString	0..1	attr	Filter to match packets with the IPv4 address range. **Tags:** atp.Status=candidate

Table 3.97: StreamFilterIpv4Address

Class	**StreamFilterIpv6Address**			
Note	IPv6 address range definition. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	StreamFilterRuleIpTp.destinationIpv6Address, StreamFilterRuleIpTp.sourceIpv6Address			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ipv6Address	Ip6AddressString	0..1	attr	Filter to match packets with the IPv6 address. **Tags:** atp.Status=candidate
ipv6Address Mask	Ip6AddressString	0..1	attr	Filter to match packets with the IPv6 address range. **Tags:** atp.Status=candidate

Table 3.98: StreamFilterIpv6Address

Class	**StreamFilterPortRange**			
Note	Configuration of filter rules for IP and TP. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	StreamFilterRuleIpTp.destinationPort, StreamFilterRuleIpTp.sourcePort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
max	PositiveInteger	0..1	attr	Filter to match packets with the maximum UDP/TCP port number. **Tags:** atp.Status=candidate
min	PositiveInteger	0..1	attr	Filter to match packets with the minimum UDP/TCP port number. **Tags:** atp.Status=candidate

Table 3.99: StreamFilterPortRange

Class	**StreamFilterIEEE1722Tp**			
Note	Configuration of filter rules for IP and TP. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	SwitchStreamFilterRule.ieee1722TpRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
streamId	PositiveUnlimitedInteger	0..1	attr	Filter to match IEEE1722Tp packets with the stream Id number. Defined as 64bit stream id. **Tags:** atp.Status=candidate

Table 3.100: StreamFilterIEEE1722Tp

The actions to be performed when a frame is matched by both, `ingressPort` / `egressPort` and `SwitchStreamFilterRule` are defined at the `SwitchStream-Identification`.

[TPS_SYST_03114] Actions for `SwitchStreamIdentification`

Status: DRAFT

⌈The configurable actions to be performed when a frame matches a `SwitchStream-Identification` are:

- `filterActionDropFrame`
- `filterActionBlockSource`
- `filterActionVlanModification`
- `filterActionDestPortModification`

⌋

Class	**SwitchStreamFilterActionDestPortModification**			
Note	Defines the action to modify the destination port(s) determined by the frame forwarding process for an particular Ethernet frame. Either the egress destination of an Ethernet frame is extended or overwritten. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SwitchStreamIdentification.filterActionDestPortModification			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
egressPort	CouplingPort	*	ref	Reference to the egress ports used as the target of the filter action to modify the egress port. **Tags:** atp.Splitkey=egressPort atp.Status=candidate
modification	SwitchStreamFilter ActionPortModification Enum	0..1	attr	Defines the method to modify the egress destination. Either overwrite or extend the egress destination. **Tags:** atp.Status=candidate

Table 3.101: SwitchStreamFilterActionDestPortModification

Enumeration	**SwitchStreamFilterActionPortModificationEnum**
Note	Definition how the SwitchStreamFilterActionPortModification is applied. **Tags:** atp.Status=candidate This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	SwitchStreamFilterActionDestPortModification.modification
Literal	***Description***
extend	Extend the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This corresponds to the union of the existing egress destination port vector and the configured egress destination port(s). **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate
limit	Limit the egress destination port vector of an Ethernet frame to the configured egress destination port(s). This corresponds to the intersection of the existing egress destination port vector and the configured egress destination port(s). **Tags:** atp.EnumerationLiteralIndex=2 atp.Status=candidate
overwrite	Overwrite the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This results in the existing egress destination port vector being replaced by the configured egress destination port(s). **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate

Table 3.102: SwitchStreamFilterActionPortModificationEnum

The optional definition of a SwitchStreamFilterEntry becomes active when a frame matches the first entry of the ordered referenced SwitchStreamIdentifications in the role SwitchStreamFilterEntry.streamIdentificationHandle.

The list of CouplingElementSwitchDetails.streamFilter elements is ordered, thus it is essential in which order the SwitchStreamFilterEntry entries are defined. The earlier entries in the list are matched before later entries in the list. Thus, if a SwitchStreamIdentification is referenced by several SwitchStreamFilterEntry elements in the role streamIdentificationHandle, then only the first match from the list is getting effective. Later matches are not considered.

[TPS_SYST_03115] SwitchStreamFilterEntry for SwitchStreamIdentification

Status: DRAFT

⌈If a SwitchStreamFilterEntry refers to a SwitchStreamIdentification in the role streamIdentificationHandle, then this SwitchStreamFilterEntry is executed when a frame matches the SwitchStreamIdentification.

If a SwitchStreamFilterEntry has the attribute streamIdentificationWildcard set to *true*, then any SwitchStreamIdentification will be matching this SwitchStreamFilterEntry.⌋

Further actions defined in the context of the SwitchStreamFilterEntry are:

- assigning an atsInstanceEntry

- assigning a streamGate
- assigning a flowMetering

[TPS_SYST_03111] Ethernet switch SwitchAtsInstanceEntry

Status: DRAFT

⌈The attributes committedBurstSize and committedInformationRate define the bucket parameters for the ATS.

The reference trafficShaperGroup assigns each SwitchAtsInstanceEntry to a SwitchAsynchronousTrafficShaperGroupEntry.⌋

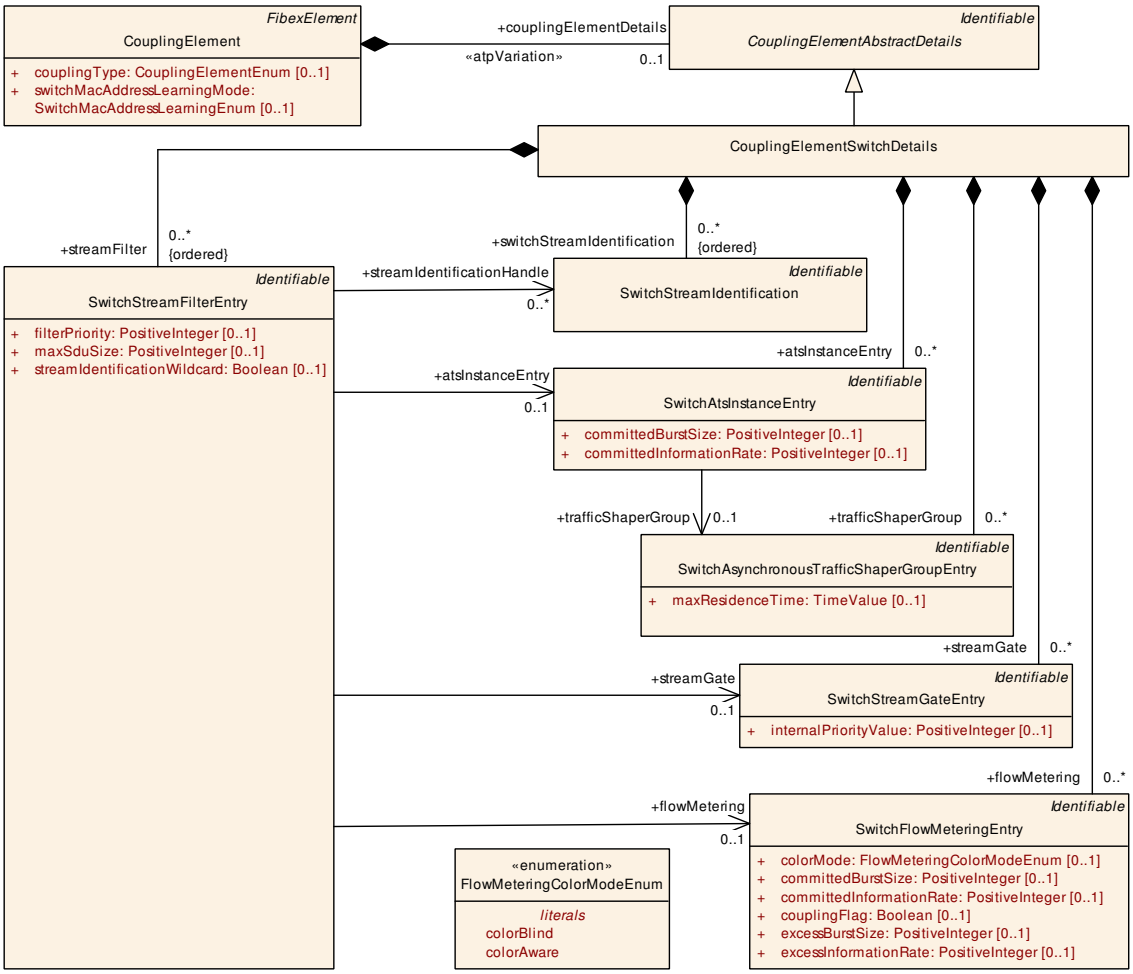

Figure 3.25: Ethernet SwitchStreamFilterEntry

Class	**SwitchStreamFilterEntry**
Note	Defines a Stream Filter Entry. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	**SwitchStreamFilterEntry**			
Aggregated by	CouplingElementSwitchDetails.streamFilter			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
atsInstance Entry	SwitchAtsInstanceEntry	0..1	ref	Reference to the ATS Instance Entry. **Tags:** atp.Splitkey=atsInstanceEntry atp.Status=candidate
filterPriority	PositiveInteger	0..1	attr	Defines the Priority of this Stream Filter Entry. **Tags:** atp.Status=candidate
flowMetering	SwitchFlowMetering Entry	0..1	ref	Reference to a Flow Metering Entry. **Tags:** atp.Splitkey=flowMetering atp.Status=candidate
maxSduSize	PositiveInteger	0..1	attr	Defines the maximum SDU size (size of an Ethernet package) which is acceptable to be processed by the Ethernet switch. **Tags:** atp.Status=candidate
streamGate	SwitchStreamGateEntry	0..1	ref	Reference to a Stream Gate Entry. **Tags:** atp.Splitkey=streamGate atp.Status=candidate
stream Identification Handle	SwitchStream Identification	*	ref	Reference to the SwitchStreamIdentifications this Stream FilterEntry applies to. **Tags:** atp.Splitkey=streamIdentificationHandle atp.Status=candidate
stream Identification Wildcard	Boolean	0..1	attr	Defines whether this Stream Filter Entry includes the wildcard for SwitchStreamIdentification. **Tags:** atp.Status=candidate

Table 3.103: SwitchStreamFilterEntry

Class	**SwitchAtsInstanceEntry**			
Note	Defines an Asynchronous Traffic Shaper (ATS) for the stream filter. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElementSwitchDetails.atsInstanceEntry			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
committedBurst Size	PositiveInteger	0..1	attr	Maximum token capacity of the token bucket in bit. **Tags:** atp.Status=candidate
committed InformationRate	PositiveInteger	0..1	attr	Defines the rate at which the token bucket is refilled with tokens in bit per second. **Tags:** atp.Status=candidate
trafficShaper Group	SwitchAsynchronous TrafficShaperGroup Entry	0..1	ref	Reference to the Traffic Shaper Group this Asynchronous Traffic Shaper is part of. **Tags:** atp.Splitkey=trafficShaperGroup atp.Status=candidate

Table 3.104: SwitchAtsInstanceEntry

Class	SwitchAsynchronousTrafficShaperGroupEntry			
Note	Defines an Asynchronous Traffic Shapter (ATS) Group for a switch. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElementSwitchDetails.trafficShaperGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxResidence Time	TimeValue	0..1	attr	Defines the maximum duration limit for which frames can reside in a switch (in seconds). **Tags:** atp.Status=candidate

Table 3.105: SwitchAsynchronousTrafficShaperGroupEntry

Class	SwitchStreamGateEntry			
Note	Defines a Asynchronous Traffic Shapter (ATS) Group for a switch. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElementSwitchDetails.streamGate			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
internalPriority Value	PositiveInteger	0..1	attr	Internal Priority Value (IPV), a priority value that determines the assigned traffic class. **Tags:** atp.Status=candidate

Table 3.106: SwitchStreamGateEntry

Class	SwitchFlowMeteringEntry			
Note	Defines a Flow Metering Entry for a switch. **Tags:** atp.Status=candidate This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CouplingElementSwitchDetails.flowMetering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
colorMode	FlowMeteringColor ModeEnum	0..1	attr	Defines whether color-aware or color-blind mode shall be used. **Tags:** atp.Status=candidate
committedBurst Size	PositiveInteger	0..1	attr	Committed Burst Size (CBS) (accepted burst size in green token bucket). **Tags:** atp.Status=candidate
committed InformationRate	PositiveInteger	0..1	attr	Committed Information Rate (CIR) (accepted rate in green token bucket) in bits per second. **Tags:** atp.Status=candidate
couplingFlag	Boolean	0..1	attr	Coupling Flag that defines if unused "green" tokens in the first bucket are transferred to the second bucket as "yellow" tokens. **Tags:** atp.Status=candidate

△

Class	**SwitchFlowMeteringEntry**			
excessBurst Size	PositiveInteger	0..1	attr	Excess burst size (EBS) (accepted burst size in yellow token bucket). **Tags:** atp.Status=candidate
excess InformationRate	PositiveInteger	0..1	attr	Excess Information Rate (EIR) (accepted rate in yellow token bucket) in bits per second. **Tags:** atp.Status=candidate

Table 3.107: SwitchFlowMeteringEntry

Enumeration	**FlowMeteringColorModeEnum**
Note	Defines whether Flow Metering color-aware or color-blind mode is used. **Tags:** atp.Status=candidate This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	SwitchFlowMeteringEntry.colorMode
Literal	***Description***
colorAware	Flow Metering color aware mode. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate
colorBlind	Flow Metering color blind mode. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 3.108: FlowMeteringColorModeEnum

The applicability of attributes of `CouplingPort` and related classes on the usage in an Ethernet endpoint or Ethernet switch is summarized in [TPS_SYST_02422].

[constr_9368] Applicable attributes for an Ethernet switch

Imposition time: CP: `IT_SysDesc`

⌈If a `CouplingPort` is aggregated by a `CouplingElement` and that `CouplingElement` has the attribute `couplingType` set to the value `CouplingElementEnum.switch`, then the attributes defined in the column "Switch Port" of [TPS_SYST_02422] are applicable.⌋

[constr_9369] Applicable attributes for an Ethernet node on the classic platform

Imposition time: CP: `IT_SysDesc`

⌈If a `CouplingPort` is aggregated by an `EthernetCommunicationController` and that `EthernetCommunicationController` is in turn aggregated by an `EcuInstance`, then the attributes defined in the column "Endpoint" of [TPS_SYST_02422] are applicable.⌋

[TPS_SYST_02422] Applicability of Ethernet model elements ⌈

Model Element	Switch Port	Endpoint
CouplingPort		
CouplingPort.connectionNegotiationBehavior	1	1
CouplingPort.couplingPortDetails	0..1	0..1
CouplingPort.couplingPortRole	0..1	0..1
CouplingPort.defaultVlan	0..1	0
CouplingPort.macAddressVlanAssignment	0..*	0
CouplingPort.macLayerType	0..1	0..1
CouplingPort.macSecProps	0..*	0..*
CouplingPort.physicalLayerType	1	1
CouplingPort.plcaProps	0	0..1
CouplingPort.pncMapping	0..*	0..*
CouplingPort.receiveActivity	0..1	0
CouplingPort.vlanMembership	0..*	0
CouplingPort.wakeupSleepOnDatalineConfig	0..1	0..1
CouplingPortDetails		
CouplingPortDetails.couplingPortStructuralElement	0..*	0..*
CouplingPortDetails.defaultTrafficClass	0..1	0
CouplingPortDetails.ethernetPriorityRegeneration	0..*	0
CouplingPortDetails.ethernetTrafficClassAssignment	0..*	0
CouplingPortDetails.framePreemptionSupport	0..1	0..1
CouplingPortDetails.globalTimeProps	0..1	0
CouplingPortDetails.lastEgressScheduler	0..1	0..1
CouplingPortDetails.vlanTranslationTable	0..*	0
CouplingPortScheduler		
CouplingPortScheduler.portScheduler	1	1
CouplingPortScheduler.predecessor	1..*	1..*
CouplingPortFifo		
CouplingPortFifo.assignedTrafficClass	0..*	0
CouplingPortFifo.minimumFifoLength	1	0
CouplingPortFifo.shaper	0..1	0..1
CouplingPortFifo.trafficClassPreemptionSupport	0..1	0..1
CouplingPortAsynchronousTrafficShaper	0..1	0..1
CouplingPortCreditBasedShaper	0..1	0..1
CouplingPortCreditBasedShaper.idleSlope	0..1	0..1
CouplingPortEnhancedTrafficShaper	0..1	0..1
CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent	0..1	0..1
CouplingPortEnhancedTrafficShaper. etsAvailableBandwidthInWeightValue	0..1	0..1

⌋

3.3.5.7 TcpIp stack configuration properties

The EcuInstance references the following elements and allows to set Ecu specific TcpIp stack configuration options in the System Description:

- EthIpProps - used to configure IPv4 and IPv6
- EthTcpIpProps - used to configure TCP and UDP
- EthTcpIpIcmpProps - used to configure ICMP

3.3.5.7.1 IP configuration properties

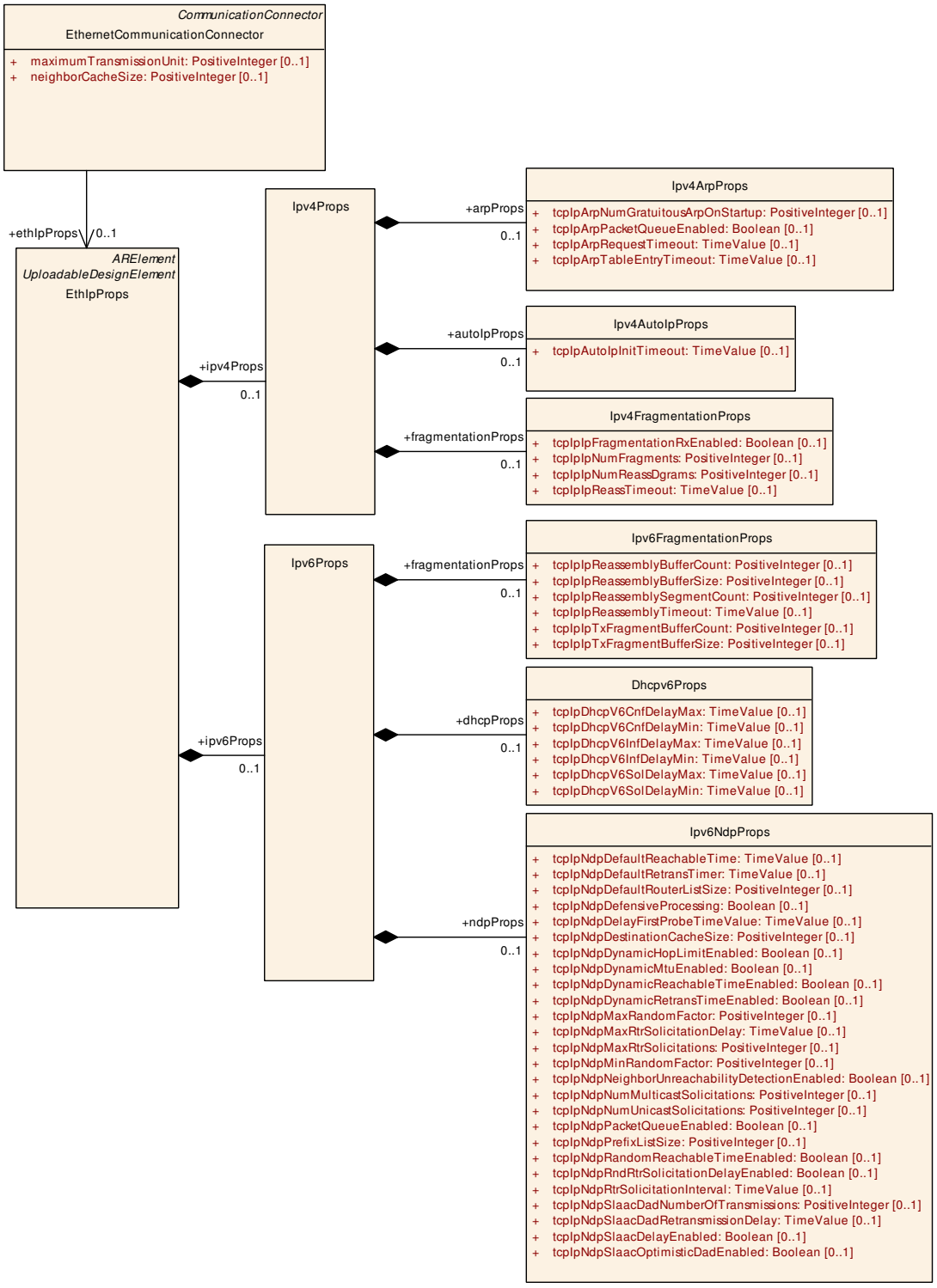

Figure 3.26: Ecu specific IP configuration options

Class	EthIpProps			
Note	This meta-class is used to configure the EcuInstance specific IP attributes. **Tags:** atp.recommendedPackage=EthIpProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ipv4Props	Ipv4Props	0..1	aggr	Configuration options for IPv4. **Tags:** atp.Splitkey=ipv4Props
ipv6Props	Ipv6Props	0..1	aggr	Configuration options for IPv6. **Tags:** atp.Splitkey=ipv6Props

Table 3.109: EthIpProps

Class	Ipv4Props			
Note	This meta-class specifies the configuration options for IPv4.			
Base	*ARObject*			
Aggregated by	EthIpProps.ipv4Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arpProps	Ipv4ArpProps	0..1	aggr	Configuration properties for the ARP (Address Resolution Protocol). **Tags:** atp.Splitkey=arpProps
autoIpProps	Ipv4AutoIpProps	0..1	aggr	Configuration options for Auto-IP (automatic private IP addressing). **Tags:** atp.Splitkey=autoIpProps
fragmentation Props	Ipv4Fragmentation Props	0..1	aggr	Configuration options for IPv4 packet fragmentation/ reassembly. **Tags:** atp.Splitkey=fragmentationProps

Table 3.110: Ipv4Props

Class	Ipv4ArpProps			
Note	Specifies the configuration options for the ARP (Address Resolution Protocol).			
Base	*ARObject*			
Aggregated by	Ipv4Props.arpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpArpNum GratuitousArp OnStartup	PositiveInteger	0..1	attr	This attribute specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.
tcpIpArpPacket QueueEnabled	Boolean	0..1	attr	This attribute enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.
tcpIpArp Request Timeout	TimeValue	0..1	attr	This attribute specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the TcpIp shall skip the transmission of any further ARP requests to the same destination within a duration of tcpIpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1).
tcpIpArpTable EntryTimeout	TimeValue	0..1	attr	This attribute specifies the timeout in seconds after which an unused ARP entry is removed.

Table 3.111: Ipv4ArpProps

[constr_5126] Value range of Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup shall be in the range of 0..255.⌋

Class	**Ipv4AutoIpProps**			
Note	Specifies the configuration options for Auto-IP (automatic private IP addressing).			
Base	*ARObject*			
Aggregated by	Ipv4Props.autoIpProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tcpIpAutoIpInit Timeout	TimeValue	0..1	attr	This attribute specifies the time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.

Table 3.112: Ipv4AutoIpProps

Class	**Ipv4FragmentationProps**			
Note	Specifies the configuration options for IPv4 packet fragmentation/reassembly.			
Base	*ARObject*			
Aggregated by	Ipv4Props.fragmentationProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tcpIpIp Fragmentation RxEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).
tcpIpIpNum Fragments	PositiveInteger	0..1	attr	Specifies the maximum number of IP fragments per datagram.
tcpIpIpNum ReassDgrams	PositiveInteger	0..1	attr	Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel.
tcpIpIpReass Timeout	TimeValue	0..1	attr	Specifies the timeout in [s] after which an incomplete datagram gets discarded.

Table 3.113: Ipv4FragmentationProps

[constr_5127] Value range of Ipv4FragmentationProps.tcpIpIpNumFragments

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4FragmentationProps.tcpIpIpNumFragments shall be in the range of 0..255.⌋

[constr_5128] Value range of Ipv4FragmentationProps.tcpIpIpNumReassDgrams

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv4FragmentationProps.tcpIpIpNumReassDgrams shall be in the range of 0..65535.⌋

Class	Ipv6Props			
Note	This meta-class specifies the configuration options for IPv6.			
Base	*ARObject*			
Aggregated by	EthIpProps.ipv6Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dhcpProps	Dhcpv6Props	0..1	aggr	Configuration properties for DHCPv6. **Tags:** atp.Splitkey=dhcpProps
fragmentation Props	Ipv6Fragmentation Props	0..1	aggr	Configuration properties for IPv6 packet fragmentation/ reassembly. **Tags:** atp.Splitkey=fragmentationProps
ndpProps	Ipv6NdpProps	0..1	aggr	Configuration properties for the Neighbor Discovery Protocol for IPv6. **Tags:** atp.Splitkey=ndpProps

Table 3.114: Ipv6Props

Class	Ipv6FragmentationProps			
Note	This meta-class specifies the configuration options for IPv6 packet fragmentation/reassembly.			
Base	*ARObject*			
Aggregated by	Ipv6Props.fragmentationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpIp Reassembly BufferCount	PositiveInteger	0..1	attr	Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded. A value of 0 disables fragment reassembly.
tcpIpIp Reassembly BufferSize	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.
tcpIpIp Reassembly SegmentCount	PositiveInteger	0..1	attr	Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed. To deal with fragments received out of order this value should be configured bigger than 1.
tcpIpIp Reassembly Timeout	TimeValue	0..1	attr	Specifies the timeout in seconds after which an incomplete datagram gets discarded.
tcpIpIpTx FragmentBuffer Count	PositiveInteger	0..1	attr	These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented. A value of 0 disables tx fragmentation.
tcpIpIpTx FragmentBuffer Size	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.

Table 3.115: Ipv6FragmentationProps

[constr_5129] Value range of `Ipv6FragmentationProps.tcpIpIpReassembly-BufferCount`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpReassembly-BufferCount` shall be in the range of 0..255.⌋

[constr_5130] Value range of Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize shall be in the range of 1500..65535.⌋

[constr_5131] Value range of Ipv6FragmentationProps.tcpIpIpReassemblyTimeout

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassemblyTimeout shall be in the range of 0.001..100.⌋

[constr_5132] Value range of Ipv6FragmentationProps.tcpIpIpReassemblySegmentCount

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpReassemblySegmentCount shall be in the range of 1..255.⌋

[constr_5133] Value range of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount shall be in the range of 1..1000.⌋

[constr_5134] Value range of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize shall be in the range of 1500..65535.⌋

Class	**Dhcpv6Props**			
Note	This meta-class specifies the configuration options for DHCPv6.			
Base	*ARObject*			
Aggregated by	Ipv6Props.dhcpProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpDhcpV6 CnfDelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcpV6 CnfDelayMin	TimeValue	0..1	attr	Minimum delay in seconds before the first Confirm message will be sent.

▽

△

Class	Dhcpv6Props			
tcpIpDhcpV6InfDelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcpV6InfDelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Information Request message will be sent.
tcpIpDhcpV6SolDelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcpV6SolDelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Solicit message will be sent.

Table 3.116: Dhcpv6Props

[constr_5135] Value range of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin and Dhcpv6Props.tcpIpDhcpV6CnfDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin.⌋

[constr_5136] Value range of Dhcpv6Props.tcpIpDhcpV6InfDelayMin and Dhcpv6Props.tcpIpDhcpV6InfDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6InfDelayMin.⌋

[constr_5137] Value range of Dhcpv6Props.tcpIpDhcpV6SolDelayMin and Dhcpv6Props.tcpIpDhcpV6SolDelayMax

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMin and the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMax shall be in the range of 0..100 and the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMax shall be greater than the value of Dhcpv6Props.tcpIpDhcpV6SolDelayMin.⌋

Class	Ipv6NdpProps			
Note	This meta-class specifies the configuration options for the Neighbor Discovery Protocol for IPv6.			
Base	ARObject			
Aggregated by	Ipv6Props.ndpProps			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**Ipv6NdpProps**			
tcpIpNdpDefault ReachableTime	TimeValue	0..1	attr	Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RetransTimer	TimeValue	0..1	attr	Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RouterListSize	PositiveInteger	0..1	attr	Maximum number of default router entries.
tcpIpNdp Defensive Processing	Boolean	0..1	attr	If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861.
tcpIpNdpDelay FirstProbeTime Value	TimeValue	0..1	attr	Delay before sending the first NUD probe in (s).
tcpIpNdp Destination CacheSize	PositiveInteger	0..1	attr	Maximum number of entries in the destination cache.
tcpIpNdp DynamicHop LimitEnabled	Boolean	0..1	attr	If enabled the default hop limit may be reconfigured based on received Router Advertisements.
tcpIpNdp DynamicMtu Enabled	Boolean	0..1	attr	Allow dynamic reconfiguration of link MTU via Router Advertisements.
tcpIpNdp Dynamic ReachableTime Enabled	Boolean	0..1	attr	If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.
tcpIpNdp Dynamic RetransTime Enabled	Boolean	0..1	attr	If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.
tcpIpNdpMax RandomFactor	PositiveInteger	0..1	attr	Maximum random factor used for randomization
tcpIpNdpMaxRtr Solicitation Delay	TimeValue	0..1	attr	Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).
tcpIpNdpMaxRtr Solicitations	PositiveInteger	0..1	attr	Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.
tcpIpNdpMin RandomFactor	PositiveInteger	0..1	attr	Minimum random factor used for randomization
tcpIpNdp Neighbor Unreachability Detection Enabled	Boolean	0..1	attr	Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.
tcpIpNdpNum Multicast Solicitations	PositiveInteger	0..1	attr	Maximum number of multicast solicitations that will be sent when performing address resolution.
tcpIpNdpNum Unicast Solicitations	PositiveInteger	0..1	attr	Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection.
tcpIpNdpPacket QueueEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.
tcpIpNdpPrefix ListSize	PositiveInteger	0..1	attr	Maximum number of entries in the on-link prefix list.

▽

△

Class	Ipv6NdpProps			
tcpIpNdp Random ReachableTime Enabled	Boolean	0..1	attr	If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time.
tcpIpNdpRndRtr Solicitation DelayEnabled	Boolean	0..1	attr	If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds.
tcpIpNdpRtr Solicitation Interval	TimeValue	0..1	attr	Interval between consecutive Router Solicitations in (s).
tcpIpNdpSlaac DadNumberOf Transmissions	PositiveInteger	0..1	attr	Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.
tcpIpNdpSlaac Dad Retransmission Delay	TimeValue	0..1	attr	Sets the maximum value for the address configuration delay (s).
tcpIpNdpSlaac DelayEnabled	Boolean	0..1	attr	If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].
tcpIpNdpSlaac OptimisticDad Enabled	Boolean	0..1	attr	Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.

Table 3.117: Ipv6NdpProps

[constr_5138] Value range of `Ipv6NdpProps.tcpIpNdpSlaacDadNumberOf-Transmissions`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpSlaacDadNumberOf-Transmissions` shall be in the range of 0..254.⌋

[constr_5139] Value range of `Ipv6NdpProps.tcpIpNdpSlaacDad-RetransmissionDelay`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpSlaacDadRetransmission-Delay` shall be in the range of 0..10.⌋

[constr_5140] Value range of `Ipv6NdpProps.tcpIpNdpDefaultReachableTime`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDefaultReachableTime` shall be in the range of 0..120.⌋

[constr_5141] Value range of `Ipv6NdpProps.tcpIpNdpDefaultRetransTimer`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDefaultRetransTimer` shall be in the range of 0..60.⌋

[constr_5142] Value range of Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations shall be in the range of 0..255.⌋

[constr_5143] Value range of Ipv6NdpProps.tcpIpNdpNumMulticastSolicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpNumMulticastSolicitations shall be in the range of 0..255.⌋

[constr_5144] Value range of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue shall be in the range of 0..60.⌋

[constr_5145] Value range of Ipv6NdpProps.tcpIpNdpMinRandomFactor

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMinRandomFactor shall be in the range of 0..100.⌋

[constr_5146] Value range of Ipv6NdpProps.tcpIpNdpMaxRandomFactor

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRandomFactor shall be in the range of 0..100.⌋

[constr_5147] Value range of Ipv6NdpProps.tcpIpNdpDestinationCacheSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDestinationCacheSize shall be in the range of 1..254.⌋

[constr_5148] Value range of Ipv6NdpProps.tcpIpNdpPrefixListSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpPrefixListSize shall be in the range of 1..254.⌋

[constr_5149] Value range of Ipv6NdpProps.tcpIpNdpDefaultRouterListSize

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpDefaultRouterListSize shall be in the range of 2..254.⌋

[constr_5151] Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations shall be in the range of 0..255.⌋

[constr_5152] Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitation-Delay

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitationDelay shall be in the range of 0.001..60.⌋

[constr_5153] Value range of Ipv6NdpProps.tcpIpNdpRtrSolicitation-Interval

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval shall be in the range of 0.001..60.⌋

3.3.5.7.2 TCP and UDP configuration properties

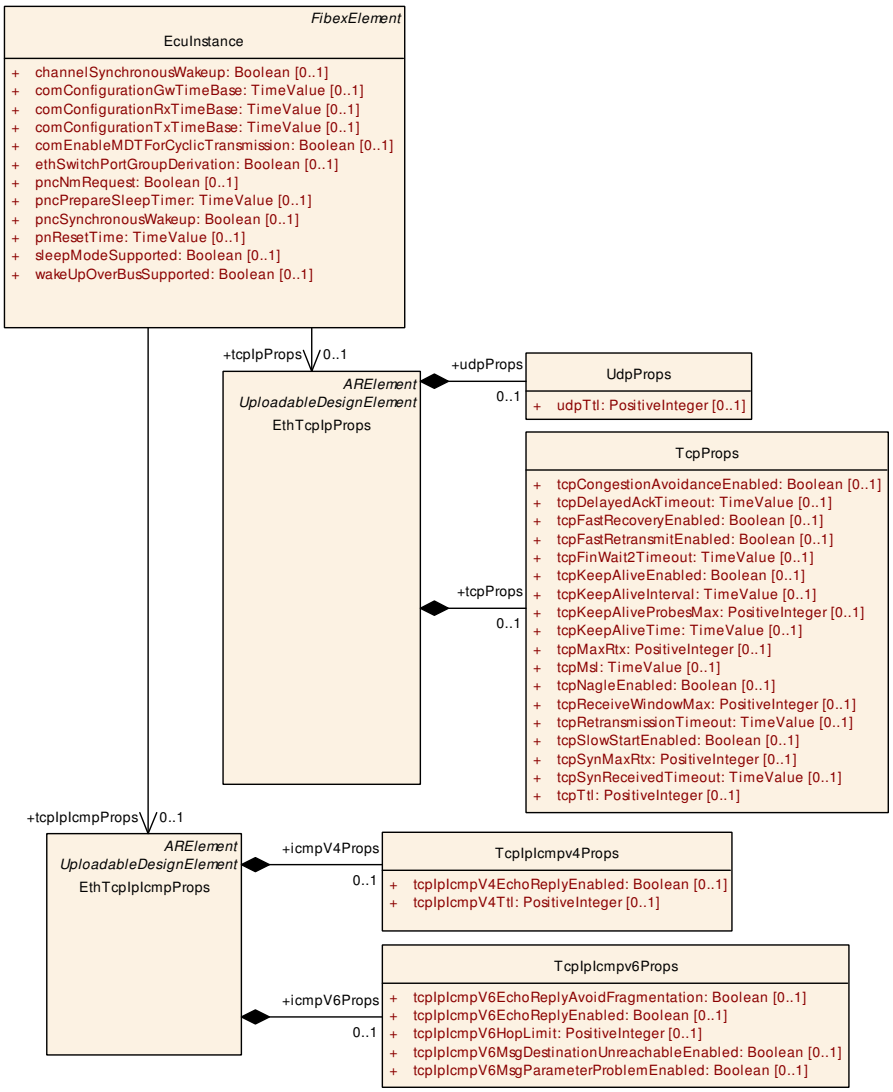

Figure 3.27: Ecu specific TCP/UDP and ICMP configuration options

Class	**EthTcpIpProps**			
Note	This meta-class is used to configure the EcuInstance specific TcpIp Stack attributes. **Tags:** atp.recommendedPackage=EthTcpIpProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpProps	TcpProps	0..1	aggr	TCP configuration properties **Tags:** atp.Splitkey=tcpProps
udpProps	UdpProps	0..1	aggr	UDP configuration properties **Tags:** atp.Splitkey=udpProps

Table 3.118: EthTcpIpProps

Class	UdpProps			
Note	This meta-class specifies the configuration options for UDP (User Datagram Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpProps.udpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
udpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing UDP packets.

Table 3.119: UdpProps

[constr_5118] Value range of UdpProps.udpTtl

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, the value of UdpProps.udpTtl shall be in the range of 1..255.⌋

Class	TcpProps			
Note	This meta-class specifies the configuration options for TCP (Transmission Control Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpProps.tcpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpCongestion Avoidance Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.
tcpDelayedAck Timeout	TimeValue	0..1	attr	The maximal time an acknowledgement is delayed for transmission in seconds.
tcpFast Recovery Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.
tcpFast Retransmit Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.
tcpFinWait2 Timeout	TimeValue	0..1	attr	Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.
tcpKeepAlive Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6.
tcpKeepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
tcpKeepAlive ProbesMax	PositiveInteger	0..1	attr	Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.
tcpKeepAlive Time	TimeValue	0..1	attr	Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe.
tcpMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if tcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.
tcpMsl	TimeValue	0..1	attr	Maximum segment lifetime in [s].

▽

△

Class	TcpProps			
tcpNagle Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated per Socket (with the attribute TcpTp.nagle Algorithm).
tcpReceive WindowMax	PositiveInteger	0..1	attr	Default value of maximum receive window in bytes.
tcp Retransmission Timeout	TimeValue	0..1	attr	Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled, no TCP segments shall be retransmitted.
tcpSlowStart Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681.
tcpSynMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP SYN is retransmitted.
tcpSynReceived Timeout	TimeValue	0..1	attr	Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgement after having both received and sent a connection request.
tcpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing TCP packets.

Table 3.120: TcpProps

[constr_5119] Value range of `TcpProps.tcpTtl`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpTtl` shall be in the range of 1..255.⌋

[constr_5120] Value range of `TcpProps.tcpDelayedAckTimeout`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpDelayedAckTimeout` shall be in the range of 0..0.5.⌋

[constr_5121] Value range of `TcpProps.tcpSynMaxRtx`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpSynMaxRtx` shall be in the range of 0..255.⌋

[constr_5122] Value range of `TcpProps.tcpMaxRtx`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpMaxRtx` shall be in the range of 0..255.⌋

[constr_5123] Value range of `TcpProps.tcpKeepAliveProbesMax`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpKeepAliveProbesMax` shall be in the range of 0..65535.⌋

[constr_5124] Value range of `TcpProps.tcpReceiveWindowMax`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpProps.tcpReceiveWindowMax` shall be in the range of 0..65535.⌋

3.3.5.7.3 ICMP configuration properties

Class	**EthTcpIpIcmpProps**			
Note	This meta-class is used to configure the EcuInstance specific ICMP (Internet Control Message Protocol) attributes **Tags:** atp.recommendedPackage=EthTcpIcmpProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
icmpV4Props	TcpIpIcmpv4Props	0..1	aggr	ICMPv4 configuration properties **Tags:** atp.Splitkey=icmpV4Props
icmpV6Props	TcpIpIcmpv6Props	0..1	aggr	ICMPv6 configuration properties **Tags:** atp.Splitkey=icmpV6Props

Table 3.121: EthTcpIpIcmpProps

Class	**TcpIpIcmpv4Props**			
Note	This meta-class specifies the configuration options for ICMPv4 (Internet Control Message Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpIcmpProps.icmpV4Props			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpIcmpV4 EchoReply Enabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcpIpIcmpV4Ttl	PositiveInteger	0..1	attr	This attribute is only relevant in case that ICMP (Internet Control Message Protocol) is used. It specifies the default Time-to-live value of outgoing ICMP packets.

Table 3.122: TcpIpIcmpv4Props

Class	**TcpIpIcmpv6Props**			
Note	This meta-class specifies the configuration options for ICMPv6 (Internet Control Message Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpIcmpProps.icmpV6Props			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tcpIpIcmpV6 EchoReply Avoid Fragmentation	Boolean	0..1	attr	This attribute defines whether the echo reply is only transmitted in case that the incoming ICMPv6 Echo Request (Pings) fits the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation.

▽

△

Class	TcpIpIcmpv6Props			
tcpIpIcmpV6 EchoReply Enabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcpIpIcmpV6 HopLimit	PositiveInteger	0..1	attr	Default Hop-Limit value of outgoing ICMPv6 packets.
tcpIpIcmpV6 MsgDestination Unreachable Enabled	Boolean	0..1	attr	This attribute Enables/Disables the transmission of Destination Unreachable Messages.
tcpIpIcmpV6 MsgParameter Problem Enabled	Boolean	0..1	attr	If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet.

Table 3.123: TcpIpIcmpv6Props

[constr_5125] Value range of `TcpIpIcmpv4Props.tcpIpIcmpV4Ttl`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpIpIcmpv4Props.tcpIpIcmpV4Ttl` shall be in the range of 1..255.⌋

[constr_5154] Value range of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit` shall be in the range of 1..255.⌋

3.3.5.8 Ethernet wake-up and sleep on dataline

AUTOSAR supports the wake-up and sleep mechanism that complies with the Open Alliance TC10 specification (OA TC10, see [14]).

[TPS_SYST_03052] Enabling of wake-up and sleep mechanism ⌈The wake-up and sleep mechanism that complies with the Open Alliance TC10 specification (OA TC10) is enabled by defining the reference from `CouplingPort` to `EthernetWakeupSleepOnDatalineConfig` in the role `wakeupSleepOnDatalineConfig`.⌋

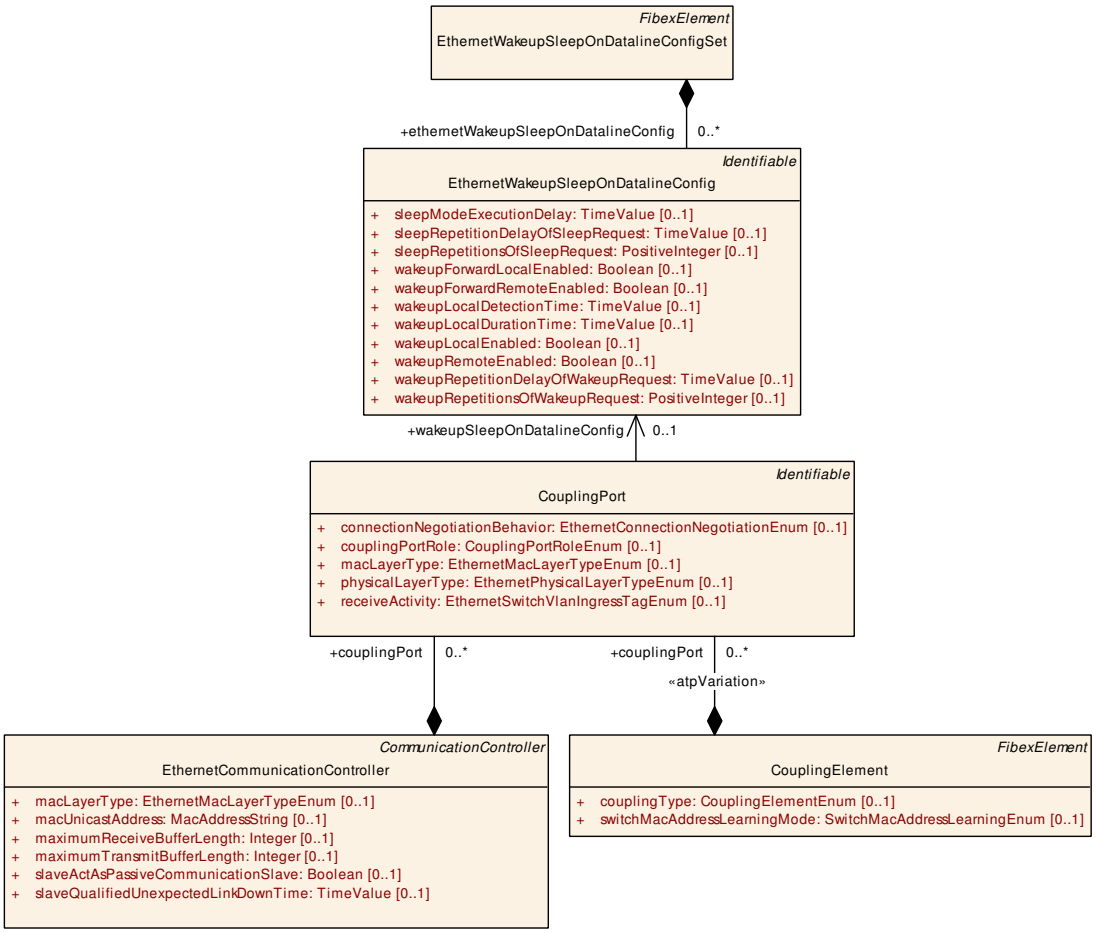

Figure 3.28: Wake on dataline model elements

The OA TC10 specifies service primitives to abstract the Ethernet hardware for ECUs connected via Automotive Ethernet (<bandwith>Base-T1). Drivers use the service primitives to trigger a wake-up and sleep on dataline and react on appropriate indications: *Sleep.request*, *Sleep.indication*, *Wakeup.indication*, *Wakeup.request*, *SleepFail.indication*, and *SleepAbort.request*.

Note:

- *SleepAbort.request* is not considered by AUTOSAR, as the AUTOSAR Network Management ensures a synchronized shutdown on the network. Thus, there is no need for an ECU to reject a *Sleep.request* upon the Network Management.
- *Inhibit.Indication* is not a service primitive, but an optional interface. This optional interface is not considered in AUTOSAR.

Class	**EthernetWakeupSleepOnDatalineConfig**
Note	EthernetWakeupSleepOnDatalineConfigSet is the main element that aggregates different config set regarding the wakeup and sleep on data line. An EthernetWakeupSleepOnDatalineConfigSet could aggregate multiple different configurations regarding the wakeup and sleep on dataline (EthernetWakeupSleepOnDatalineConfig).

△

Class	EthernetWakeupSleepOnDatalineConfig			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetWakeupSleepOnDatalineConfigSet.ethernetWakeupSleepOnDatalineConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
sleepMode ExecutionDelay	TimeValue	0..1	attr	Delay in seconds to perform a sleep request if the Ethernet hardware (PHY) detect a pending wake-up. This is used to avoid the race condition, if a sleep was requested while a wake-up of a neighboring PHY was received via a local wake-up connection (e.g. I/O pin).
sleepRepetition DelayOfSleep Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a sleep request. This is used to retry a synchronized shutdown of the connected Ethernet hardware (PHY) of the link partner.
sleep RepetitionsOf SleepRequest	PositiveInteger	0..1	attr	Count of repetitions for a sleep on dataline. If a sleep is rejected by the linked communication partner, the sleep is repeated until the count of repetitions exceed. If count of repetitions exceed, the Ethernet hardware (PHY) transit to sleep without acknowledgement of the connected link partner.
wakeupForward LocalEnabled	Boolean	0..1	attr	If enabled, then a remote wake-up received on the physical dataline (e.g. 100BASE-T1) is forwarded as local wake-up (e.g. via an I/O pin). If disabled, then a remote wake-up is not forwarded as local wake-up.
wakeupForward RemoteEnabled	Boolean	0..1	attr	If enabled, then a local wake-up is forwarded to the physical dataline (e.g. 100BASE-T1). If disabled, then a local wake-up is not forwarded to the physical dataline.
wakeupLocal DetectionTime	TimeValue	0..1	attr	Specify the detection time if a local wake-up in seconds is present on the local wake-up connection (e.g. I/O pin). A local wake-up has to be present at least for wakeupLocal DetectionTime to be detected a valid local wake-up.
wakeupLocal DurationTime	TimeValue	0..1	attr	Specify the duration of a local wake-up in seconds to be present on the local wake-up connection (e.g. I/O pin).
wakeupLocal Enabled	Boolean	0..1	attr	If enabled, then a local wake-up received via a local connection (e.g. I/O pin) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reacting on a local wake-up.
wakeupRemote Enabled	Boolean	0..1	attr	If enabled, then a remote wake-up received via the physical dataline (e.g. 100BASE-T1) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reaction on a remote wake-up.
wakeup RepetitionDelay OfWakeup Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.
wakeup RepetitionsOf Wakeup Request	PositiveInteger	0..1	attr	Count of repetitions for a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.

Table 3.124: EthernetWakeupSleepOnDatalineConfig

Class	EthernetWakeupSleepOnDatalineConfigSet
Note	This meta-class is the main element that aggregates different config set regarding the ethernet wakeup and sleep on data line. **Tags:** atp.recommendedPackage=EthernetWakeupSleepOnDatalineConfigSets

▽

△

Class	EthernetWakeupSleepOnDatalineConfigSet			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ethernet WakeupSleep OnDataline Config	EthernetWakeupSleep OnDatalineConfig	*	aggr	The relationship defines a collection of EthernetWakeup SleepOnDatalineConfig configurations which are available. **Tags:** atp.Splitkey=ethernetWakeupSleepOnDataline Config, ethernetWakeupSleepOnDatalineConfig.short Name

Table 3.125: EthernetWakeupSleepOnDatalineConfigSet

[constr_3601] Mandatory attributes of EthernetWakeupSleepOnDataline-Config

Imposition time: CP: IT_EcuExt, AP: IT_SysDes

⌈The following attributes of EthernetWakeupSleepOnDatalineConfig shall be defined:

- wakeupLocalEnabled
- wakeupRemoteEnabled

⌋

[constr_3602] Existence of wakeupForwardLocalEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupForwardLocalEnabled shall be defined if wakeupRemote-Enabled is set to TRUE.⌋

[constr_3603] Existence of wakeupLocalDurationTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupLocalDurationTime shall be defined if wakeupForward-LocalEnabled is set to TRUE.⌋

[constr_3604] Existence of wakeupForwardRemoteEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupForwardRemoteEnabled shall be defined if wakeupLocal-Enabled is set to TRUE.⌋

[constr_3605] Existence of wakeupLocalDetectionTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupLocalDetectionTime shall be defined if wakeupForward-RemoteEnabled is set to TRUE.⌋

[constr_3606] Values of wakeupLocalDurationTime and wakeupLocalDetectionTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If defined, then the value of wakeupLocalDurationTime shall be greater than the value of wakeupLocalDetectionTime.⌋

[constr_3609] Values of wakeupLocalDurationTime in the context of a CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈All CouplingPorts which have the reference wakeupSleepOnDatalineConfig defined and

- where the CouplingPorts are aggregated by the same CouplingElement and
- where the referenced EthernetWakeupSleepOnDatalineConfig has the attribute wakeupLocalDurationTime defined

shall refer to EthernetWakeupSleepOnDatalineConfigs where the value of wakeupLocalDurationTime is identical for all referencing CouplingPorts.⌋

[constr_3610] Values of wakeupLocalDetectionTime in the context of a CouplingElement

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈All CouplingPorts which have the reference wakeupSleepOnDatalineConfig defined and

- where the CouplingPorts are aggregated by the same CouplingElement and
- where the referenced EthernetWakeupSleepOnDatalineConfig has the attribute wakeupLocalDetectionTime defined

shall refer to EthernetWakeupSleepOnDatalineConfigs where the value of wakeupLocalDetectionTime is identical for all referencing CouplingPorts.⌋

Note: [constr_3609] and [constr_3610] ensure the same timing behavior within the used Ethernet hardware (e.g. Ethernet switch), if those CouplingPorts reference different EthernetWakeupSleepOnDatalineConfigs.

[constr_3607] Existence of sleepRepetitionDelayOfSleepRequest

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute sleepRepetitionDelayOfSleepRequest shall be defined if sleepRepetitionsOfSleepRequest is defined and has a value greater than 0.⌋

[constr_3608] Existence of wakeupRepetitionDelayOfWakeupRequest

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute wakeupRepetitionDelayOfWakeupRequest shall only be defined if wakeupRepetitionsOfWakeupRequest is defined and has a value greater than 0.⌋

Note: The OA TC10 [14] wake-up on dataline feature can be used instead of a wake-up line. The different timing behavior has to be considered. If using a wake-up line, the wake-up pulse is present for all connected ECUs at the same point in time. If using wake-up on dataline, the wake-up (WUP / WUR) has to be forwarded by the receiving Ethernet hardware (PHY). The wake-up is propagated over the network and therefore it is sequentially present for the receiving ECUs.

The following chapters describe the behavior in detail with respect to the OA TC10 service primitives and their modelling in the System Template.

3.3.5.8.1 Ethernet Communication Controller

[constr_3600] Setting of EthernetCommunicationController.slaveActAsPassiveCommunicationSlave

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The attribute EthernetCommunicationController.slaveActAsPassiveCommunicationSlave may only be set to TRUE, if the following conditions apply:

- the EthernetCommunicationController is not referenced by any NmNode in the role controller
- the EthernetCommunicationController aggregates at least one CouplingPort
- the couplingPortRole of that CouplingPort is set to standardPort
- the physicalLayerType of that CouplingPort is set to either _100BASE_T1, _1000BASE_T1 or _10BASE_T1S

In all other cases the attribute slaveActAsPassiveCommunicationSlave shall be set to FALSE or shall not be defined.⌋

Note: An Ethernet ECU which aggregates an EthernetCommunicationController that is acting as a passive slave is not using Nm frames for a synchronized shutdown. A synchronized shutdown has to be provided by the used Ethernet hardware. E.g. Ethernet hardware compliant with Open Alliance TC10.

Note further: It is only allowed for Ethernet ECUs which are NOT maintaining an Ethernet switch on the corresponding communication channel to act as a passive slave. A passive slave follows the communication request of the corresponding communication master.

[constr_3611] Existence of `EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `slaveQualifiedUnexpectedLinkDownTime` shall be defined if `slaveActAsPassiveCommunicationSlave` is set to TRUE.⌋

[TPS_SYST_03053] Semantics of `EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime` ⌈`EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime` specifies the time when an unexpected link down is evaluated as link down and indicated to the AUTOSAR communication stack.

If `slaveActAsPassiveCommunicationSlave` is set to FALSE or not defined, then the communication channel is not acting as a passive communication slave.⌋

The link down time qualification is used for an `EthernetCommunicationController` where `slaveActAsPassiveCommunicationSlave` is set to TRUE. The time should cover an error scenario where the corresponding communication master was not able to release the communication by triggering an Sleep.Request (e.g. communication master was unexpectedly reset).

3.3.5.8.2 Service primitives for wake-up

An Ethernet ECU which wants to communicate with other ECUs within the Ethernet switched network topology has to trigger a wake-up on the network to propagate the communication request to the communication partners. The ECU which triggers the wake-up is the requesting ECU. The requesting ECU calls the service primitive *Wakeup.request* to trigger a wake-up. The wake-up is transmitted via the physical dataline (e.g. 100BASE-T1) to the connected link partner. The wake-up could be a wake-up pulse (WUP), if the link to the connected link partner is down, or a wake-up request (WUR), if the link is up (link is already established).

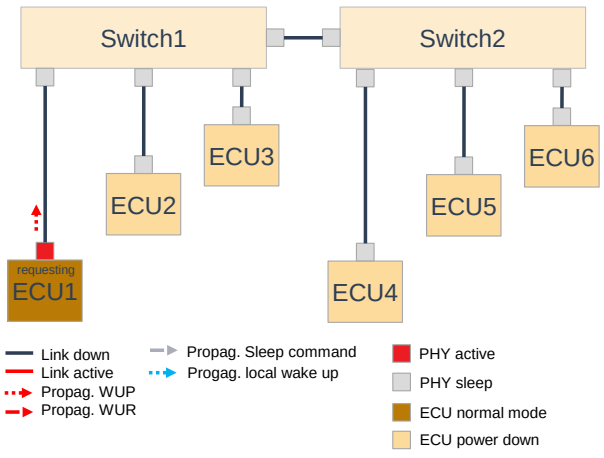

Figure 3.29: While the link is down on the connected dataline, the PHY of ECU1 transmit a wake-up on the network

If a WUP is transmitted by ECU1 and received by the Ethernet hardware (PHY) of Switch1, the receiving Ethernet hardware of Switch1 is woken up. After the receiving Ethernet hardware is initialized, the Switch1 is powered up (INHIBIT pin of power supply is set by the PHY (see OA TC10)) and a *Wakeup.indication* is generated.

(Note: If a WUR is transmitted, the receiving Ethernet hardware (PHY) generates immediately a *Wakeup.indication*, because the receiving Ethernet hardware is already initialized and in normal mode). Simultaneously the received wake-up could be forwarded as local wake-up to the neighboring PHYs of Switch1, if the PHYs are configured accordingly.

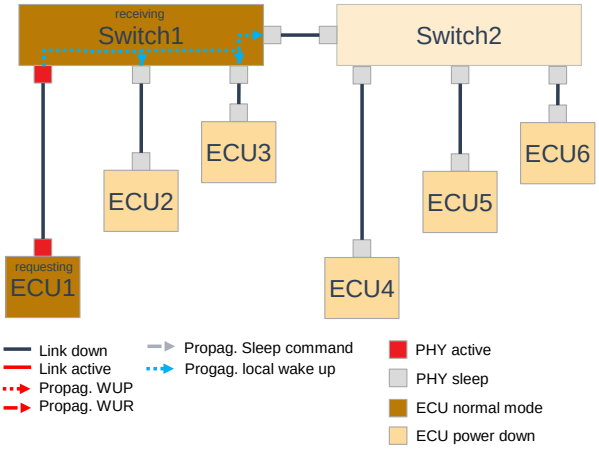

Figure 3.30: Switch1 is woken up by the PHY that received the wake-up. The received wake-up on the network is forwarded as local wake-up to the neighboring PHYs

In multi-PHY and Ethernet switch scenarios, a received wake-up is most likely forwarded to all connected link partners (other datalines) without host ECU (ECU that maintain a Ethernet switch) involvement to fulfill wake-up propagation time requirements.

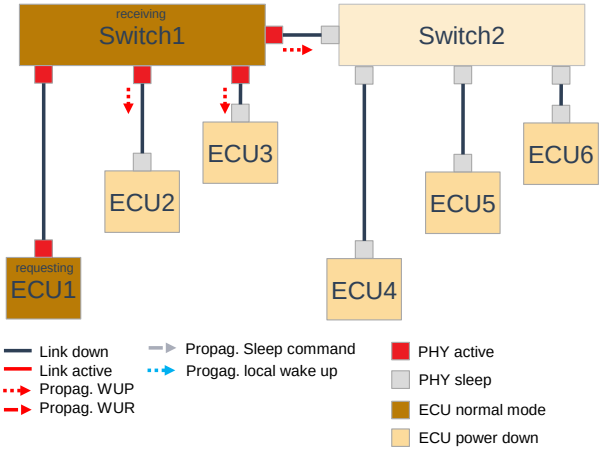

Figure 3.31: The propagated local wake-up is forwarded by the neighboring PHYs as WUP to the connected ECUs (ECU2, ECU3 and Switch2)

The forwarding behavior of each PHY can be modelled in the SystemTemplate. Each PHY is modelled as `CouplingPort`. Each `CouplingPort` could enable the OA TC10 compliant wake-up and sleep on dataline by defining a `wakeupSleepOnDataline-Config` reference.

3.3.5.8.3 Service primitives for sleep

An ECU which is ready to go to sleep calls the service primitive *Sleep.request* (ECU1). The sleep request is transmitted via the physical dataline (e.g. 100BASE-T1) to the connected communication partner as LPS (low power sleep signal), here Switch1. Please note: LPS are send as continues burst with respect to the specified timing in OA TC10.

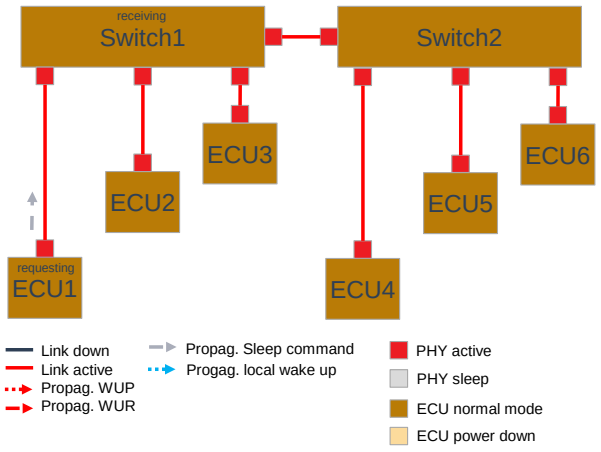

Figure 3.32: ECU1 triggers a *Sleep.request* and the Ethernet hardware of ECU1 sends LPS on the dataline

The receiving Ethernet hardware (PHY) generates a *Sleep.indication* to notify the receiving ECU (here Switch1) that the Ethernet hardware (PHY) of the requesting ECU1 is requesting to go to sleep. The receiving Ethernet hardware (PHY) of the connected communication partner follow the defined PHY power mode sequence described in OA TC10 and acknowledge the received *Sleep.request* with a *Sleep.request* back to requesting ECU (ECU1) with respect to the specified timings in OA TC10.

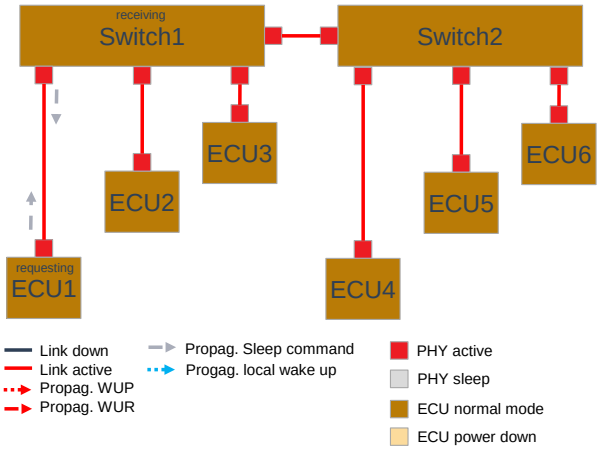

Figure 3.33: Ethernet hardware (PHY) of Switch1 receives LPS and acknowledges the indicated *Sleep.request* of ECU1 by sending back LPS to ECU1

If the requesting ECU (ECU1) received a *Sleep.request* from the receiving ECU (Switch1) within the specified time of OA TC10 the Ethernet hardware of both ECUs (requesting and receiving ECU) transit to sleep state, i.e. the Ethernet hardware of both, ECU1 and Switch1 transit to a low power down mode and the link connection is down.

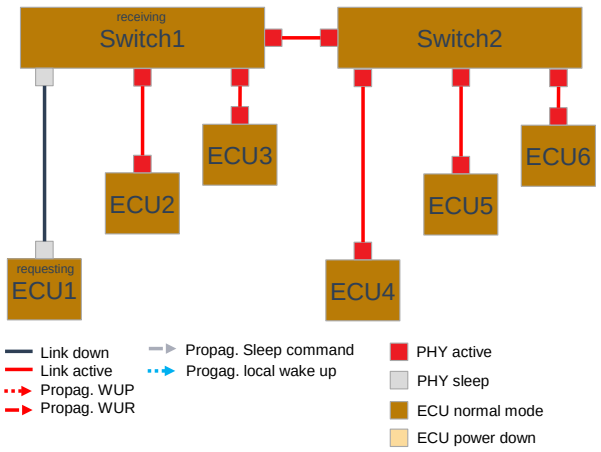

Figure 3.34: ECU1 send and receive LPS with respect to the specified timings in OA TC10 and therefore the Etherent hardware (PHY) of ECU1 and Switch1 transit to a low power sleep mode and the link of the dataline is down

ECUs which participate in an AUTOSAR network management (NM), switch off their communication hardware according to already defined NM shutdown process. Thus, an additional handling in the AUTOSAR stack for an Ethernet communication channel to react on a *Sleep.indication* upon the network management shutdown process is superfluous. The *Sleep.indication* is only evaluated by ECUs which have `EthernetCommunicationController.slaveActAsPassiveCommunicationSlave` set to TRUE (see details in chapter 3.3.5.8.4).

If the NM decides to go to sleep, then the linked ECUs may switch off their connected hardware at slightly different points in time (e.g. Ethernet switches switch off their

Ethernet switch ports with a configured time delay (see couplingPortSwitchoff-Delay)). The ECU which earlier switches off the communication hardware will trigger a *Sleep.request*. The connected Ethernet hardware (PHY) of the communication partner will go to sleep according to the defined PHY power mode sequence described in OA TC10. Afterwards the ECU which later switches off its Ethernet hardware has to check the power mode of its Ethernet hardware (PHY). If the Ethernet hardware is already in sleep mode, then the ECU will leave the hardware state as it is and do not trigger a *Sleep.request*. Otherwise the ECU will trigger a *Sleep.request*.

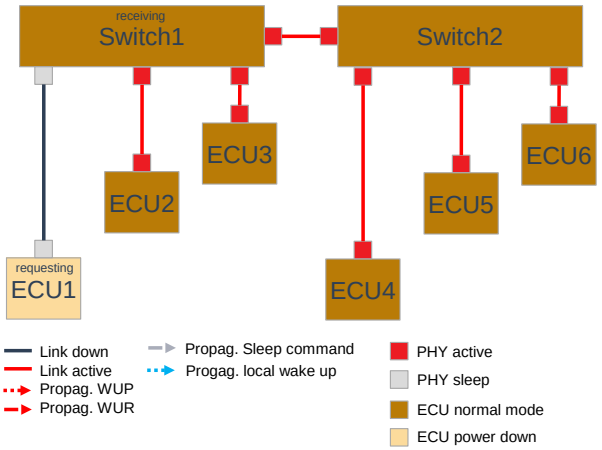

Figure 3.35: ECU1 (requesting ECU) is already in power down mode, while Switch1 is waiting until couplingPortSwitchoffDelay has expired. If couplingPort-SwitchoffDelay has expired, Switch1 will detect that the Ethernet hardware is already in sleep state and therefore will NOT trigger a *Sleep.request*

According to the specified service primitive *Sleep.AbortRequest*, the receiving ECU could decide to reject a *Sleep.request* via *Sleep.AbortRequest*. AUTOSAR does NOT support the service primitive *Sleep.AbortRequest*, since the AUTOSAR network management (NM) provide a synchronized shut down of the ECUs. Thus, there is no need to enable an explicit reject of a received *Sleep.request*.

Even though the explicit rejection of a *Sleep.request* is NOT supported, the ECU which requested the *Sleep.request* could be indicated that the *Sleep.request* was NOT accepted by the Ethernet hardware (PHY) of the receiving ECU. If the *Sleep.request* was not acknowledge by Ethernet hardware (PHY) of the receiving ECU, the requesting ECU is signalled via the service primitive *SleepFail.indication*. Reason is an error scenario, where the *Sleep.request* was not received by Ethernet hardware of the receiving ECU or the acknowledgement of the *Sleep.request* back to the requesting ECU was lost on the network (e.g. disturbance of the LPS by a EMC pulse, loose contact of the dataline ... a.s.o.). The ECUs are always evaluating if a *SleepFail.indication* was indicated.

The handling of a detected *SleepFail.indication* is modelled by defining the repetition of a *Sleep.request* (sleepRepetitionsOfSleepRequest) and the delay to re-trigger a *Sleep.request* (sleepRepetitionDelayOfSleepRequest). If the count of repetitions is exceed and *SleepFail.indication* is still signalled, the Ethernet hardware (PHY)

is forced to transit to a sleep state and indicate the upper layer of the AUTOSAR communication stack a *Sleep.indication*. This should prevent the requesting ECU to be kept awake if no *Sleep.request* was acknowledged by the receiving ECU.

3.3.5.8.4 Ethernet communication channel that act as passive communication slave

ECUs which are connected to communication channels that do not participate in the AUTOSAR network management can be controlled by a master / slave relationship. The connected Ethernet communication channel is controlled by using the following service primitives: *Wakeup.request*, *Wakeup.indication*, *Sleep.request*, and *Sleep.indication*.

A host ECU (Ethernet ECU which maintains an Ethernet switch) requests to wake-up an Ethernet communication channel of a connected ECU where the corresponding `EthernetCommunicationController` has set `slaveActAsPassiveCommunicationSlave` to TRUE (passive communication slave) by triggering a *Wakeup.request*. The wake-up brings the Ethernet hardware (PHY) of the receiving ECU from a sleep mode to a normal mode. The receiving ECU is powered on and its application may provide data which is consumed by the requesting ECU (communication master).

If the requesting ECU (communication master) decides to shutdown the communication channel, the host ECU request the communication channel to go to sleep by triggering a *Sleep.request*. The requested *Sleep.request* is received by the connected ECU on the communication channel with acts as a communication slave. The received *Sleep.request* will be acknowledged by the Ethernet hardware (PHY) of the receiving ECU (passive communication slave) and simultaneously a *Sleep.indication* is signalled. The receiving ECU evaluates the *Sleep.indication*.

If a *Sleep.indication* is detected, this indication is forwarded to the communication stack and the affected communication channel is released. Additionally the application could be indicated about the communication release to execute some shut down actions. Thus, an Ethernet communication channel which acts as passive communication slave, always follows the *Wakeup.indication*/*Sleep.indication* of the corresponding communication master. To cover an error scenario where the communication master could not trigger a *Sleep.request*, due to unexpected reset, the receiving ECU, where an Ethernet channel is acting as passive communication slave, has to detect a link down. If an unexpected link down last longer than `slaveQualifiedUnexpectedLinkDownTime`, the Ethernet channel, which is acting as passive communication slave, is released autonomously by the receiving ECU.

3.3.6 10BASE-T1S Ethernet

10BASE-T1S is a 10Mbps Single pair Ethernet physical layer technology that is specified by IEEE 802.3cg. The multi-drop feature of 10BASE-T1S allows the usage of

a single bus-line to connect ECUs. The PLCA (Physical Layer Collision Avoidance) mechanism avoids collision on PHY level and offers a fair medium access to every participant.

All nodes are identified on the bus via nodeIDs (plcaLocalNodeId) starting from 0 (standardized as the referenced head-node). The Head-Node on the PLCA based network controls the traffic on the bus.

[TPS_SYST_02299] Modeling of 10Base-T1S networks ⌈The modeling of a 10BASE-T1S bus in a System Description is done with a CouplingPortConnection that points with the nodePort reference to CouplingPorts that represent the 10Base-T1S PHYs connected to the network.⌋

[constr_5157] Mixing of Point-To-Point and Multi-Drop is not allowed in a CouplingPortConnection

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The CouplingPortConnection is allowed to reference a CouplingPort either:

- in the role firstPort and/or secondPort or
- in the role nodePort

⌋

In other words a CouplingPortConnection shall not use the firstPort and/or secondPort reference (Point-to-Point) and the nodePort reference (Multi-Drop) at the same time.

The PLCA runs cycles on the network. Within each cycle each node with a unique plcaLocalNodeId is assigned with a transmit opportunity. The plcaTransmit-OpportunityTimer is identical for all nodes and is therefore configured in the CouplingPortConnection.

The cycle starts with a BEACON that is sent by the head node. During the transmit opportunity the node is able to transmit data or to skip its transmit opportunity. If a node does not need to transmit data the next node is allowed to start its transmit opportunity earlier.

At each BEACON reception client nodes restart their currentNodeID counter and increment it every time a Transmit Opportunity is used or yield. If the currentNodeID matches the plcaLocalNodeId, the corresponding node is allowed to transmit an Ethernet frame for this Transmit Opportunity. The plcaTransmitOpportunity-Timer is reset for every Transmit Opportunity once the transceiver detects activity on the bus and recognizes the transmission of data. At each Transmit Opportunity no more than one single Ethernet frame will be sent. On the other hand, if a node has the necessity to sent more packets in one Transmit Opportunity, a burst mode can be used. The burst mode is configured by plcaMaxBurstTimer and plcaMaxBurstCount.

[TPS_SYST_02300] Enabling of PLCA on a `CouplingPort` ⌈The PLCA (Physical Layer Collision Avoidance) mechanism is enabled on a `CouplingPort` if the `plcaProps` are aggregated by the same `CouplingPort`.⌋

Class	**PlcaProps**			
Note	This meta-class allows to configure the PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the CouplingPort (PHY).			
Base	*ARObject*			
Aggregated by	CouplingPort.plcaProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
plcaLocalNodeId	PositiveInteger	0..1	attr	This attribute defines the node ID when the PLCA mode for 10BASE-T1S is used.
plcaMaxBurstCount	PositiveInteger	0..1	attr	Defines maximum packets allowed to be transmitted within a TO. This configuration can be different from one ECU to another within the PLCA mixed segment.
plcaMaxBurstTimer	PositiveInteger	0..1	attr	Limits the burst frames in bit time. This configuration can be different from one ECU to another within the PLCA mixed segment. For PLCA burst mode to work properly this timer should be set greater than one IPG.

Table 3.126: PlcaProps

[constr_5158] Usage of `plcaProps` only allowed on 10BASE-T1S networks

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `CouplingPort` is allowed to aggregate `plcaProps` only if:

- the `CouplingPort.physicalLayerType` is set to 10BASE-T1S
- the `CouplingPort.macLayerType` is set to `xMII`
- the `CouplingPort` is referenced by a `CouplingPortConnection` with the `nodePort` reference.

⌋

Please note that it is possible to have a mix network with PLCA and CSMA/CD configured nodes.

[TPS_SYST_02301] CSMA/CD configured nodes on a 10BASE-T1S network ⌈If a `CouplingPort` is referenced by a `CouplingPortConnection` with the `nodePort` reference and the `CouplingPort.physicalLayerType` is set to 10BASE-T1S and this `CouplingPort` does not aggregate `plcaProps` then this `CouplingPort` represents a CSMA/CD configured node in a 10BASE-T1S network.⌋

The following example shows a configured 10BASE-T1S network with four nodes in a System Description.

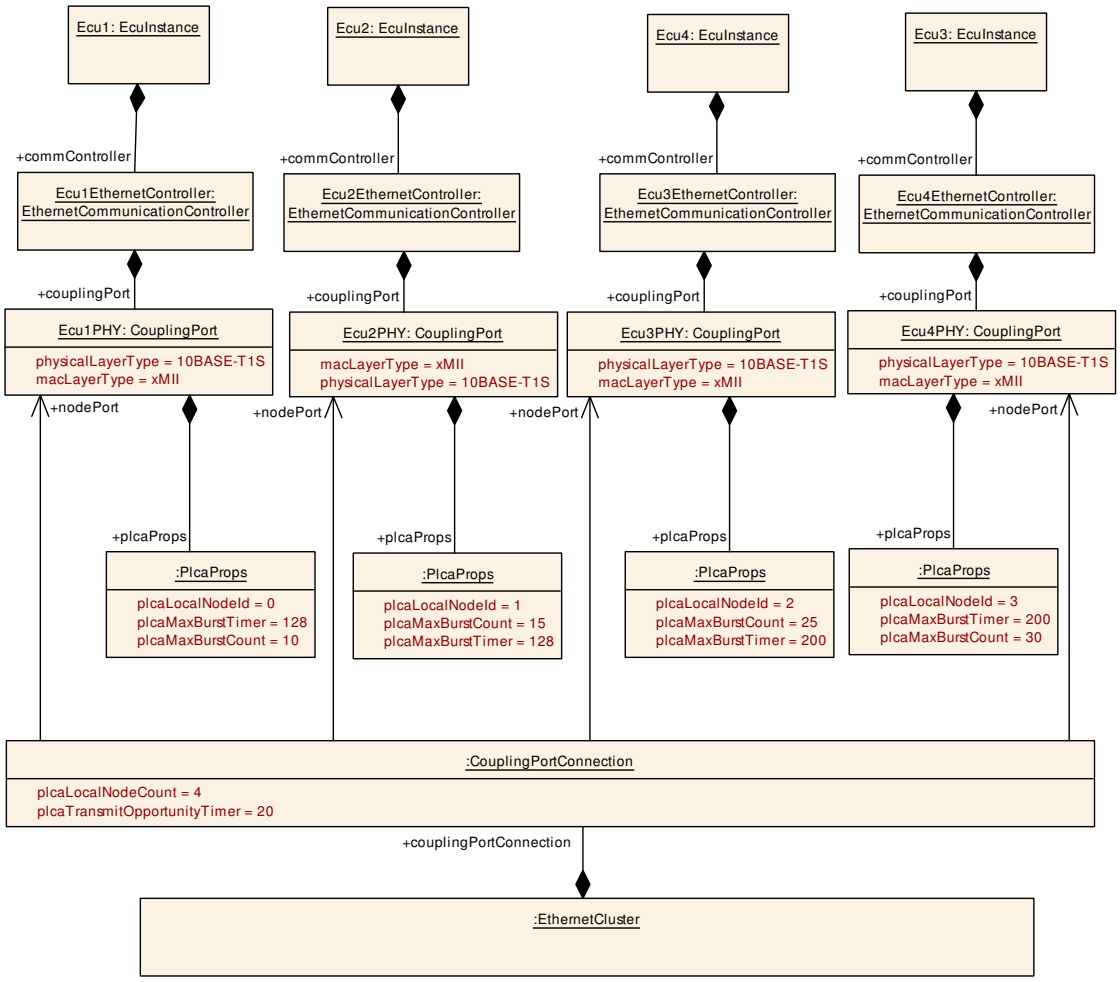

Figure 3.36: Example for a 10BASE-T1S network description

[constr_5159] Mandatory `CouplingPortConnection` settings if multi-drop feature is used

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingPortConnection` uses the `nodePort` reference then the attribute `CouplingPortConnection.plcaLocalNodeCount` and the attribute `CouplingPortConnection.plcaTransmitOpportunityTimer` shall be set to a value.⌋

[constr_5160] Mandatory `PlcaProps` settings if multi-drop feature is used

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `CouplingPort` is referenced by a `CouplingPortConnection` in the role `nodePort` then the `CouplingPort` shall aggregate the `PlcaProps` and the following attributes shall be set to a value:

- `plcaMaxBurstCount`
- `plcaMaxBurstTimer`
- `plcaLocalNodeId`

⌋

3.3.7 Connectivity Association for MACsec and CANsec

A Connectivity Association is a logical association between two or more communication partners that defines the security properties for the communication between these partners.

The Connectivity Association is used to design the secure communication from a topological point of view. The participating communication partners are associated with each other to form the Connectivity Association.

MACsec [IEEE Std 802.1AE, IEEE Std 802.1X] and CANsec [CiA 613-2], on the implementation level, are defined by specifying which messages are NOT secured. This is called bypassing the security for a specific set of messages. The Connectivity Association is used to define which messages are secured. And if the messages are secured, then the Connectivity Association also defines which security credential to be applied for that specific message. The Bypass definition specifies which messages are not secured. Bringing both definitions together (Connectivity Association and Bypass communication) there needs to be a full coverage of all the possible messages in the communication channel.

The abstract class `SecurityConnectivityAssociation` is the base class to specify the security requirements and bypass requirements for the communication to be secured using MACsec or CANsec.

Class	***SecurityConnectivityAssociation*** (abstract)			
Note	Definition of a Connectivity Association for MACsec and CANsec.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*AbstractMacSecCommunication*, CanSecConnectivityAssociation			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.127: SecurityConnectivityAssociation

3.3.7.1 MACsec Connectivity Association

The specific Connectivity Association for MACsec is `MacSecConnectivityAssociation`. The modeling approach is to reference the involved `CouplingPort`s that define the members of the Connectivity Association in the role `MacSecConnectivityAssociation.couplingPort`. Additionally a set of VLANs is referenced in the role `vlan` for which the communication shall be secured using the MACsec protocol.

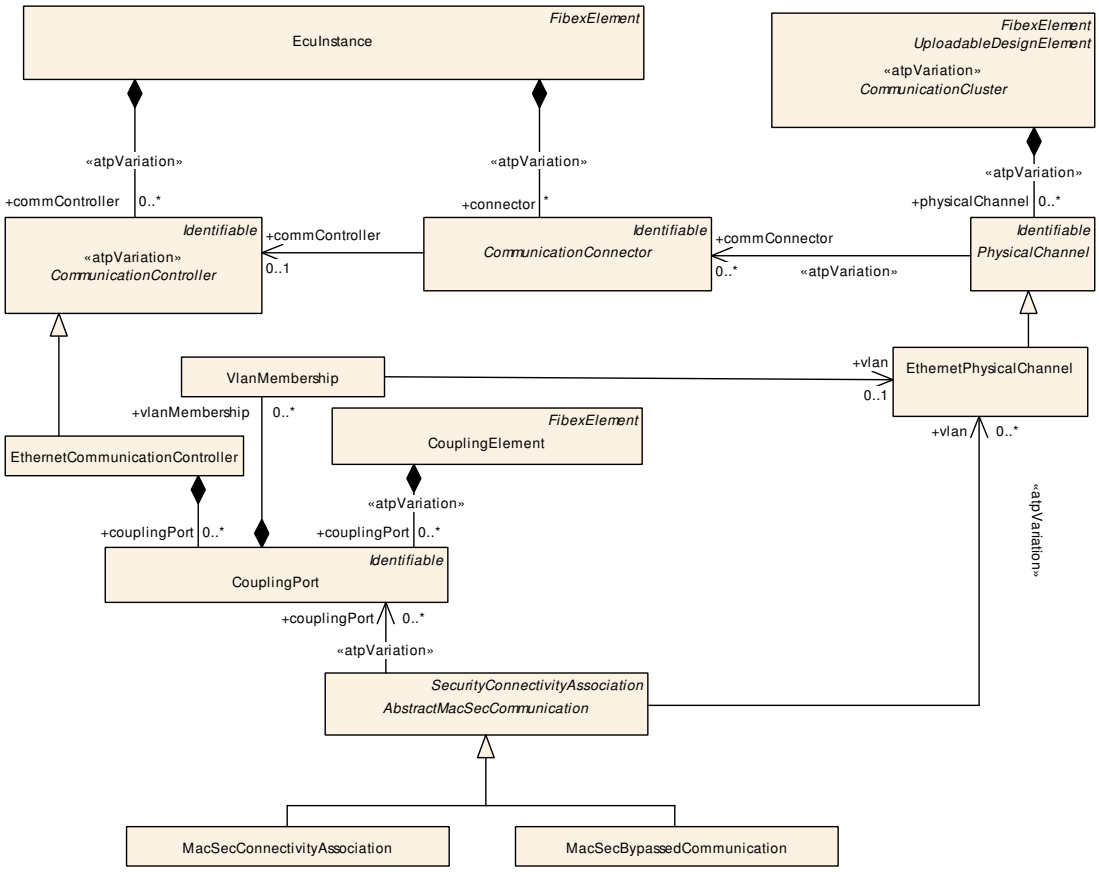

Figure 3.37: MACsec Connectivity Association

Class	***AbstractMacSecCommunication*** (abstract)			
Note	Abstract class for the definition of a Connectivity Association and bypass for MACsec.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *SecurityConnectivityAssociation*			
Subclasses	MacSecBypassedCommunication, MacSecConnectivityAssociation			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort	CouplingPort	*	ref	Reference to the CouplingPorts this MacSec communication definition is handling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=couplingPort, couplingPort.couplingPort, couplingPort.variationPoint.shortLabel vh.latestBindingTime=postBuild
vlan	EthernetPhysical Channel	*	ref	Reference to the VLANs this MacSec communication definition is handling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=vlan, vlan.ethernetPhysicalChannel, vlan.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.128: AbstractMacSecCommunication

Class	MacSecConnectivityAssociation			
Note	Definition of a Connectivity Association for MACsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *AbstractMacSecCommunication*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 3.129: MacSecConnectivityAssociation

Two setups are supported by the `MacSecConnectivityAssociation`, MACsec for switched Ethernet networks and MACsec for 10BASE-T1S buses.

In case of a switched Ethernet network, the `MacSecConnectivityAssociation` is defined for pairs of Ethernet ports: One `couplingPort` refers to the Ethernet switch `CouplingPort` and the other `couplingPort` refers to the `CouplingPort` of the respective connected `EcuInstance`.

[constr_9421] Exactly two `MacSecConnectivityAssociation.couplingPort` references for a switched Ethernet connection

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If a `AbstractMacSecCommunication` (`MacSecConnectivityAssociation` or `MacSecBypassedCommunication`) refers to a `CouplingPort` in the role `couplingPort` that is aggregated by a `CouplingElement` with `couplingType` set to `switch`, then exactly one other `couplingPort` reference shall reference a `CouplingPort` that is aggregated by an `EthernetCommunicationController`.⌋

The topology example shown in Figure 3.38 illustrates a switched Ethernet network where the Ethernet switch is connecting three ECUs. The `MacSecConnectivity-Association`s are defined for each pair of Ethernet switch port and connected ECU port to specify the security configuration for the communication. The model example shown in Figure 3.39 illustrates the 3 `MacSecConnectivityAssociation`s for the pairs of Ethernet switch port and connected ECU port.

Each `MacSecConnectivityAssociation` can refer to a set of `Ethernet-PhysicalChannel`s (VLANs) using the reference `vlan` for which the communication shall be secured using MACsec.

[constr_9422] Consistent reference to a VLAN using `AbstractMacSec-Communication.vlan`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If an `EthernetPhysicalChannel` is referenced by a `AbstractMacSec-Communication` (`MacSecConnectivityAssociation` or `MacSecBypassed-Communication`) in the role `AbstractMacSecCommunication.vlan`, then this `EthernetPhysicalChannel` shall be referenced by the `VlanMembership` of the `CouplingPort` that is referenced by the `AbstractMacSecCommunication` in the role `couplingPort`.⌋

The Bypass for MACsec is defined by the MacSecBypassedCommunication which references the involved CouplingPorts in the role MacSecBypassed-Communication.couplingPort and additionally references a set of VLANs in the role vlan for which the communication shall not be secured using the MACsec protocol.

Note that [constr_9421] and [constr_9422] also apply to the MacSecBypassed-Communication definitions.

Class	**MacSecBypassedCommunication**			
Note	Definition of bypassed communication for MACsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *AbstractMacSecCommunication*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 3.130: MacSecBypassedCommunication

The reason there are both model approaches supported (MacSecConnectivity-Association and MacSecBypassedCommunication) is that a check can be applied such that every message communication on a CouplingPort is either secured or bypassed. This check can be applied by checking that the union of all VLANs referenced by MacSecConnectivityAssociation and MacSec-BypassedCommunication for a specific CouplingPort is equal to the set of all VLANs referenced by the VlanMembership of that CouplingPort.

[constr_9423] Coverage of all VLANs referenced by a CouplingPort by MacSecConnectivityAssociation and MacSecBypassedCommunication

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CouplingPort is referenced by at least one AbstractMacSec-Communication (MacSecConnectivityAssociation or MacSecBypassed-Communication) in the role couplingPort, then all EthernetPhysical-Channels referenced by the VlanMembership of that CouplingPort shall be referenced by at least one AbstractMacSecCommunication (MacSecConnectivity-Association or MacSecBypassedCommunication) in the role AbstractMac-SecCommunication.vlan.⌋

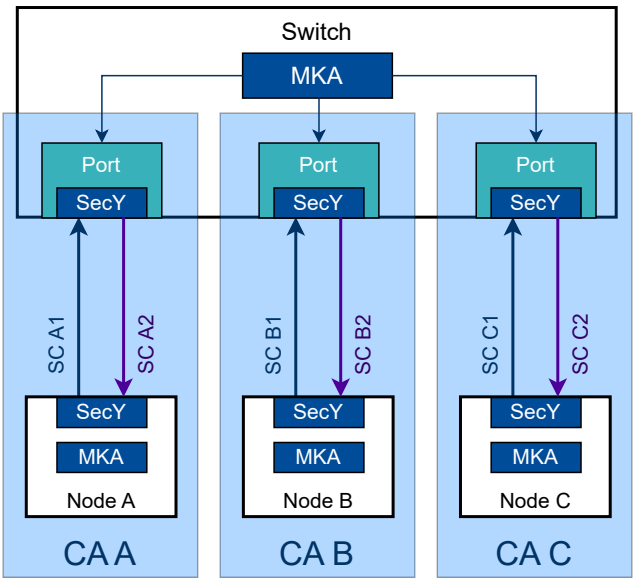

Figure 3.38: Example topology of switched Ethernet network

Note that the "Secure Channels" illustrated in Figure 3.38 are runtime entities that are established between the communication partners of a `MacSecConnectivityAssociation` based on the defined security configuration. The "Secure Channels" are not explicitly modelled in the System Description, but they are implicitly defined by the `MacSecConnectivityAssociation` and the referenced security configuration.

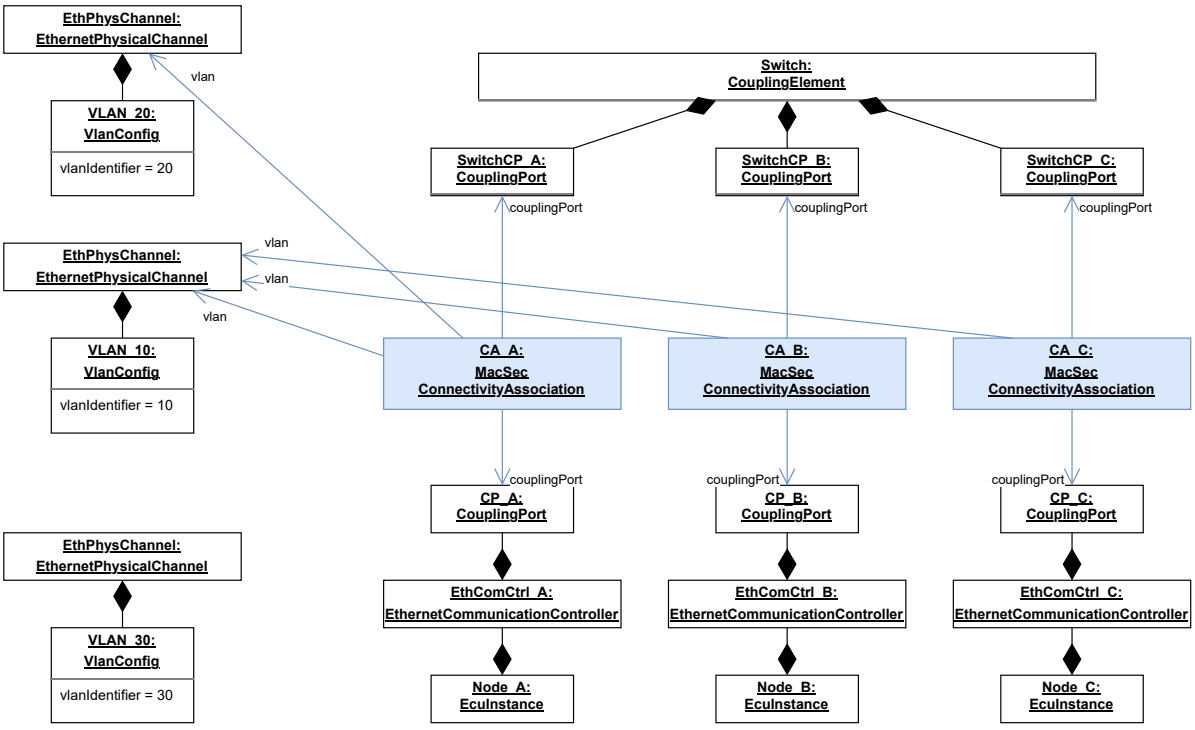

Figure 3.39: `MacSecConnectivityAssociation` for a switched Ethernet network

The MacSecBypassedCommunication definitions in Figure 3.40 indicate the communication that is not secured by MACsec (bypassed). The MacSecBypassedCommunications reference the same CouplingPorts as the MacSecConnectivityAssociation in the role couplingPort. The referenced VLANs in the role vlan define the communication that is bypassed by MACsec.

VLAN 10 is in this example secured on all 3 links, thus there is no need to include *VLAN 10* in the MacSecBypassedCommunication definitions. *VLAN 20* is only secured on the link between *Node A* and the Ethernet switch, thus it is included in the MacSecBypassedCommunication definitions for the links between *Node B* and the Ethernet switch and between *Node C* and the Ethernet switch. *VLAN 30* is not secured on any of the links, thus it is included in all three MacSecBypassedCommunication definitions.

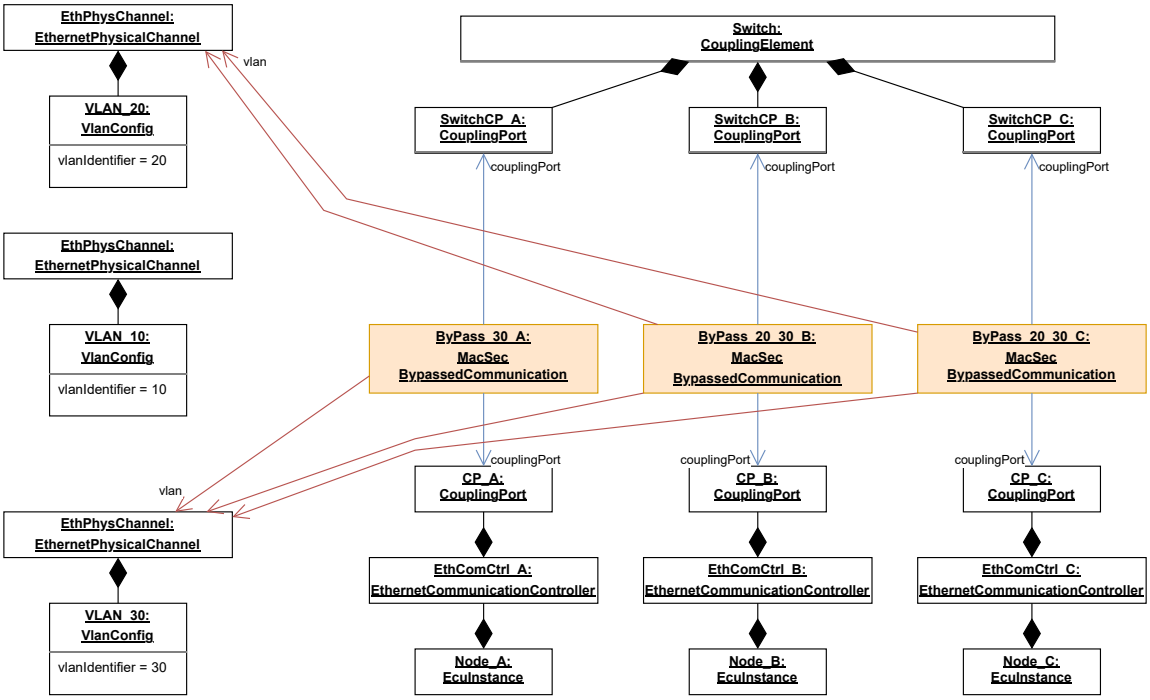

Figure 3.40: MacSecBypassedCommunication for a switched Ethernet network

A 10BASE-T1S bus is modelled according to [TPS_SYST_02299] with a CouplingPortConnection that references the involved CouplingPorts in the role nodePort.

A MacSecConnectivityAssociation is defined for those nodes that shall participate in the MacSecConnectivityAssociation by referencing the 10BASE-T1S CouplingPorts in the role AbstractMacSecCommunication.couplingPort.

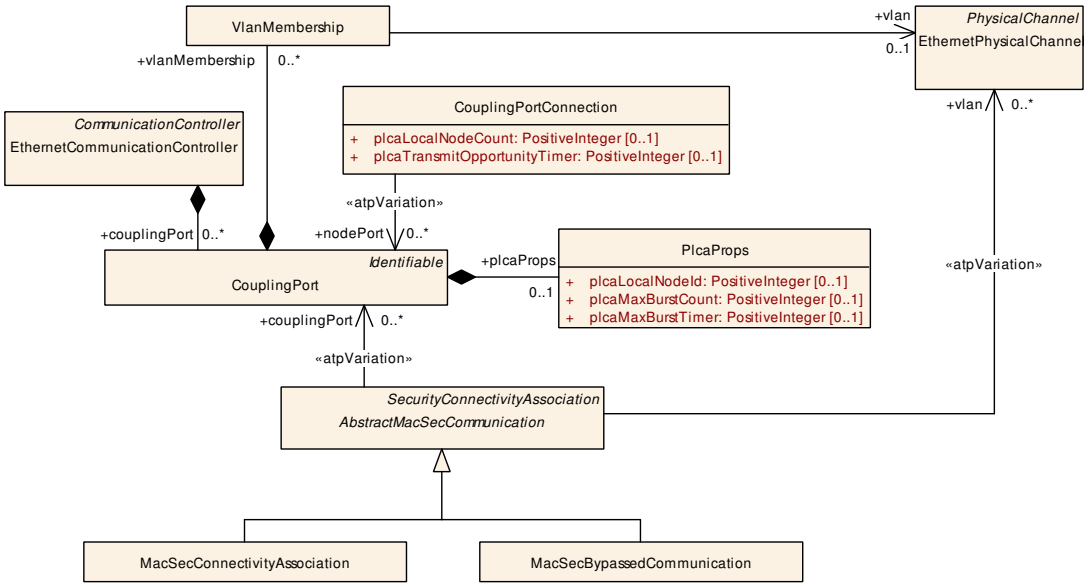

Figure 3.41: 10BASE-T1S MACsec Connectivity Association

[constr_9424] 10BASE-T1S `MacSecConnectivityAssociation CouplingPortConnection` reference consistency

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If at least one `CouplingPort` that is referenced by a `AbstractMacSecCommunication` in the role `AbstractMacSecCommunication.couplingPort` is referenced by a `CouplingPortConnection` with the `nodePort` reference, then all `CouplingPort`s referenced by that `AbstractMacSecCommunication` in the role `AbstractMacSecCommunication.couplingPort` shall also be referenced by the same `CouplingPortConnection` with the `nodePort` reference.⌋

Note that for the `AbstractMacSecCommunication.vlan` reference in the scope of 10BASE-T1S also [constr_9422] applies.

The topology example shown in Figure 3.42 illustrates a 10BASE-T1S bus where 4 ECUs are connected using 10BASE-T1S. Two `MacSecConnectivityAssociation`s are defined for this topology. The model example shown in Figure 3.43 illustrates the 2 `MacSecConnectivityAssociation`s and their relation to VLANs.

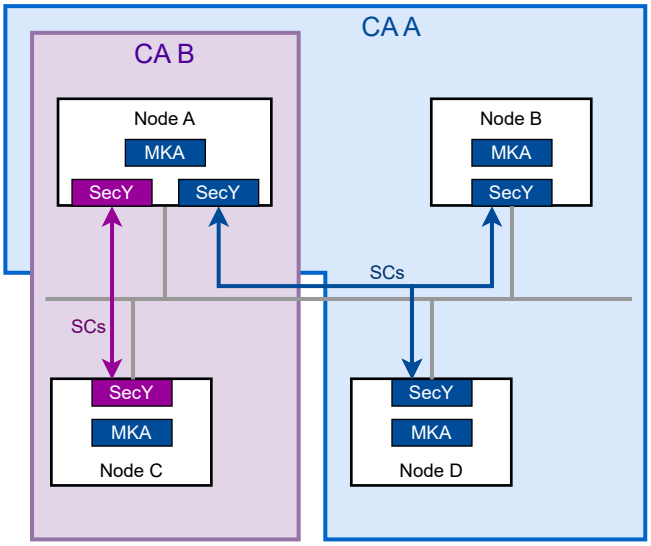

Figure 3.42: Example topology of 10BASE-T1S bus

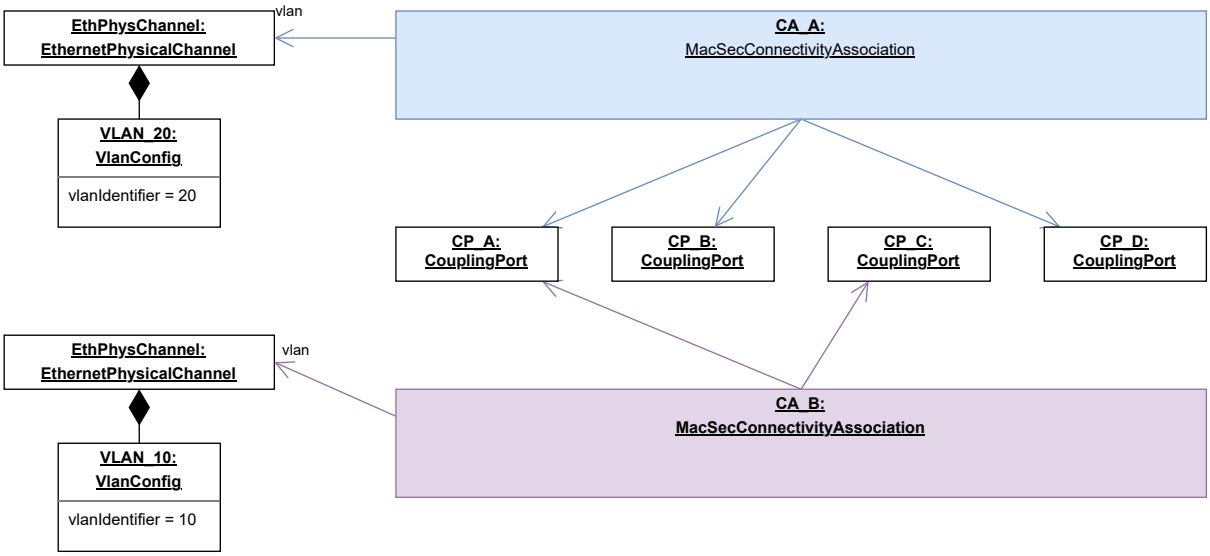

Figure 3.43: `MacSecConnectivityAssociation` for a 10BASE-T1S bus

The `MacSecBypassedCommunication` definition is shown in Figure 3.44. The `MacSecBypassedCommunication` references the same `CouplingPort`s as the `MacSecConnectivityAssociation` in the role `couplingPort`. The referenced VLANs in the role `vlan` define the communication that is bypassed by MACsec.

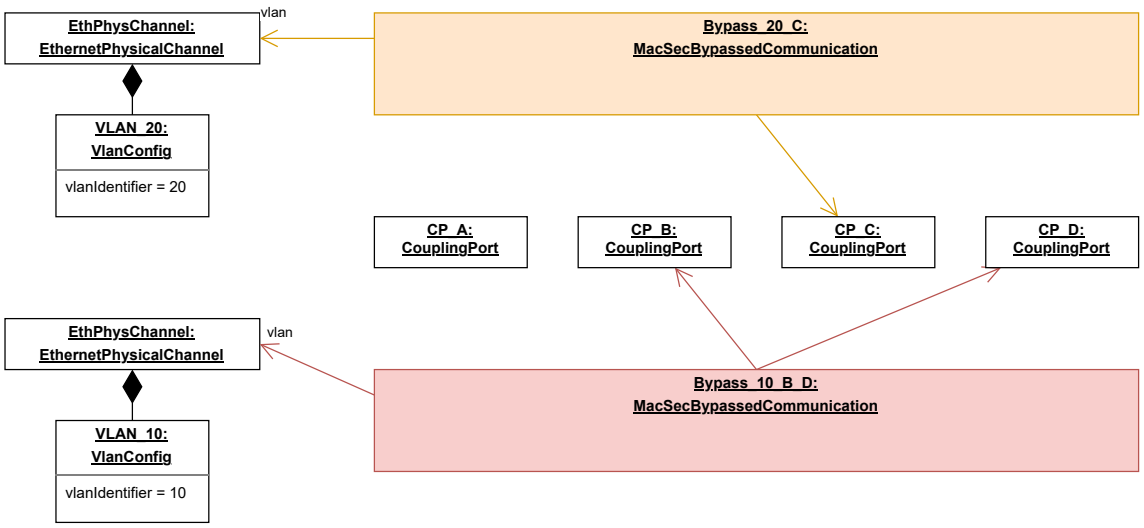

Figure 3.44: `MacSecBypassedCommunication` for a 10BASE-T1S bus

3.3.7.2 CANsec Connectivity Association

The specific Connectivity Association for CANsec is `CanSecConnectivityAssociation`. The modeling approach is to reference the involved `CanCommunicationConnector`s that define the members of the Connectivity Association in the role `CanSecConnectivityAssociation.canCommunicationConnector`.

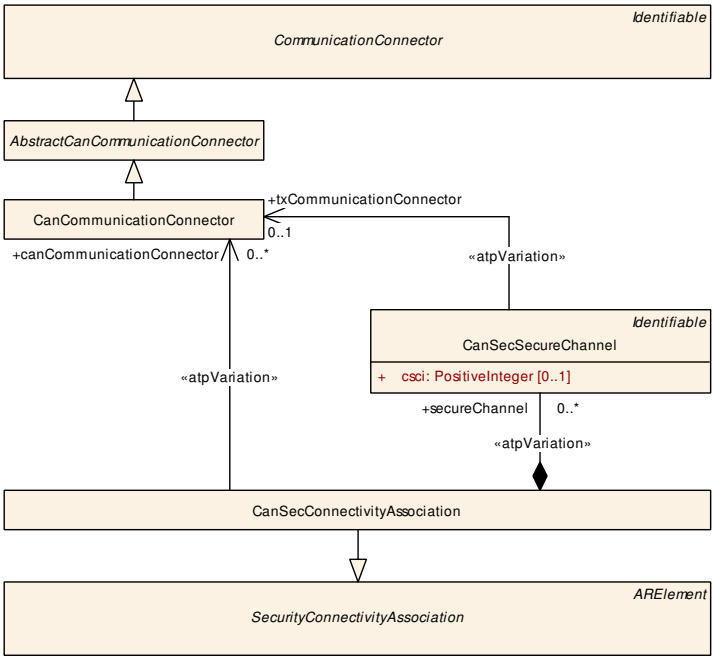

Figure 3.45: CANsec Connectivity Association and Secure Channel

Class	CanSecConnectivityAssociation			
Note	Definition of a Connectivity Association for CANsec. **Tags:** atp.recommendedPackage=SecurityConnectivityAssociations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *SecurityConnectivityAssociation*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
can Communication Connector	CanCommunication Connector	*	ref	Reference to the CanCommunicationConnectors this Can SecConnectivityAssociation is communicating on. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canCommunicationConnector, can CommunicationConnector.canCommunicationConnector, canCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild
secureChannel	CanSecSecureChannel	*	aggr	Collection of CanSecSecureChannels in the scope of this CanSecConnectivityAssociation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secureChannel, secureChannel.shortName, secureChannel.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.131: CanSecConnectivityAssociation

Class	CanSecSecureChannel			
Note	Definition of a CanSecSecureChannel.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanSecConnectivityAssociation.secureChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
csci	PositiveInteger	0..1	attr	CANsec Secure Channel Identifier.
tx Communication Connector	CanCommunication Connector	0..1	ref	Reference to the transmitting communication connector. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=txCommunicationConnector, tx CommunicationConnector.canCommunicationConnector, txCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.132: CanSecSecureChannel

[constr_9425] Existence of `CanSecSecureChannel.csci`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanSecConnectivityAssociation` the attribute `CanSecSecureChannel.csci` shall exist.⌋

The topology example shown in Figure 3.46 illustrates a CAN XL bus where 4 ECUs are connected. The `CanSecConnectivityAssociation` includes 3 `canCommunicationConnector` references to the involved `CanCommunicationConnector`s of the connected ECUs. The model example shown in Figure 3.47 illustrates the modeling for this topology.

This simple example only uses one `CanSecSecureChannel` per transmitting ECU. Implicitly all participants on the CAN bus are able to receive the messages, and those ECUs in the `CanSecConnectivityAssociation` are able to decrypt the messages.

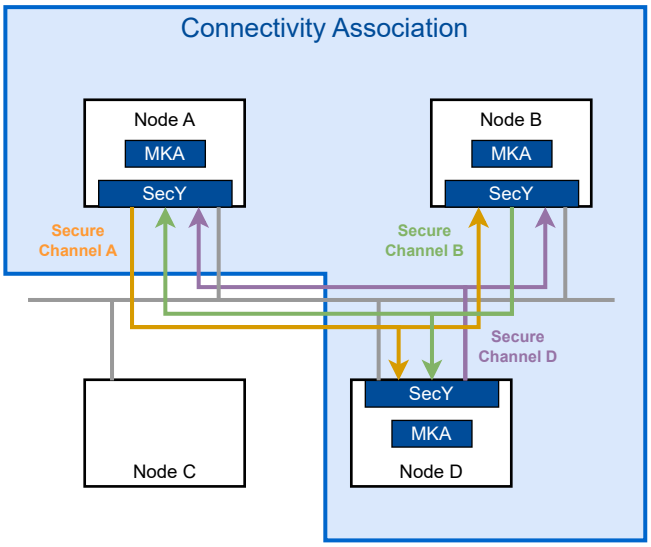

Figure 3.46: Example topology of CANsec bus

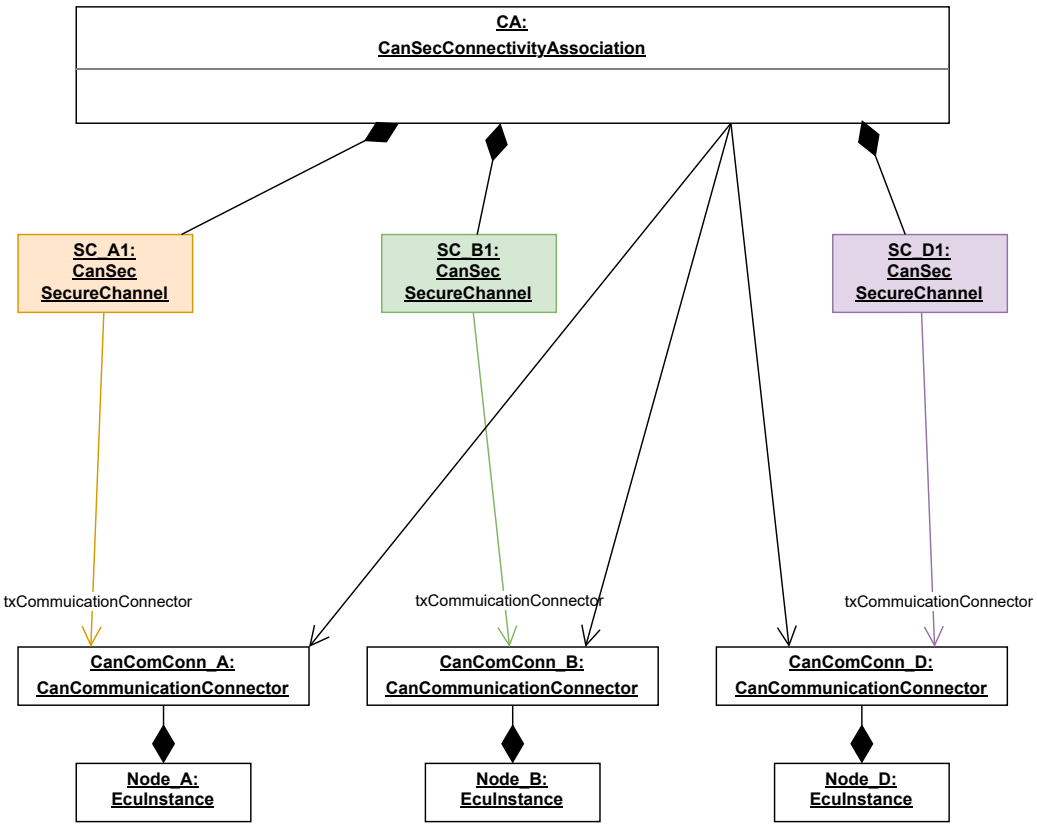

Figure 3.47: `CanSecConnectivityAssociation` for a CANsec bus

In contrast to the MACsec modeling approach, where there is no explicit model for secure channels, the CANsec modeling approach uses the meta-class CanSecSecureChannel, which is aggregated by the CanSecConnectivityAssociation in the role secureChannel.

The rationale for the explicit modeling of CanSecSecureChannel is to provide means for explicit assignment of messages (frames and SoAd messages) to these channels. This is relevant for the CAN arbitration priority handling definition.

Which messages are transported on which CanSecSecureChannel is defined by the MessageToCanSecSecureChannelMapping. This meta-class references the messages (frames and/or SoAd messages) in the role canFrameTriggering and soConIPduIdentifier, and the CanSecSecureChannel in the role secureChannel.

For CANsec there is the possibility to define messages that are bypassed by CANsec. This is done by the MessageToBypassCanSecCommunicationMapping which references the messages (frames and/or SoAd messages) in the role canFrameTriggering and soConIPduIdentifier. The bypassing for CANsec is a definition that is not related to dedicated CanSecSecureChannels or CanSecConnectivityAssociations. The MessageToBypassCanSecCommunicationMapping refers to the whole CanCommunicationConnector in the role canCommunicationConnector where the bypassing shall be applied.

The reason there are both model approaches supported (MessageToCanSecSecureChannelMapping and MessageToBypassCanSecCommunicationMapping) is that a check can be applied such that every message communication on a CanCommunicationConnector is either secured or bypassed.

[constr_9426] Coverage of all CanFrameTriggerings and SoConIPduIdentifiers in the scope of a CanCommunicationConnector by MessageToCanSecSecureChannelMapping and MessageToBypassCanSecCommunicationMapping

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a CanFrameTriggering or SoConIPduIdentifier is referenced by a AbstractMessageToSecureChannelMapping (MessageToCanSecSecureChannelMapping or MessageToBypassCanSecCommunicationMapping) in the role canFrameTriggering or soConIPduIdentifier, then every CanFrameTriggering and SoConIPduIdentifier that is aggregated by the same PhysicalChannel where that PhysicalChannel refers to the same CanCommunicationConnector shall be referenced by either a MessageToCanSecSecureChannelMapping or a MessageToBypassCanSecCommunicationMapping in the role canFrameTriggering or soConIPduIdentifier.⌋

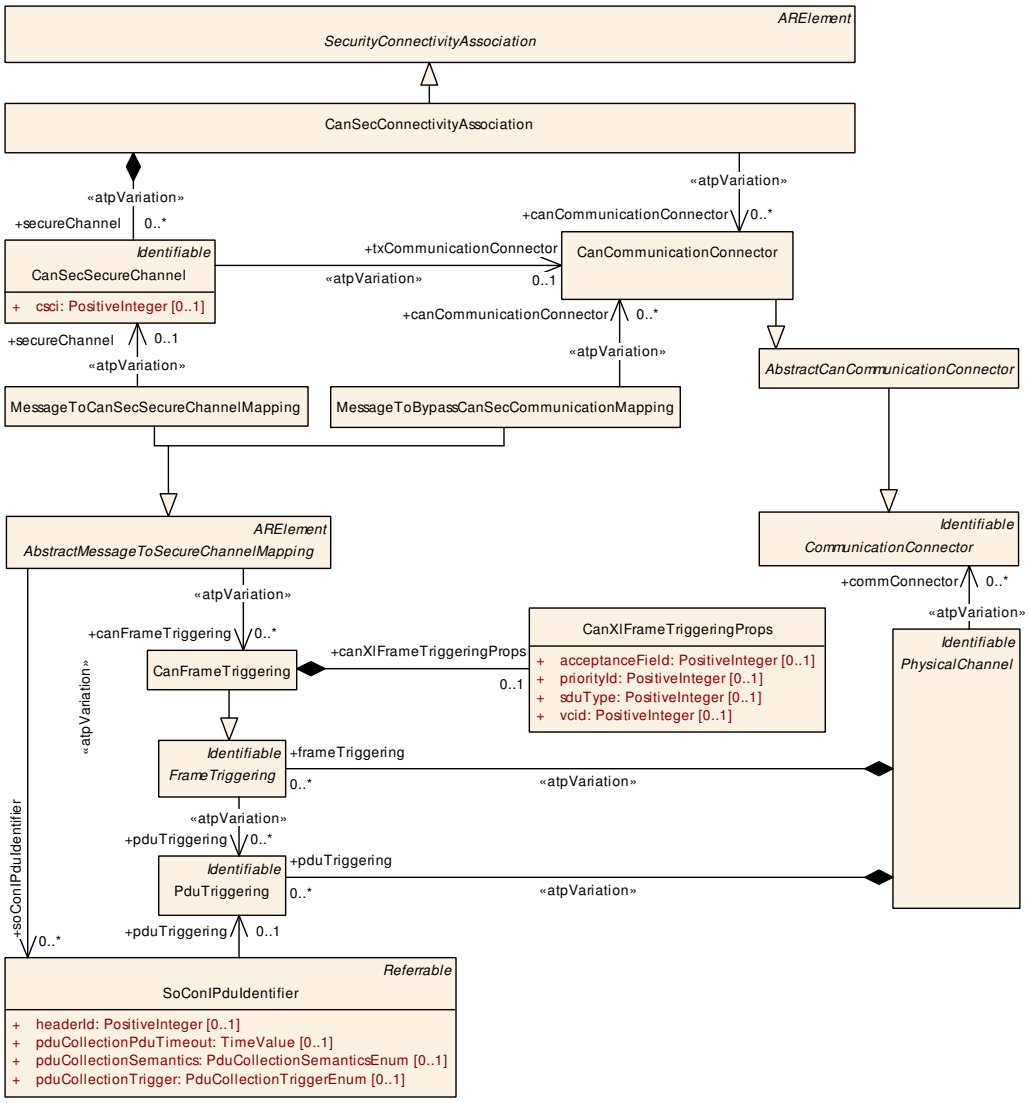

Figure 3.48: Secure Channel message mapping

Class	***AbstractMessageToSecureChannelMapping*** (abstract)			
Note	Abstract class for the definition of mapping FrameTriggerings and SoConIPduIdentifiers to a secure channel. This may also include bypass mapping, where explicitly no security shall be applied.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	MessageToBypassCanSecCommunicationMapping, MessageToCanSecSecureChannelMapping			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canFrame Triggering	CanFrameTriggering	*	ref	Reference to the mapped CanFrameTriggerings. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canFrameTriggering, canFrame Triggering.canFrameTriggering, canFrame Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

△

Class	AbstractMessageToSecureChannelMapping (abstract)			
soConIPdu Identifier	SoConIPduIdentifier	*	ref	Reference to the mapped SoConIPduIdentifiers. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=soConIPduIdentifier, soConIPduIdentifier.soConIPduIdentifier, soConIPduIdentifier.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.133: AbstractMessageToSecureChannelMapping

Class	MessageToCanSecSecureChannelMapping			
Note	Mapping of FrameTriggerings and SoConIPduIdentifiers to a CanSecSecureChannel. **Tags:** atp.recommendedPackage=SecureChannelMappings			
Base	*ARElement*, *ARObject*, *AbstractMessageToSecureChannelMapping*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
secureChannel	CanSecSecureChannel	0..1	ref	Reference to the SecureChannel to which the Frame Triggerings are mapped. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secureChannel, secureChannel.canSecSecureChannel, secureChannel.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.134: MessageToCanSecSecureChannelMapping

Class	MessageToBypassCanSecCommunicationMapping			
Note	Mapping of FrameTriggerings and SoConIPduIdentifiers to explicitly be bypassed communication. **Tags:** atp.recommendedPackage=SecureChannelMappings			
Base	*ARElement*, *ARObject*, *AbstractMessageToSecureChannelMapping*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
can Communication Connector	CanCommunication Connector	*	ref	Reference to the CanCommunicationConnectors on which the messages shall be bypassed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canCommunicationConnector, canCommunicationConnector.canCommunicationConnector, canCommunicationConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 3.135: MessageToBypassCanSecCommunicationMapping

[constr_9427] Existence of `MessageToCanSecSecureChannelMapping.secureChannel`

Imposition time: CP: `IT_SysDesc`

⌈For each `MessageToCanSecSecureChannelMapping` the reference `MessageToCanSecSecureChannelMapping.secureChannel` shall exist.⌋

[constr_9428] Consistent `MessageToCanSecSecureChannelMapping.secureChannel` and `AbstractMessageToSecureChannelMapping.canFrameTriggering` mapping

Imposition time: CP: `IT_SysDesc`

⌈Every `CanFrameTriggering` that is referenced by the `MessageToCanSecSecureChannelMapping` in the role `canFrameTriggering` shall be aggregated by a `PhysicalChannel` and that `PhysicalChannel` shall refer to the same `CanCommunicationConnector` that is referenced by the `MessageToCanSecSecureChannelMapping.secureChannel` in the role `txCommunicationConnector`.⌋

[constr_9428] is visualized in Figure 3.48, which shows the relation between the `MessageToCanSecSecureChannelMapping`, the `CanFrameTriggering`, the `PhysicalChannel`, and the `CanCommunicationConnector` on one side and the `CanSecSecureChannel` on the other side. The reason to have this circular dependency is to allow a separate definition of the `CanSecSecureChannel` (and its assigment to the `CanCommunicationConnector`) and the `MessageToCanSecSecureChannelMapping` with its references to the `CanFrameTriggering`.

The example shown in Figure 3.49 illustrates an ECU for which 3 `CanSecSecureChannel`s are defined for transmission. Consequently, that ECU is also able to receive messages from other ECUs participating in the same `CanSecConnectivityAssociation`. This example assumes that also *Node B* and *Node C* define 3 transmitting `CanSecSecureChannel`s each. This results in a total of 9 `CanSecSecureChannel`s being sent and received by each ECU participating in the `CanSecConnectivityAssociation`. The model example is shown in Figure 3.50.

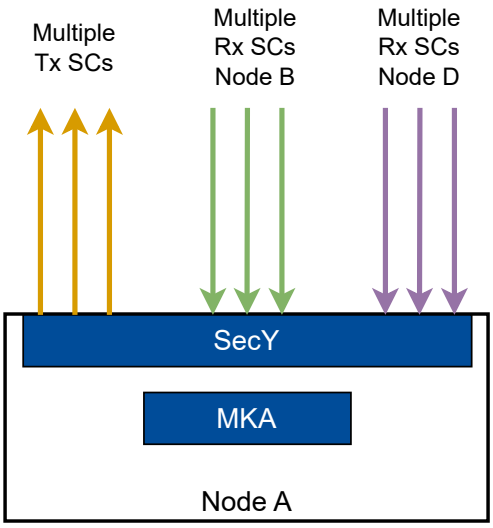

Figure 3.49: Example frame mapping for CANsec bus

The example in Figure 3.50 also shows one FrameTriggering (*FrameTriggering_Unprotected*) to be bypassed on all 3 CanCommunicationControllers.

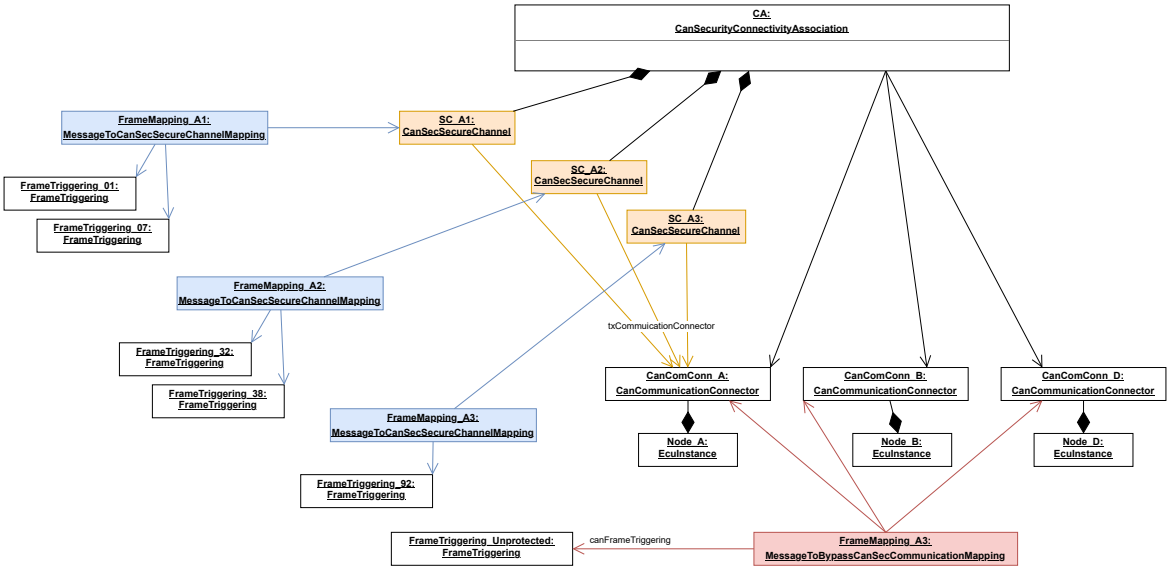

Figure 3.50: MessageToCanSecSecureChannelMapping for a CANsec bus

The implementation of the MessageToBypassCanSecCommunicationMapping and MessageToCanSecSecureChannelMapping is typically done in hardware, where first all bypass rules of one CanCommunicationController are combined into one bypass rule set in the ECU configuration. A second rule set is then applied to determine to which CanSecSecureChannel each message belongs.

Note that the definition of bypass rules and secure channel assignments in ECU configuration is based on the attributes defined in the CanXlFrameTriggeringProps of the corresponding CanFrameTriggering. A unambiguous assignment of a message to be bypassed and assigned to a dedicated secure channel is only possible if the attributes of the respective CanXlFrameTriggeringProps are diverse enough to allow a unique assignment.

3.3.8 MACsec

MACsec (Media Access Control security) is defined by IEEE standard 802.1AE and operates at the medium access control layer and defines connectionless data confidentiality and integrity for media access independent protocols. The MACsec standard specifies a set of protocols to meet the security requirements for protecting data traversing Ethernet LANs. MACsec allows unauthorized LAN connections to be identified and excluded from communication within the network. In common with IPsec and TLS, MACsec defines a security infrastructure to provide data confidentiality, data integrity and data origin authentication. With security on the MAC layer, protocols not based on IP as well as protocols using multicast can be protected by MACsec.

The MACsec standard relies on the usage of authentic partners, which are called MAC Security Entities (SecYs). The standard organizes this by MACsec Protocol Data Units (MPDU), which resides in the MAC layer (ISO/OSI layer 2). The MAC Security Key Agreement Entity (KaY) will take care of properly configuring and managing of the MAC Security Entities to enable the secure communication channel.

In the System Description the configuration of MACsec is supported on CouplingPorts that in turn are aggregated either by a CouplingElement or by an EthernetCommunicationController. This allows the configuration of MACsec on a switch port or on an EthernetCommunicationController of an EcuInstance. Please note that the same modeling approach is used in the Adaptive Platform Manifest as well.

[TPS_SYST_02386] MACsec configuration ⌈A CouplingPort that aggregates the MacSecProps in the role macSecProps defines a MAC Security Entity.⌋

[constr_5361] MACsec configuration is only allowed on CouplingPorts aggregated by switch ports and EthernetCommunicationController

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈Only if a CouplingPort is either aggregated by an EthernetCommunicationController or by a CouplingElement with couplingType set to switch, then this CouplingPort is allowed in turn to aggregate a MacSecProps in the role macSecProps.⌋

[TPS_SYST_02387] MAC Security Key Agreement Entity configuration ⌈The MacSecLocalKayProps element that is referenced by the MacSecProps in the role macSecKayConfig in combination with MacSecGlobalKayProps that is referenced by MacSecLocalKayProps in the role globalKayProps defines the configuration settings for the MACsec Key Agreement that is responsible for the peer discovery and key negotiation to secure the Ethernet link.⌋

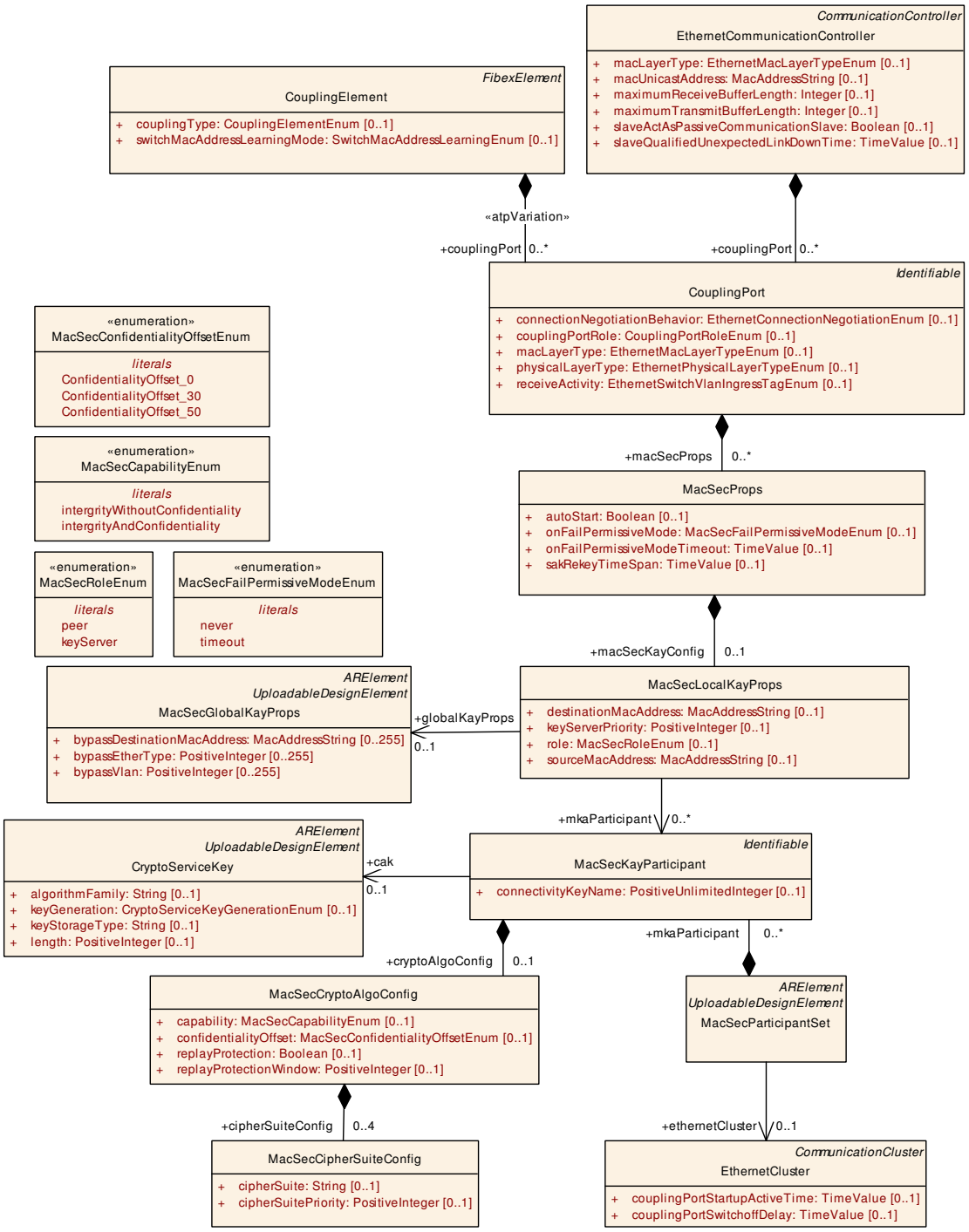

Figure 3.51: MACsec configuration

Class	**MacSecProps**
Note	This meta-class allows to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:** atp.Status=candidate
Base	*ARObject*

▽

△

Class	**MacSecProps**			
Aggregated by	CouplingPort.macSecProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
autoStart	Boolean	0..1	attr	This attribute defines how the Port Access Entity (PAE) is started: • true := Autostart • false := Manual Start **Tags:** atp.Status=candidate
macSecKay Config	MacSecLocalKayProps	0..1	aggr	Properties to configure the MKA instance (KaY) for a controlled CouplingPort (PaE). **Tags:** atp.Splitkey=macSecKayConfig atp.Status=candidate
onFail Permissive Mode	MacSecFailPermissive ModeEnum	0..1	attr	This attribute sets the behavior of the Port Access Entity in case MACsec does not succeed. **Tags:** atp.Status=candidate
onFail Permissive ModeTimeout	TimeValue	0..1	attr	Timeout in seconds to enable the controlled port in case onFailPermissiveMode is set to Timeout. **Tags:** atp.Status=candidate
sakRekeyTime Span	TimeValue	0..1	attr	Time in seconds to trigger the rekey of an in use SAK (Static Secure Association key). If set to 0, the rekey will not be triggered after a time span. **Tags:** atp.Status=candidate

Table 3.136: MacSecProps

Class	**MacSecLocalKayProps**			
Note	Configuration of the MAC Security Key Agreement Entity (KaY). **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	MacSecProps.macSecKayConfig			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
destinationMac Address	MacAddressString	0..1	attr	This attribute defines the destination MAC Address that is used to calculate the ICV (Integrity Check Value). **Tags:** atp.Status=candidate
globalKayProps	MacSecGlobalKay Props	0..1	ref	Reference to properties that are shared between MAC Security Key Agreement Entities. **Tags:** atp.Splitkey=globalKayProps atp.Status=candidate
keyServer Priority	PositiveInteger	0..1	attr	This attribute defines the key-server priority. **Tags:** atp.Status=candidate
mkaParticipant	MacSecKayParticipant	*	ref	Reference to MKA participant settings supported on the CouplingPort. **Tags:** atp.Splitkey=mkaParticipant atp.Status=candidate
role	MacSecRoleEnum	0..1	attr	Role of the MAC Security Key Agreement Entity **Tags:** atp.Status=candidate

▽

△

Class	MacSecLocalKayProps			
sourceMac Address	MacAddressString	0..1	attr	This attribute defines the source MAC Address that is used to calculate the ICV (Integrity Check Value). **Tags:** atp.Status=candidate

Table 3.137: MacSecLocalKayProps

Class	MacSecGlobalKayProps			
Note	Configuration of the MAC Security Key Agreement Entity properties that are shared by different KaY configurations. **Tags:** atp.Status=candidate atp.recommendedPackage=MacSecGlobalKayProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bypass DestinationMac Address	MacAddressString	0..255	attr	The attribute is used to define MAC destination addresses that are bypassed by MACsec. The provided MAC addresses will not be MACsec protected. Please note, IEEE 802.1X-2020 chapter 11.1.1 (Table 11-1) define specific MAC destination addresses to be bypassed.
bypassEther Type	PositiveInteger	0..255	attr	This attribute is used to define EtherTypes that are bypassed by MACsec. The providedEtherType will not be MACsec protected. **Tags:** atp.Status=candidate
bypassVlan	PositiveInteger	0..255	attr	This attribute is used to define VLAN-IDs that are bypassed by MACsec. The provided VLAN-IDs will not be MACsec protected. (VLAN-ID 0 is interpreted as no-VLAN --> Bypass untagged traffic) **Tags:** atp.Status=candidate

Table 3.138: MacSecGlobalKayProps

Class	MacSecParticipantSet			
Note	Collection of MACsec Kay Participants on an Ethernet Link. **Tags:** atp.Status=candidate atp.recommendedPackage=MacSecKayParticipantSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ethernetCluster	EthernetCluster	0..1	ref	Reference to the EthernetCluster (Link) on which the KaY participants are located **Tags:** atp.Splitkey=ethernetCluster atp.Status=candidate

▽

△

Class	**MacSecParticipantSet**			
mkaParticipant	MacSecKayParticipant	*	aggr	Configuration of a MKA Participant. **Tags:** atp.Splitkey=mkaParticipant, mkaParticipant.shortName atp.Status=candidate

Table 3.139: MacSecParticipantSet

Class	**MacSecKayParticipant**			
Note	This meta-class configures a MKA participant. **Tags:** atp.Status=candidate atp.recommendedPackage=MacSecKayParticipants			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	MacSecParticipantSet.mkaParticipant			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cak	CryptoServiceKey	0..1	ref	Reference to the Connectivity Association Key (CAK) **Tags:** atp.Splitkey=cak
connectivityKey Name	PositiveUnlimitedInteger	0..1	attr	Connectivity Key Name (CKN) defined as a hexadecimal number
cryptoAlgo Config	MacSecCryptoAlgo Config	0..1	aggr	Cryptography that is used by the MKA Participant. **Tags:** atp.Splitkey=cryptoAlgoConfig atp.Status=candidate

Table 3.140: MacSecKayParticipant

Class	**MacSecCryptoAlgoConfig**			
Note	This meta-class defines the cryptography configuration for MACsec. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	MacSecKayParticipant.cryptoAlgoConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
capability	MacSecCapabilityEnum	0..1	attr	This attribute defines the MACsec capability. **Tags:** atp.Status=candidate
cipherSuite Config	MacSecCipherSuite Config	0..4	aggr	Cipher suite configuration to use with MACsec. **Tags:** atp.Splitkey=cipherSuiteConfig atp.Status=candidate
confidentiality Offset	MacSecConfidentiality OffsetEnum	0..1	attr	The MACsec confidentiality offset specifies the number of bytes starting from the frame header. MACsec encrypts only the bytes after the offset in a frame. **Tags:** atp.Status=candidate
replayProtection	Boolean	0..1	attr	This attribute is used to configure the MACsec replay protection. **Tags:** atp.Status=candidate

▽

△

Class	**MacSecCryptoAlgoConfig**			
replayProtection Window	PositiveInteger	0..1	attr	In case replay protection is active, this attribute defines the replay protection window. **Tags:** atp.Status=candidate

Table 3.141: MacSecCryptoAlgoConfig

Class	**MacSecCipherSuiteConfig**			
Note	This meta-class defines the cipher suite configuration to use with MACsec. cipherSuitePriority is present in case the MKA instance acts as a Key Server to select the cipher suite to use for MACsec. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	MacSecCryptoAlgoConfig.cipherSuiteConfig			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
cipherSuite	String	0..1	attr	Cipher Suite to use for MACsec. **Tags:** atp.Status=candidate
cipherSuite Priority	PositiveInteger	0..1	attr	In case the MKA instance acts as a Key Server, the priority is used to select the Cipher Suite to use with MACsec from the supported Ciphers. **Tags:** atp.Status=candidate

Table 3.142: MacSecCipherSuiteConfig

[TPS_SYST_02388] Standardized values for the attribute `cipherSuite` of meta-class `MacSecCipherSuiteConfig` ⌈The following values for `MacSecCipherSuiteConfig.cipherSuite` are reserved by the AUTOSAR standard:

- GCM-AES-128
- GCM-AES-256
- GCM-AES-XPN-128
- GCM-AES-XPN-256

⌋

[TPS_SYST_02389] Semantics of `MacSecCipherSuiteConfig.cipherSuitePriority` ⌈The `MacSecCryptoAlgoConfig` can define up to four `MacSecCipherSuiteConfig`s. If more then one `MacSecCipherSuiteConfig` is defined then the `cipherSuitePriority` decides about the chosen cipher. The `cipherSuitePriority` of value 1 means the highest priority and 4 means the lowest priority.⌋

Enumeration	**MacSecConfidentialityOffsetEnum**
Note	This enum defines the MACsec capability options. **Tags:** atp.Status=candidate
Aggregated by	MacSecCryptoAlgoConfig.confidentialityOffset
Literal	*Description*

▽

△

Enumeration	**MacSecConfidentialityOffsetEnum**
Confidentiality Offset_0	confidentiality offset of 0. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=CONFIDENTIALITY-OFFSET--0
Confidentiality Offset_30	confidentiality offset of 30. **Tags:** atp.EnumerationLiteralIndex=1 xml.name=CONFIDENTIALITY-OFFSET--30
Confidentiality Offset_50	confidentiality offset of 50. **Tags:** atp.EnumerationLiteralIndex=2 xml.name=CONFIDENTIALITY-OFFSET--50

Table 3.143: MacSecConfidentialityOffsetEnum

Enumeration	**MacSecCapabilityEnum**
Note	This enum defines the MACsec capability options. **Tags:** atp.Status=candidate
Aggregated by	MacSecCryptoAlgoConfig.capability
Literal	***Description***
intergrityAnd Confidentiality	Option that ensures confidentiality and integrity **Tags:** atp.EnumerationLiteralIndex=1
intergrityWithout Confidentiality	Option that ensures integrity without confidentiality **Tags:** atp.EnumerationLiteralIndex=0

Table 3.144: MacSecCapabilityEnum

Enumeration	**MacSecRoleEnum**
Note	This enum defines the MACsec Role options. **Tags:** atp.Status=candidate
Aggregated by	MacSecLocalKayProps.role
Literal	***Description***
keyServer	Port acts in the KeyServer role **Tags:** atp.EnumerationLiteralIndex=1
peer	Port acts in the peer role **Tags:** atp.EnumerationLiteralIndex=0

Table 3.145: MacSecRoleEnum

Enumeration	**MacSecFailPermissiveModeEnum**
Note	Behavior options of the Port Access Entity in case MACsec does not succeed. **Tags:** atp.Status=candidate
Aggregated by	MacSecProps.onFailPermissiveMode
Literal	***Description***

▽

△

Enumeration	**MacSecFailPermissiveModeEnum**
never	The controlled port will never be set to enabled if the participants cannot establish and successfully use a MACsec Secure Channel. **Tags:** atp.EnumerationLiteralIndex=0
timeout	The controlled port will be set to enabled and MACsec will not be used in the port if the timeout value (onFailPermissiveModeTimeout) is reached and the following conditions apply: - A participant belonging to the same CA was recognized and authenticated. - A secure channel could be established. - Both participants can transmit and receive MACsec protected traffic through the SC. **Tags:** atp.EnumerationLiteralIndex=1

Table 3.146: MacSecFailPermissiveModeEnum

For MACsec to start up, a key distribution based according to the MAC Security Key Agreement (MKA) protocol runs first. The Connectivity Association Key (CAK) used for MKA is referenced as `MacSecKayParticipant.cak`. The Connectivity Key Name (CKN), which identifies the CAK, is described by the parameter `MacSecKayParticipant.connectivityKeyName`. The CAK and the CKN must match on both ends of a link to create a MACsec-secured link.

While one might choose only to communicate encrypted, the key distribution itself needs to be unencrypted but integrity protected. In addition, user may choose to allow additional Ethernet frames unprotected, e.g., for installing keys into ECUs. These frames can be selected via so-called bypass rules based on e.g., EtherType or VLAN-ID defined in `MacSecGlobalKayProps`.

Each `MacSecKayParticipant` that is referenced by `MacSecLocalKayProps` in the role `mkaParticipant` defines the Connectivity Association Key and the `MacSecCryptoAlgoConfig`. The `MacSecKayParticipant` settings in the connectivity association need to match on both `CouplingPort`s of the link to enable MACsec.

For better control and predictability, the MAC Security Key Agreement participant that will get the KeyServer role (the node generating the MACsec Key for this link) can be configured by `role`.

3.3.9 CDD

The System Template allows the integration of custom bus systems on the topology level.

[TPS_SYST_01127] CDD Topology support ⌈The elements `UserDefinedCluster`, `UserDefinedPhysicalChannel`, `UserDefinedCommunicationConnector` and `UserDefinedCommunicationController` can be used to describe alternative communication technologies (e.g. I2C, USB, serial line) that are integrated in AUTOSAR as Complex Drivers.⌋

The Pdu-based communication via Complex Drivers is described in chapter 6.14.

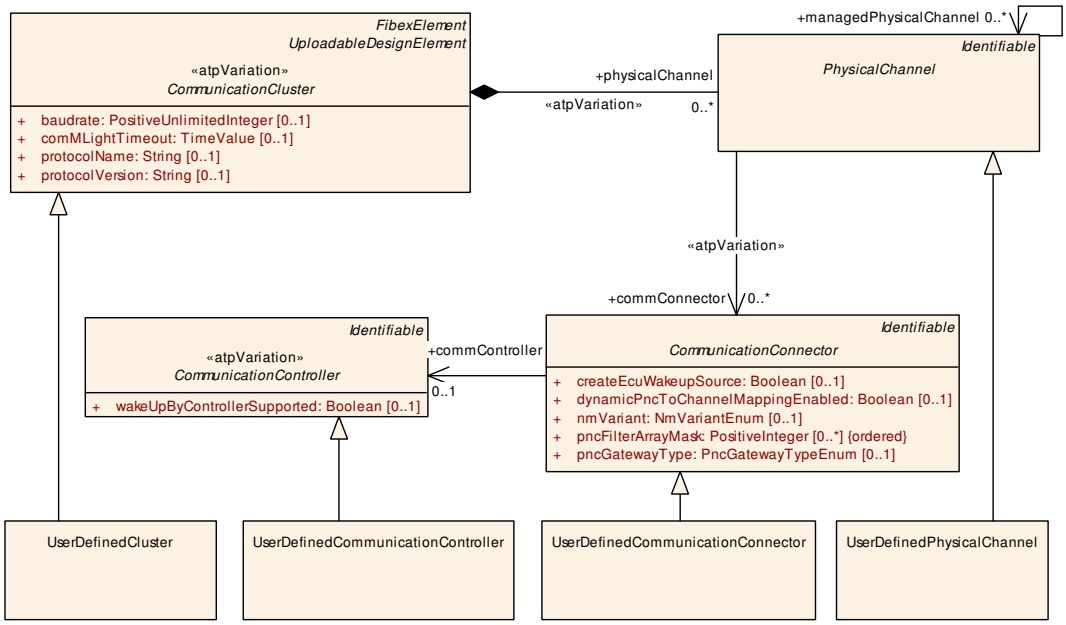

Figure 3.52: User defined topology elements

Class	«atpVariation» **UserDefinedCluster**			
Note	This element allows the modeling of arbitrary Communication Clusters (e.g. bus systems that are not supported by AUTOSAR). **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	*ARElement*, *ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.147: UserDefinedCluster

Class	**UserDefinedPhysicalChannel**			
Note	This element allows the modeling of arbitrary Physical Channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.148: UserDefinedPhysicalChannel

Class	**UserDefinedCommunicationConnector**			
Note	This element allows the modeling of arbitrary Communication Connectors.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 3.149: UserDefinedCommunicationConnector

Class	«atpVariation» **UserDefinedCommunicationController**			
Note	This element allows the modeling of arbitrary Communication Controllers.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 3.150: UserDefinedCommunicationController

3.4 Mapping of Topology Entities onto Hardware Elements

As explained in the previous sections, the System Template contains all classes necessary to describe the physical topology in an AUTOSAR system. Based on this description, the communication matrix can be realized as explained in chapter 6.

[TPS_SYST_01019] Mapping of topology elements to elements of the ECU Resource Template ⌈It is possible to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template.⌋

It can be specified which `HwElement` is realizing each given `EcuInstance`, providing the means for algorithms to map software components onto the systems `EcuInstance`. By specifying which `hwCommunicationPort`[3] on a `hwCommunicationController`[4] implements the topology's `CommunicationConnector` on a `CommunicationController`, the hardware-oriented parameters in the Communication-drivers may be derived in ECU configuration phase.

Please note that this is a rather specific type of mapping, optionally binding ECU-local topology elements to specific hardware resources. It should not be confused with the System Mapping part of the System Description, where system-wide mapping decisions are described, like e.g. the mapping of Software Components onto ECUs or the mapping of Data Element Prototypes onto System Signals (for the System Mapping, see chapter 5).

[3] `HwPinGroup` which is of category Communication Port
[4] `HwElement` which is of category Communication Controller

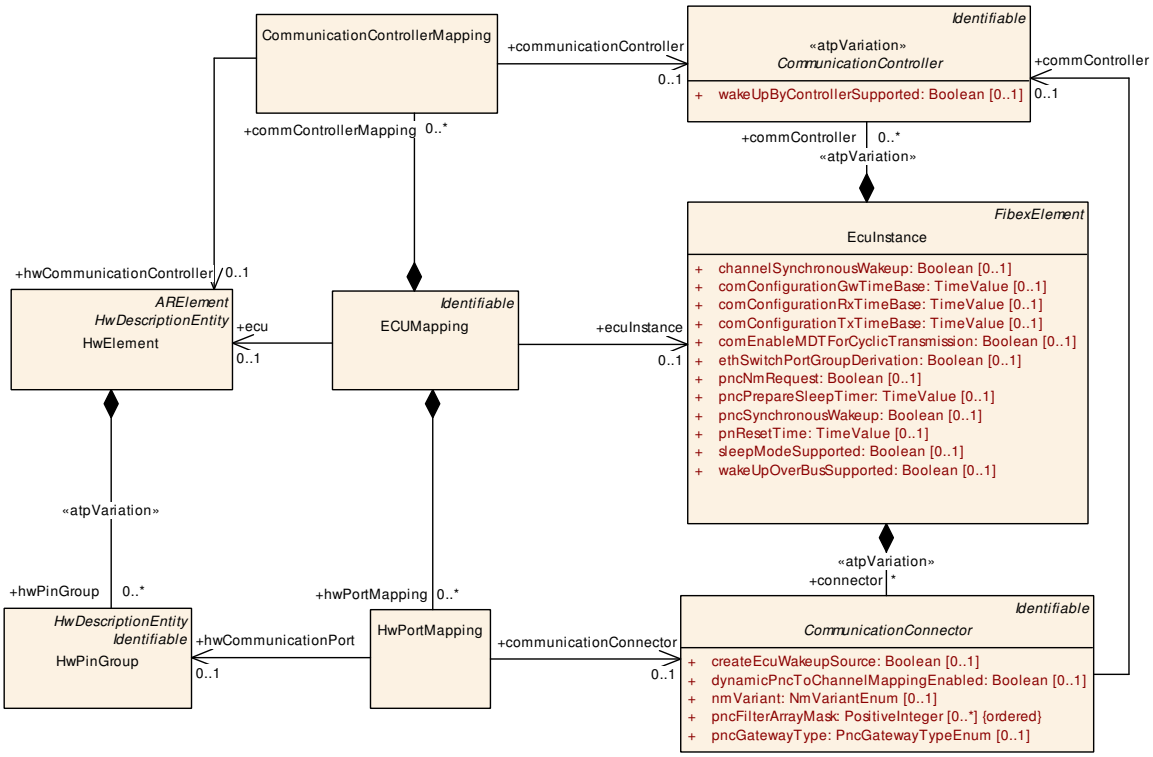

Figure 3.53: Mapping of topology description elements in the System Template onto hardware elements defined in the ECU Resource Template (ECUResourceMapping)

[constr_3006] valid EcuMapping

Imposition time: CP: IT_SysDesc

⌈The referenced hwCommunicationController and hwCommunicationPort shall be part of the referenced ecu.

ECUMapping.ecu.nestedElement contains ECUMapping.commController-Mapping.hwCommunicationController

ECUMapping.ecu.nestedElement contains ECUMapping.hwPortMapping.hwCommunicationPort⌋

3.4.1 ECU Mapping

ECUMapping allows to assign a HwElement to an EcuInstance used in a physical topology.

[TPS_SYST_01013] EcuInstance stands for its own ⌈An EcuInstance can be defined in a stand alone and reusable way without a need to have an ECUMapping.⌋

[constr_3030] valid relationship between ECUMapping and EcuInstance

Imposition time: CP: IT_SysDesc

⌈If an EcuInstance is assigned to a HwElement the EcuInstance shall belong to the same System as the ECUMapping.⌋

[constr_3248] Category of HwElement for ECUMapping

Imposition time: CP: IT_SysDesc

⌈The HwElement which is referenced from ECUMapping in the role ecu shall be of category MicroController⌋

There exists an inconsistency between the System Template and the ECU Resource Template concerning the usage of the term "Ecu". In the System Template "Ecu" is used to determine one instance of an AUTOSAR Stack (e.g. like in EcuInstance). In the Ecu Resource Template "Ecu" is used to describe the physical box (HwElement of category Ecu) containing the electronics which may contain several processing units with several AUTOSAR Stack instances running.

Class	**ECUMapping**			
Note	ECUMapping allows to assign an ECU hardware type (defined in the ECU Resource Template) to an ECUInstance used in a physical topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.ecuResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
commControllerMapping	CommunicationControllerMapping	*	aggr	The ECUMapping contains the mapping of all CommunicationControllers of the ECU. **Tags:** atp.Splitkey=commControllerMapping
ecu	HwElement	0..1	ref	Reference to a HwElement of category ECU in the ECU Resource Template. **Tags:** atp.Splitkey=ecu
ecuInstance	EcuInstance	0..1	ref	Reference to the EcuInstance in the System Template **Tags:** atp.Splitkey=ecuInstance
hwPortMapping	HwPortMapping	*	aggr	The ECUMapping contains the mapping of all HW Communication Ports of the ECU. **Tags:** atp.Splitkey=hwPortMapping

Table 3.151: ECUMapping

[constr_5399] Existence of ecu

Imposition time: CP: IT_SysDesc

⌈For each ECUMapping, the reference to HwElement in the role ecu shall exist.⌋

[constr_5400] Existence of ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each ECUMapping, the reference to EcuInstance in the role ecuInstance shall exist.⌋

3.4.2 Communication Controller Mapping

[TPS_SYST_01014] Semantics of `CommunicationControllerMapping` ⌈`CommunicationControllerMapping` specifies the `HwElement` to realize the specified `CommunicationController` in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers.⌋

Class	**CommunicationControllerMapping**			
Note	CommunicationControllerMapping specifies the CommunicationPeripheral hardware (defined in the ECU Resource Template) to realize the specified CommunicationController in a physical topology.			
Base	*ARObject*			
Aggregated by	ECUMapping.commControllerMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Controller	Communication Controller	0..1	ref	Reference to the CommunicationController in the System Template **Tags:** atp.Splitkey=communicationController
hw Communication Controller	HwElement	0..1	ref	Reference to a HwElement of category Communication Controller in the ECU Resource Template. **Tags:** atp.Splitkey=hwCommunicationController

Table 3.152: CommunicationControllerMapping

[constr_5401] Existence of `communicationController`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommunicationControllerMapping`, the reference to `CommunicationController` in the role `communicationController` shall exist.⌋

[constr_5402] Existence of `hwCommunicationController`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommunicationControllerMapping`, the reference to `HwElement` in the role `hwCommunicationController` shall exist.⌋

3.4.3 HW-Port Mapping

[TPS_SYST_01015] Semantics of `HwPortMapping` ⌈`HwPortMapping` specifies the hardware to realize the specified `CommunicationConnector` in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers.⌋

Class	**HwPortMapping**			
Note	HWPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.			
Base	*ARObject*			
Aggregated by	ECUMapping.hwPortMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	Communication Connector	0..1	ref	Reference to the CommunicationConnector in the System Template **Tags:** atp.Splitkey=communicationConnector
hw Communication Port	HwPinGroup	0..1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the Hw CommunicationController is described in the Ecu Resource Description. **Tags:** atp.Splitkey=hwCommunicationPort

Table 3.153: HwPortMapping

[constr_5403] Existence of `communicationConnector`

Imposition time: CP: `IT_SysDesc`

⌈For each `HwPortMapping`, the reference to `CommunicationConnector` in the role `communicationConnector` shall exist.⌋

[constr_5404] Existence of `hwCommunicationPort`

Imposition time: CP: `IT_SysDesc`

⌈For each `HwPortMapping`, the reference to `HwPinGroup` in the role `hwCommunicationPort` shall exist.⌋

4 Top-level Software Composition

One of the most important inputs for the System Generator is the knowledge about the Application Software Components, their communication capabilities and the connections between them: Each SystemSignal (chapter 6.2) that is going to be exchanged between mapped Software Components onto different ECUs is a consequence of a connection between such application Software Components.

In AUTOSAR, Software Components can either be atomic (AtomicSwComponentType) or may consist of a composition of other Software Components CompositionSwComponentType [4]. In order to assemble non-trivial applications from AUTOSAR components, such compositions can be built up hierarchically, until the outermost CompositionSwComponentType forms a kind of top-level composition.

[constr_3031] Complete System Description does not have ports on the outermost composition

Imposition time: CP: IT_SysDesc

⌈In a complete System with category ABSTRACT_SYSTEM_DESCRIPTION or System with category SYSTEM_DESCRIPTION this outermost CompositionSwComponentType has the unique feature that it doesn't have any outside ports, but all the SWC contained in it are connected to each other and fully specified by their SwComponentTypes, PortPrototypes, PortInterfaces, VariableDataPrototypes, InternalBehavior etc.⌋

[TPS_SYST_01016] System Extract, Ecu System Description and Ecu Extract may have ports ⌈In a System with category SYSTEM_EXTRACT and a System with category ECU_SYSTEM_DESCRIPTION and a System with category ECU_EXTRACT outside ports for the outermost composition are allowed.⌋

[TPS_SYST_02312] Ports for outermost composition of a SW_CLUSTER_SYSTEM_DESCRIPTION ⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION outside ports for the outermost composition are allowed.⌋

Since the System/Ecu Extract represents the view on one Ecu, there may be the need to define the communication of this extract with the outside world.

Two approaches are available how the external communication of an ECU in the System Extract is described. In section 13.2 the communication mapping is performed in the hierarchical structure of software components. In section 13.3 external communication delegation ports are added to the System extract outermost composition. Each delegated port is connected via a DelegationSwConnector with ports of the included components that are used for the external communication.

A System considers such a top-level CompositionSwComponentType as its application software system input by owning exactly one RootSwComposition-Prototype class, which points to the CompositionSwComponentType forming the input via its ≪isOfType≫ relationship as shown in Figure 2.1.

[TPS_SYST_01017] The role of the top-level software composition ⌈An AUTOSAR System uses the specialized prototype class RootSwCompositionPrototype in order to designate the referenced CompositionSwComponentType as the top-level software composition.⌋

Class	**RootSwCompositionPrototype**			
Note	The RootSwCompositionPrototype represents the top-level-composition of software components within a given System. According to the use case of the System, this may for example be a more or less complete VFB description, the software of a System Extract or the software of a flat ECU Extract with only atomic SWCs. Therefore the RootSwComposition will only occasionally contain all atomic software components that are used in a complete VFB System. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System. The internal structure of such a component contains often substantial intellectual property of a supplier. Therefore a top-level software composition will often contain empty compositions which represent subsystems. The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, System.rootSoftwareComposition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calibration ParameterValue Set	CalibrationParameter ValueSet	*	ref	Used CalibrationParameterValueSet for instance specific initialization of calibration parameters. **Tags:** atp.Splitkey=calibrationParameterValueSet This Attribute is only used by the AUTOSAR Classic Platform.
flatMap	FlatMap	0..1	ref	The FlatMap used in the scope of this RootSw CompositionPrototype. **Tags:** atp.Splitkey=flatMap This Attribute is only used by the AUTOSAR Classic Platform.
software Composition	CompositionSw ComponentType	0..1	tref	We assume that there is exactly one top-level composition that includes all Component instances of the system. **Stereotypes:** isOfType **Tags:** atp.Splitkey=softwareComposition

Table 4.1: RootSwCompositionPrototype

The existence of the reference in the role RootSwCompositionPrototype.softwareComposition is governed by [constr_2633].

5 Mapping

A central part of the system generation process is the mapping of software components (SwComponentPrototypes) to ECUs, and the subsequent mapping of the communication between these software components to bus frames. Input to the software component mapping is the RootSwCompositionPrototype, which describes which software components have to be mapped, and the System Topology, which defines the ECU instances that are available as mapping targets. Once this mapping is done, also the communication matrix has to be taken into account for the next mapping step, the mapping of data elements exchanged between software components to bus frames. This communication matrix may either be predefined, or may be generated as part of this second mapping step. In the metamodel, different aspects of these mapping are aggregated by the meta class SystemMapping, as shown in Figure 5.1.

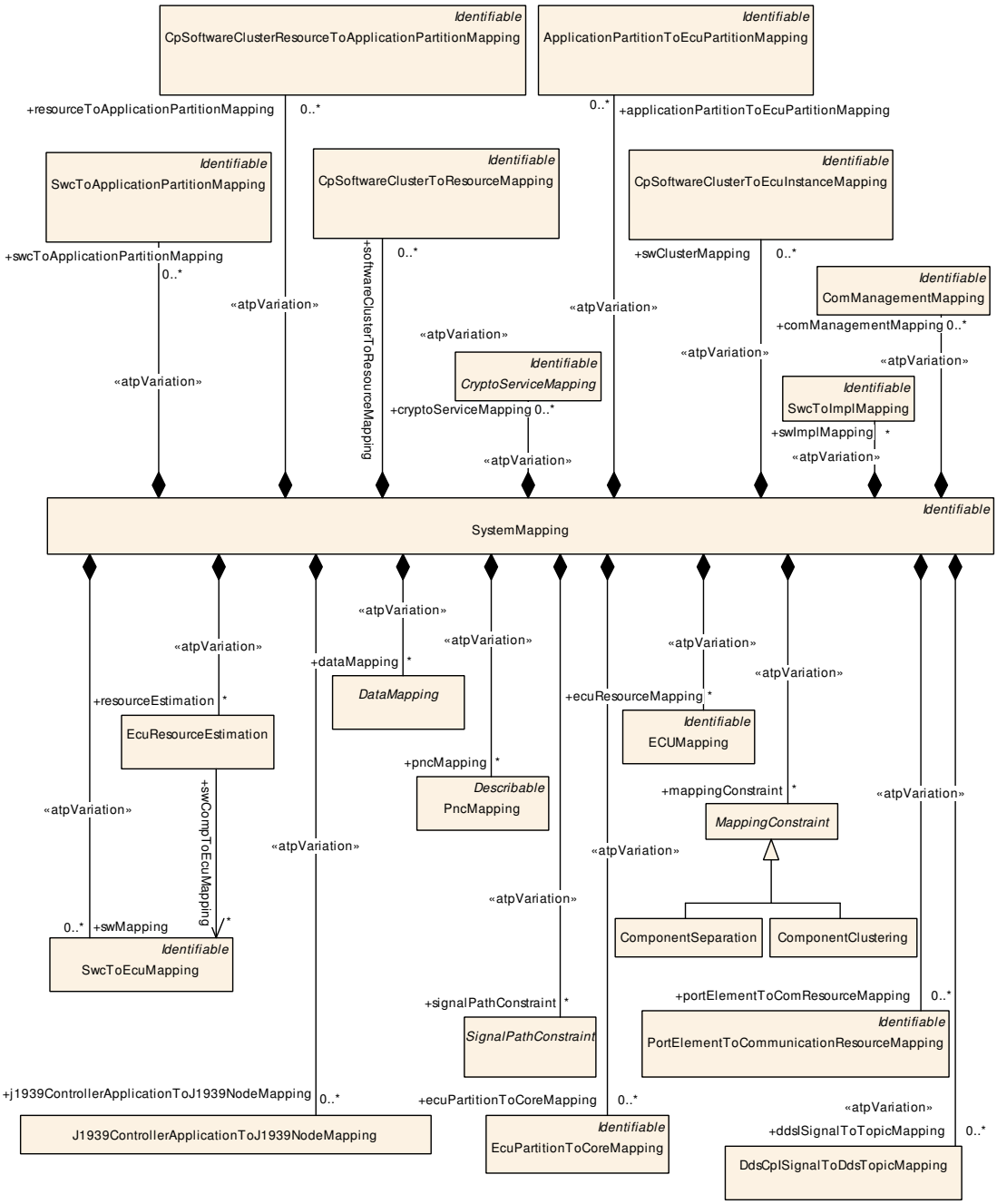

Figure 5.1: Mapping Overview (Mapping)

The following mappings are defined:

- The SwcToEcuMapping meta-class maps one or several SwComponentPrototypes to ECUs. In the System Constraint Description it is possible to predefine the mapping of SwComponentPrototypes to ECUs. The predefinition limits the system architect's freedom to map software components to arbitrary ECUs. After the system generation in the System Configuration Description, all atomic software components that are directly or indirectly part of the top level composition shall be mapped with this mapping rule. Software component mapping is described in detail in chapter 5.1.
- The meta-class EcuResourceEstimation specifies the resource estimation for RTE and basic software (see chapter 5.4).
- The ECUMapping meta-class is used to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template (see chapter 3.4).
- The DataMapping meta-class is used to map VariableDataPrototypes and ClientServerOperations in software component ports (i.e. the data exchanges between software components) to signals. The data mapping is described in detail in chapter 5.2.
- The ComManagementMapping defines the mapping of one or several Mode Management PortGroups and communication channels (see chapter 5.5.2).
- The PncMapping defines the Partial Network behavior (see chapter 5.5.1).
- The SignalPathConstraint meta-class is used to define which specific way a signal (data element or client server operation arguments) between two Software Components should take in the network without defining in which frame and with which timing it is transmitted. This Signal Path Constraint is introduced in chapter 5.2.2.
- The MappingConstraint meta-class is used to define constraints that constrain the mapping of software components. It's sub-classes allow to constraint which SwComponentPrototypes shall be mapped together on the same ECU (ComponentClustering) and which shall not be mapped to the same ECU (ComponentSeparation). The mapping constraints are described in detail in chapter 5.1.4.
- The J1939ControllerApplicationToJ1939NodeMapping maps a Software Component to which a standardized function id is assigned to a J1939Node (see chapter 3.3.2.1)
- The CpSoftwareClusterToEcuInstanceMapping meta-class maps a CpSoftwareCluster to an EcuInstance (see chapter 5.6)
- The CpSoftwareClusterToResourceMapping meta-class maps a CpSoftwareClusterServiceResource to CpSoftwareClusters (see chapter 11.2)

- The CpSoftwareClusterResourceToApplicationPartitionMapping meta-class maps a Software Cluster resource to an Application Partition (see chapter 5.6)
- The SwcToImplMapping meta-class is used to assign one Implementation to one or more SwComponentPrototypes (see chapter 5.1.2).
- The Class DdsCpISignalToDdsTopicMapping is used to assign one DdsCpTopic to one ISignals). To map one ISignal to more than one DdsCpTopic (Fan-Out), multiple DdsCpISignalToDdsTopicMapping are used. For any details, see chapter 5.7.
- The Class EcuPartitionToCoreMapping is used maps an EcuPartition to one *Core*. For details, see chapter 5.8.

Class	**SystemMapping**			
Note	The system mapping aggregates all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	System.mapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application PartitionToEcu Partition Mapping	ApplicationPartitionTo EcuPartitionMapping	*	aggr	Mapping of ApplicationPartitions to EcuPartitions **Stereotypes:** atpVariation **Tags:** atp.Splitkey=applicationPartitionToEcuPartitionMapping, applicationPartitionToEcuPartitionMapping.shortName, applicationPartitionToEcuPartitionMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
appOsTask ProxyToEcu TaskProxy Mapping	AppOsTaskProxyToEcu TaskProxyMapping	*	aggr	Mapping of an OsTaskProxy that was created in the context of a SwComponent to an OsTaskProxy that was created in the context of an Ecu. **Tags:** atp.Splitkey=appOsTaskProxyToEcuTaskProxy Mapping, appOsTaskProxyToEcuTaskProxy Mapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
com Management Mapping	ComManagement Mapping	*	aggr	Mappings between Mode Management PortGroups and communication channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=comManagementMapping, comManagement Mapping.shortName, comManagementMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	SystemMapping			
cryptoService Mapping	CryptoServiceMapping	*	aggr	This aggregation represents the collection of crypto service mappings in the context of the enclosing System Mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=cryptoServiceMapping, cryptoService Mapping.shortName, cryptoServiceMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
cyclicHandling ComDataToOs TaskProxy Mapping	CyclicHandlingCom DataToOsTaskProxy Mapping	*	aggr	Mapping of VariableDataPrototypes to an OsTaskProxy for the Cyclic Handling of Communication Data in the RTE. **Tags:** atp.Splitkey=cyclicHandlingComDataToOsTask ProxyMapping, cyclicHandlingComDataToOsTaskProxy Mapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
dataMapping	DataMapping	*	aggr	The data mappings defined. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataMapping, dataMapping.variation Point.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
ddsISignalTo TopicMapping	DdsCpISignalToDds TopicMapping	*	aggr	Collection of DdsISignalToDdsTopicMappings. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ddsISignalToTopicMapping, ddsISignalTo TopicMapping.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
ecuPartitionTo CoreMapping	EcuPartitionToCore Mapping	*	aggr	Mapping of EcuPartions to a Cores. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuPartitionToCoreMapping, ecuPartitionTo CoreMapping.shortName, ecuPartitionToCore Mapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	SystemMapping			
ecuResource Mapping	ECUMapping	*	aggr	Mapping of hardware related topology elements onto their counterpart definitions in the ECU Resource Template. atpVariation: The ECU Resource type might be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuResourceMapping, ecuResource Mapping.shortName, ecuResourceMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
j1939Controller ApplicationTo J1939Node Mapping	J1939Controller ApplicationToJ1939 NodeMapping	*	aggr	Mapping of a J1939ControllerApplication to a J1939Node. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=j1939ControllerApplicationToJ1939Node Mapping, j1939ControllerApplicationToJ1939Node Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
mapping Constraint	MappingConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of SW components to ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mappingConstraint, mapping Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
pncMapping	PncMapping	*	aggr	Mappings between Virtual Function Clusters and Partial Network Clusters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pncMapping, pncMapping.ident.shortName, pncMapping.pncIdentifier, pncMapping.shortLabel, pnc Mapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
portElementTo ComResource Mapping	PortElementTo Communication ResourceMapping	*	aggr	maps a communication resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portElementToComResourceMapping, port ElementToComResourceMapping.shortName, port ElementToComResourceMapping.variationPoint.short Label vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	SystemMapping			
resource Estimation	EcuResourceEstimation	*	aggr	Resource estimations for this set of mappings, zero or one per ECU instance. atpVariation: Used ECUs are variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=resourceEstimation, resource Estimation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
resourceTo Application Partition Mapping	CpSoftwareCluster ResourceToApplication PartitionMapping	*	aggr	Maps a Software Cluster resource to an Application Partition to restrict the usage. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=resourceToApplicationPartitionMapping, resourceToApplicationPartitionMapping.shortName, resourceToApplicationPartitionMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
rteEvent Separation	RteEventInSystem Separation	*	aggr	Separation constraint that limits the mapping freedom for the mapping of RteEvents to OsTasks in the System context. **Tags:** atp.Splitkey=rteEventSeparation, rteEvent Separation.shortName This Attribute is only used by the AUTOSAR Classic Platform.
rteEventToOs TaskProxy Mapping	RteEventInSystemToOs TaskProxyMapping	*	aggr	Constraint that enforces a mapping of RteEvent to a particular OsTask in the System context. **Tags:** atp.Splitkey=rteEventToOsTaskProxyMapping, rte EventToOsTaskProxyMapping.shortName This Attribute is only used by the AUTOSAR Classic Platform.
signalPath Constraint	SignalPathConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of data elements to signals. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=signalPathConstraint, signalPath Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
softwareCluster ToApplication Partition Mapping	CpSoftwareClusterTo ApplicationPartition Mapping	*	aggr	The mapping of ApplicationPartitions to a CpSoftware Cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=softwareClusterToApplicationPartition Mapping, softwareClusterToApplicationPartition Mapping.shortName, softwareClusterToApplication PartitionMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	SystemMapping			
softwareCluster ToResource Mapping	CpSoftwareClusterTo ResourceMapping	*	aggr	maps a service resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=softwareClusterToResourceMapping, softwareClusterToResourceMapping.shortName, softwareClusterToResourceMapping.variationPoint.short Label vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
swCluster Mapping	CpSoftwareClusterTo EcuInstanceMapping	*	aggr	The mappings of SW cluster to ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swClusterMapping, swClusterMapping.short Name, swClusterMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
swcTo Application Partition Mapping	SwcToApplication PartitionMapping	*	aggr	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swcToApplicationPartitionMapping, swcTo ApplicationPartitionMapping.shortName, swcTo ApplicationPartitionMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
swImplMapping	SwcToImplMapping	*	aggr	The mappings of AtomicSoftwareComponent Instances to Implementations. atpVariation: Derived, because SwcToEcuMapping is variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swImplMapping, swImplMapping.shortName, swImplMapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
swMapping	SwcToEcuMapping	*	aggr	The mappings of SW components to ECUs. atpVariation: SWC shall be mapped to other ECUs. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swMapping, swMapping.shortName, sw Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**SystemMapping**			
systemSignal GroupToCom Resource Mapping	SystemSignalGroupTo Communication ResourceMapping	*	aggr	Mapping of a communication resource to a SystemSignal Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemSignalGroupToComResource Mapping, systemSignalGroupToComResource Mapping.shortName, systemSignalGroupToCom ResourceMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
systemSignalTo ComResource Mapping	SystemSignalTo Communication ResourceMapping	*	aggr	Mapping of a communication resource to a SystemSignal. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=systemSignalToComResourceMapping, systemSignalToComResourceMapping.shortName, systemSignalToComResourceMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.1: SystemMapping

5.1 Software Component Mapping

A fundamental concept of AUTOSAR is that SW components may be developed independently of a specific ECU hardware, and can be mapped to an ECU in the AUTOSAR System Generation Process. The System Constraint Description acts as an input to this System Generation Phase. Nevertheless, there may be some SW components which are already mapped due to previous iterations of the system generation step, and there may be system constraints that limit the system architect's freedom to map SW components to arbitrary ECUs. In the following, the individual elements are described in more detail.

Please note that the purpose of [constr_5116] is to support the unambiguous mapping of symbols in memory sections. In the AUTOSAR Basic Software, the namespace of a Bsw Module is allocated case insensitive in order to support the definition of symbols in solely upper case notation.

[constr_5116] Uniqueness of the symbols of software-components and BSW modules

Imposition time: CP: `IT_EcuExt`

⌈For all `SwComponentPrototype`s typed by an `ApplicationSwComponentType`, `NvBlockSwComponentType` or `SensorActuatorSwComponentType` mapped to a given `EcuInstance` by means of `SwcToEcuMapping` respectively `SwcToApplicationPartitionMapping` and `ApplicationPartitionToEcuPartitionMapping` the following restriction applies:

The symbolic name of an AtomicSwComponentType referenced by a respective SwComponentPrototype in the role type shall not overlap with the module implementation prefix (MIP) of any of the basic software-modules existing on the EcuInstance.

The symbolic name of an AtomicSwComponentType is derived from the value of

- AtomicSwComponentType.symbol, or if this attribute does not exist
- AtomicSwComponentType.shortName.

⌋

More information about the nature and usage of the symbolic name of a software-component can be found in [TPS_SWCT_01110], [TPS_SWCT_01000], and [TPS_SWCT_01635].

The restriction in [constr_5116] does not apply for ServiceSwComponentTypes, ComplexDeviceDriverSwComponentTypes, and EcuAbstractionSwComponentTypes. The reason is that for the Basic Software that utilizes standardized Interfaces and AUTOSAR Interfaces needs a SwComponentType and a Basic Software Description to describe both kinds of Interfaces but only one header file for memory allocation is expected. So for example the definition of a “Dem” Service SWC and a “Dem” Basic Module Description defines a legal naming in AUTOSAR.

5.1.1 SW Component to ECU Mapping

[TPS_SYST_01001] Definition of SwcToEcuMapping ⌈With the SwcToEcuMapping element it is possible to express the mapping of SwComponentPrototypes to one EcuInstance.⌋

[constr_5486] Existence of SwcToEcuMapping.component

Imposition time: CP: IT_SysDesc

⌈For each SwcToEcuMapping, the reference to SwComponentPrototype in the role component shall exist.⌋

[constr_5487] Existence of SwcToEcuMapping.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each SwcToEcuMapping the reference to EcuInstance in the role ecuInstance shall exist.⌋

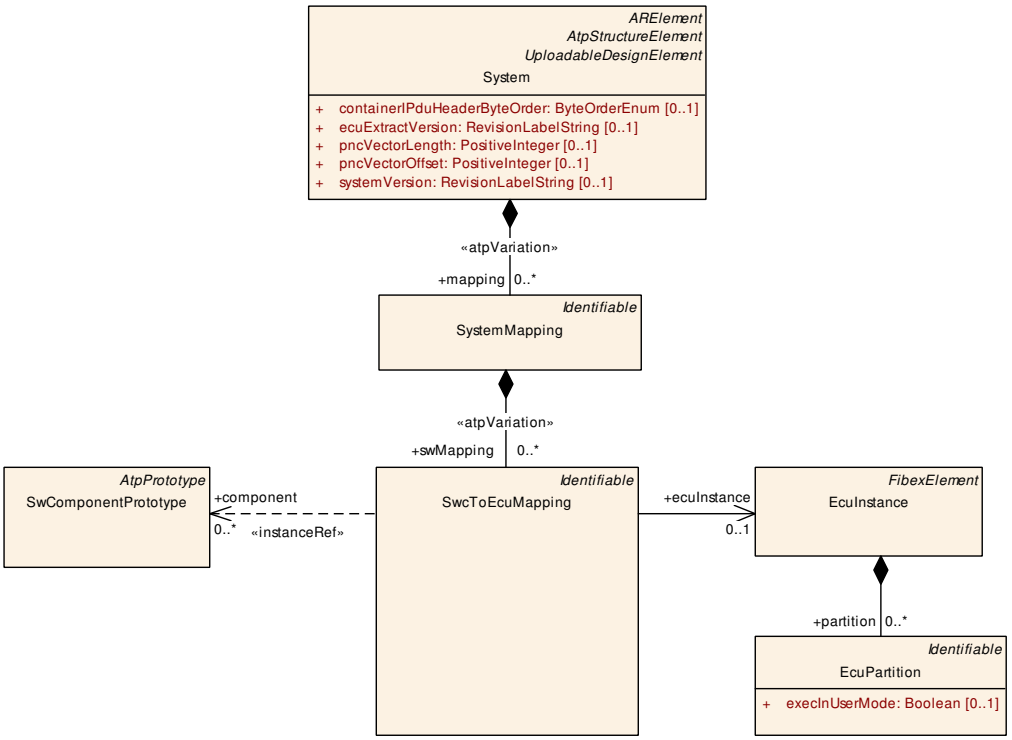

Figure 5.2: SW component to ECU mapping (SwcToEcuMapping)

The SwcToEcuMapping collects a list of all SwComponentPrototypes that shall be deployed onto the associated SwcToEcuMapping targets.

[TPS_SYST_02114] Mapping of SwComponentPrototypes onto EcuInstances
⌈It is equivalent to either

- have several SwcToEcuMappings which map a set of SwcToEcuMapping.components to a SwcToEcuMapping.ecuInstance,
- or one SwcToEcuMapping which maps the set of SwcToEcuMapping.components at once.

⌋

In that sense, the SwcToEcuMapping of SwComponentPrototypes to EcuInstance is arbitrary.

[constr_3263] Restriction of usage of SwcToEcuMapping in a System

Imposition time: CP: IT_SysDesc

⌈For all SwcToEcuMappings in a System the following restriction applies: No two SwcToEcuMappings shall have the exact same reference to

- SwComponentPrototype
- EcuInstance

⌋

SwcToEcuMapping may map either prototypes of AtomicSwComponentType or those of CompositionSwComponentType.

[TPS_SYST_01020] Unconditional mapping of atomic Software Components ⌈In case a prototype of an atomic Software Components is mapped, the mapping is unconditional.⌋

[TPS_SYST_01021] Mapping of CompositionSwComponentType ⌈In case a mapped SwComponentPrototype refers to a CompositionSwComponentType, the mapping is applied to any inner SwComponentPrototype recursively; however, it may be overwritten by additional SwcToEcuMapping mapping inner SwComponentPrototype to different EcuInstances.⌋

Usually a particular component prototype can be mapped explicitly to at most one ECU in a given system (leaving aside variant handling and the implicit mapping of "inner" prototypes mentioned above) but there are two exceptions:

- **[TPS_SYST_01022] Prototype of a ParameterSwComponentType can be mapped to more than one ECU** ⌈A prototype of a ParameterSwComponentType can be mapped to more than one ECU. This is required, because this special component does not communicate over the network, so that a copy of the prototype has to be created on each ECU were it is required.⌋

- **[TPS_SYST_01023] Prototype of an ServiceProxySwComponentType can be mapped to more than one ECU** ⌈A prototype of an ServiceProxySwComponentType can be mapped to more than one ECU even if it appears only once in the VFB system, because a prototype of this special component is required on each ECU, for which local Services are addressed via the proxy.⌋

The following table describes the SwcToEcuMapping in detail.

Class	**SwcToEcuMapping**			
Note	This meta-class is used: • to map SwComponentPrototypes to a specific ECU Instance unit, • optionally to map SwComponentPrototypes to a HwElement with category ProcessingUnit, • optionally to map SwComponentPrototypes typed by SensorActuatorSwComponentType to a Hw Element with category SensorActuator. For each combination of ECUInstance and the optional ProcessingUnit and the optional SensorActuator only one SwcToEcuMapping shall be used. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**SwcToEcuMapping**			
component	SwComponent Prototype	*	iref	References to the software component instances that are mapped to the referenced ECUInstance. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ECU. If there is aditionally a mapping of some SwComponent Prototype INSIDE the Composition to another ECU Instance the inner mapping overrides the outer mapping. **Tags:** atp.Splitkey=component, component.context Composition, component.contextComponent, component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
ecuInstance	EcuInstance	0..1	ref	Reference to a specific ECU Instance description. **Tags:** atp.Splitkey=ecuInstance

Table 5.2: SwcToEcuMapping

5.1.2 Software Component to Implementation Mapping

As several implementations may exist for the same AtomicSwComponentType, it needs to be decided on and specified which instances of a given AtomicSwComponentType are mapped to which Implementation. According to the AUTOSAR Methodology this information can either be added within the Configure System activity, or later when the RTE part is configured during Configure ECU phase. If the mapping is done in System Configuration, a SwcToImplMapping is being used for assigning one Implementation to one or more instances of SwComponentPrototype relating to the same AtomicSwComponentType. This is illustrated in Figure 5.3.

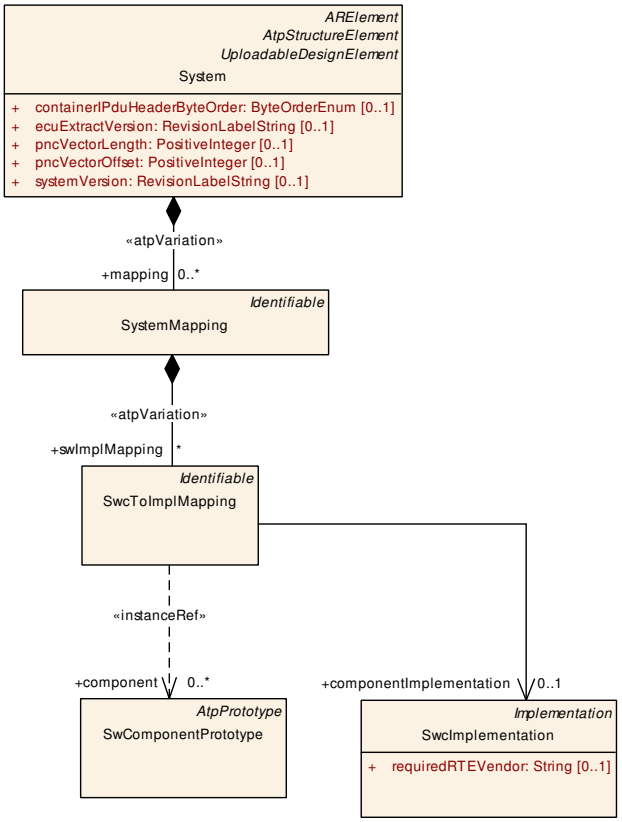

Figure 5.3: SW Component to Implementation mapping (SwcToImplMapping)

[constr_3002] valid swcToImplMapping

Imposition time: CP: IT_EcuExt

⌈The referenced SwcImplementation refers to a SwcInternalBehavior that is part of a AtomicSwComponentType. The same AtomicSwComponentType shall be the type of the referenced SwComponentPrototype.

SwcToImplMapping.componentImplementation.behavior.component == SwcToImplMapping.component.type⌋

The following table contains the detailed description of SwcToImplMapping:

Class	**SwcToImplMapping**			
Note	Map instances of an AtomicSwComponentType to a specific Implementation. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swImplMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
component	SwComponent Prototype	*	iref	Reference to the software component instances that are being mapped to the specified Implementation. The targeted SwComponentPrototype needs be of the Atomic SwComponentType being implemented by the referenced Implementation. **Tags:** atp.Splitkey=component, component.context Composition, component.contextComponent, component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
component Implementation	SwcImplementation	0..1	ref	Reference to a specific Implementation description. Implementation to be used by the specified SW component instance. This allows to achieve more precise estimates for the resource consumption that results from mapping the instance of an atomic SW component onto an ECU. **Tags:** atp.Splitkey=componentImplementation

Table 5.3: SwcToImplMapping

[constr_5488] Existence of SwcToImplMapping.component

Imposition time: CP: IT_EcuExt

⌈For each SwcToImplMapping, the reference to SwComponentPrototype in the role component shall exist at least once.⌋

[constr_5489] Existence of SwcToImplMapping.componentImplementation

Imposition time: CP: IT_EcuExt

⌈For each SwcToImplMapping, the reference to SwcImplementation in the role componentImplementation shall exist.⌋

5.1.3 SW Component to Partition Mapping

With the SwcToApplicationPartitionMapping and the ApplicationPartitionToEcuPartitionMapping an OEM has the option to predefine an allocation to memory partitions in the System Design phase. The final and complete assignment is described in the OS Configuration. The SwcToApplicationPartitionMapping defines a mapping to ApplicationPartitions that allows an allocation to a formally defined partition at a point in time when the EcuInstance is not yet known or defined. In a later methodology step this assignment can be refined with the ApplicationPartitionToEcuPartitionMapping to an EcuPartition defined in the context of an EcuInstance.

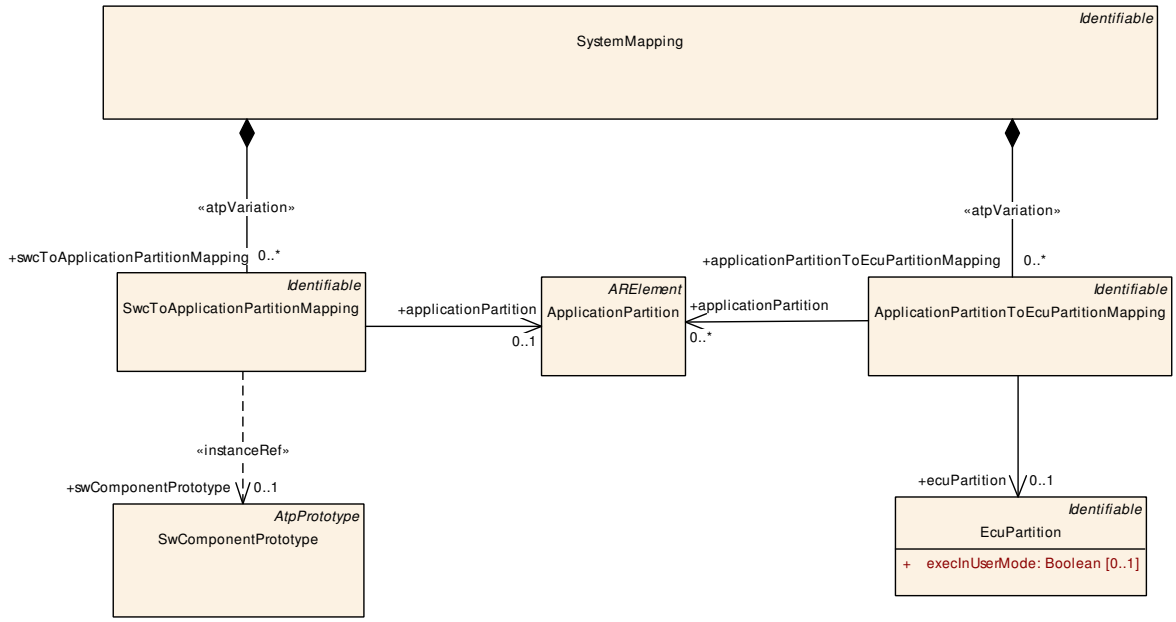

Figure 5.4: SW Component to Application Partition mapping

Class	**SwcToApplicationPartitionMapping**			
Note	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.swcToApplicationPartitionMapping, SwComponentMapping Constraints.swcToApplicationPartitionMapping, SystemMapping.swcToApplicationPartitionMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Partition	ApplicationPartition	0..1	ref	Reference to an ApplicationPartition to which a Sw ComponentPrototype is mapped. **Tags:** atp.Splitkey=applicationPartition
swComponent Prototype	SwComponent Prototype	0..1	iref	References to the software component instances that are mapped to the referenced ApplicationPartition. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ApplicationPartition. If there is additionally a mapping of some SwComponent Prototype INSIDE the Composition to another Application Partition the inner mapping overrides the outer mapping. **Tags:** atp.Splitkey=swComponentPrototype, sw ComponentPrototype.contextComposition, swComponent Prototype.contextComponent, swComponent Prototype.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table 5.4: SwcToApplicationPartitionMapping

Class	ApplicationPartition			
Note	ApplicationPartition to which SwComponentPrototypes are mapped at a point in time when the corresponding EcuInstance is not yet known or defined. In a later methodology step the Application Partition can be assigned to an EcuPartition. **Tags:** atp.recommendedPackage=ApplicationPartitions This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 5.5: ApplicationPartition

Class	ApplicationPartitionToEcuPartitionMapping			
Note	Maps ApplicationPartitions to EcuPartitions. With this mapping an OEM has the option to predefine an allocation of Software Components to EcuPartitions in the System Design phase. The final and complete assignment is described in the OS Configuration. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.applicationPartitionToEcuPartitionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Partition	ApplicationPartition	*	ref	Reference to ApplicationPartitions that are mapped to an EcuPartition. **Tags:** atp.Splitkey=applicationPartition
ecuPartition	EcuPartition	0..1	ref	Reference to EcuPartition to which the Application Partitions are assigned. **Tags:** atp.Splitkey=ecuPartition

Table 5.6: ApplicationPartitionToEcuPartitionMapping

Class	EcuPartition			
Note	Partitions are used as error containment regions. They permit the grouping of SWCs and resources and allow to describe recovery policies individually for each partition. Partitions can be terminated or restarted during run-time as a result of a detected error. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.partition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
execInUser Mode	Boolean	0..1	attr	A partition can execute either in CPU user mode (execIn UserMode = TRUE) or supervisor mode (execInUser Mode = FALSE). In user mode, the partition has a limited access to memory, to memory mapped hardware and to CPU. In user mode, the partition is mapped to a non-trusted OS-Application.

Table 5.7: EcuPartition

[constr_3232] `ApplicationPartition` is allowed to be mapped to only one `EcuPartition`

Imposition time: CP: `IT_EcuExt`

⌈Each `ApplicationPartition` shall be mapped at most once to an `EcuPartition` via the `ApplicationPartitionToEcuPartitionMapping`.⌋

[constr_3229] `SwComponentPrototype` mapped to an `ApplicationPartition` and `EcuInstance`

Imposition time: CP: `IT_EcuExt`

⌈If the `SwcToEcuMapping.ecuInstance` exists then a `SwComponentPrototype` that is mapped to an `ApplicationPartition` via the `SwcToApplicationPartitionMapping` shall only be mapped by an `ApplicationPartitionToEcuPartitionMapping` to an `EcuPartition` that is aggregated by the `EcuInstance` referenced by means of `SwcToEcuMapping.ecuInstance`.⌋

5.1.4 Software Component Mapping Constraints

In contrast to the mapping description described in the previous chapters, mapping constraints allow to define invariants that have to be fulfilled by a valid mapping. They are aggregated in the `MappingConstraint` element as introduced in chapter 5 and depicted Figure 5.1. This chapter describes which mapping constraints can be described in the System Constraint Description. The description of this meta-class can be found in the following table:

Class	***MappingConstraint*** (abstract)			
Note	Different constraints that may be used to limit the mapping of SW components to applicable ECUs, Partitions or Cores depending on the mappingScope attribute.			
Base	*ARObject*			
Subclasses	ComponentClustering, ComponentSeparation			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the mapping constraint. **Tags:** atp.Splitkey=introduction

Table 5.8: MappingConstraint

The two constraints (`ComponentClustering` and `ComponentSeparation`) shown in Figure 5.5 express the restrictions that Software Components impose on each other when performing the mapping onto the ECUs, Cores or Partitions. In fact, before the mapping process begins, it can be useful to impose the allocation of a predefined set of SW components onto the same ECU, especially if such a set is tightly linked from a functional point of view. In the same way, two critical SW components, performing some kind of redundancy, may be not suitable to run both on the same ECU. Thus, we call these two kinds of mapping constraints, respectively, `ComponentClustering` and `ComponentSeparation`.

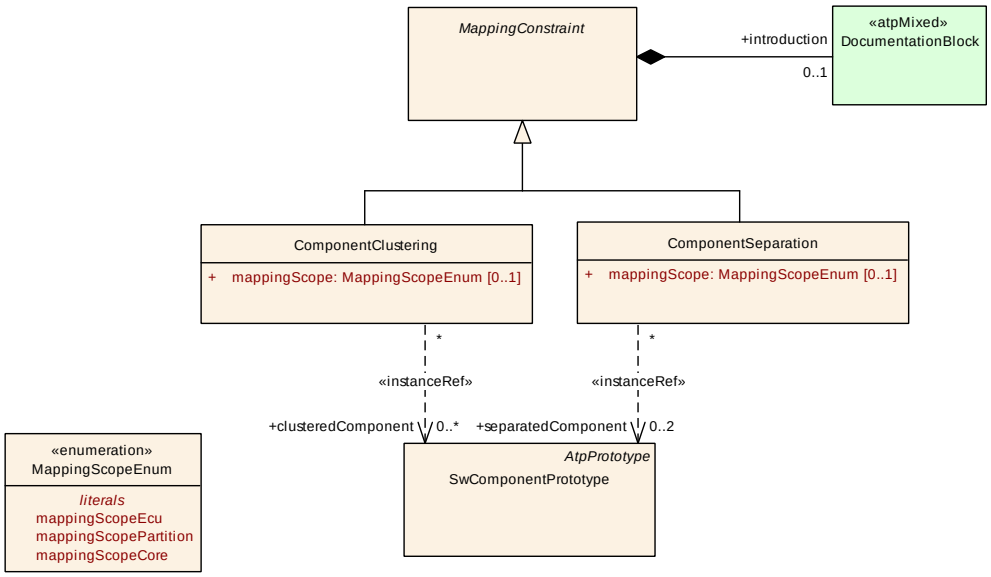

Figure 5.5: Details on ComponentClustering and ComponentSeparation (SwcClustering)

5.1.4.1 ComponentClustering

[TPS_SYST_01024] Component Clustering ⌈The `ComponentClustering` constraint (also, *clustering*) is to be used for expressing that a certain set of SW components (atomic or not) shall be mapped (allocated) onto the same ECU, Core, Partition depending on the defined `mappingScope` attribute.⌋

This is some kind of "execute together on same ECU" constraint.

The semantic of the clustering constraint is straightforward if all referenced SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01025] Clustering of Compositions ⌈All of the atomic SW components making up the composition shall be mapped onto the same ECU, Core, Partition depending on the defined `mappingScope` attribute together with all other SW components (atomic or not) referenced by the constraint.⌋

This also means that a *clustering* constraint can also refer to only a single composition.

A *clustering* constraint is part of a `MappingConstraint` element and it shall refer to one or more `SwComponentPrototype` elements, representing the instances of the SW component(s) that shall be mapped together.

Class	**ComponentClustering**
Note	Constraint that forces the mapping of all referenced SW component instances to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mappingScope is not specified then mappingScopeEcu shall be assumed.

▽

△

Class	**ComponentClustering**			
Base	*ARObject*, *MappingConstraint*			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clustered Component	SwComponent Prototype	*	iref	Reference to the components that have to be mapped together. **Tags:** atp.Splitkey=clusteredComponent, clustered Component.contextComposition, clustered Component.contextComponent, clustered Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the ComponentClustering mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScope Ecu shall be assumed.

Table 5.9: ComponentClustering

Enumeration	**MappingScopeEnum**
Note	Defines the scope for the mapping constraints.
Aggregated by	ComponentClustering.mappingScope, ComponentSeparation.mappingScope
Literal	***Description***
mappingScopeCore	The mapping constraint applies to different Cores. **Tags:** atp.EnumerationLiteralIndex=0
mappingScopeEcu	The mapping constraint applies to different Ecus. **Tags:** atp.EnumerationLiteralIndex=1
mappingScope Partition	The mapping constraint applies to different Partitions. **Tags:** atp.EnumerationLiteralIndex=2

Table 5.10: MappingScopeEnum

[constr_5491] Existence of `ComponentClustering.clusteredComponent`

Imposition time: CP: `IT_SysDesc`

⌈For each `ComponentClustering`, at least one reference to `SwComponentPrototype` in the role `clusteredComponent` shall exist.⌋

5.1.4.2 ComponentSeparation

[TPS_SYST_01045] Component Seperation ⌈The `ComponentSeparation` constraint (also, *separation*) is to be used for expressing that two SW components (atomic or not) shall not be mapped (allocated) onto the same ECU, Core, Partition depending on the defined `mappingScope` attribute.⌋

This is some kind of “do not execute together on same ECU” constraint.

The semantic of the separation constraint is straightforward if one or both SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01026] Separation of Compositions ⌈Any of the atomic SW components making up the first composition, shall not be mapped onto the same ECU, Core, Partition depending on the defined `mappingScope` attribute with any atomic SW component from the second composition.⌋

As a consequence, and to preserve consistency, an atomic SW component instance cannot be part of two compositions concerned by the same separation constraint, i.e. the two compositions have to be disjoint with regards to component instances[1].

A *separation* constraint is part of a `MappingConstraint` element and it shall refer to two `SwComponentPrototype` elements, representing the two SW component instances that shall not be allocated together.

Class	**ComponentSeparation**			
Note	Constraint that forces the two referenced SW components (called A and B in the following) not to be mapped to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mapping Scope is not specified then mappingScopeEcu shall be assumed. If a SW component (e.g. A) is a composition, none of the atomic SW components making up the A composition shall be mapped together with any of the atomic SW components making up the B composition. Furthermore, A and B shall be disjoint.			
Base	*ARObject*, *MappingConstraint*			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the Component Separation mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScopeEcu shall be assumed.
separated Component	SwComponent Prototype	0..2	iref	The two components that have to be mapped to different ECUs **Tags:** atp.Splitkey=separatedComponent, separated Component.contextComposition, separated Component.contextComponent, separated Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table 5.11: ComponentSeparation

[constr_3004] Clustering and separation shall be exclusive

Imposition time: CP: `IT_SysDesc`

⌈Clustering and separation shall be exclusive, i.e. it SHALL NOT be possible that two `SwComponentPrototype`s A and B are associated both by a `ComponentClustering` and by a `ComponentSeparation` at the same time.⌋

Please note that it is possible that one `SwComponentPrototype` is referenced by both the `ComponentClustering` and `ComponentSeparation`, so the [constr_3004] is about the pair of `SwComponentPrototype`s and not about a single one.

[1] The only case where a component instance could be in both sets is if the `ComponentSeparation` refers to two elements where one of them is a substructure of the other. Consider the case that Atomic SW Component A is aggregated by composition B, which in turn is aggregated by composition C. Then instance A is both in B and C. It is not a good idea to formulate a separation constraint stating that B and C should not be on the same ECU.

For example it shall be possible to associate ComponentClustering and ComponentSeparation at the same time, e.g.

- A and B have a ComponentClustering_1 association
- B and C have a ComponentSeparation_2 association
- A and D have a ComponentSeparation_3 association

In this setup A and B are associated by a ComponentClustering and by a ComponentSeparation without violating [constr_3004].

[constr_5492] Existence of ComponentSeparation.separatedComponent

Imposition time: CP: IT_SysDesc

⌈For each ComponentSeparation always two references to SwComponentPrototypes in the role separatedComponent shall exist.⌋

5.1.5 Affinity Constraints

This chapter defines the possibility to describe constraints for the mapping of RTEEvents to OsTasks. The meta-class OsTaskProxy is used in the System Description to represent an OsTask that is defined in the Ecu Configuration of the OS.

The mapping of an RTEEvent to an OsTaskProxy can be defined in the context of the System (see chapter 5.1.5.2) or in the context of a Software Composition (see chapter 5.1.5.1).

Class	***RTEEvent*** (abstract)			
Note	Abstract base class for all RTE-related events			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Subclasses	AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, OsTaskExecutionEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, TimingEvent, TransformerHardErrorEvent			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. **Tags:** atp.Splitkey=disabledMode, disabledMode.contextPort, disabledMode.contextModeDeclarationGroupPrototype, disabledMode.targetModeDeclaration **InstanceRef implemented by:** RModeInAtomicSwcInstanceRef
startOnEvent	RunnableEntity	0..1	ref	The referenced RunnableEntity starts when the corresponding RTEEvent is raised. **Tags:** atp.Splitkey=startOnEvent

Table 5.12: RTEEvent

Class	**OsTaskProxy**			
Note	This meta-class represents a proxy for an OsTask in the System Description. **Tags:** atp.recommendedPackage=OsTaskProxies This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
period	TimeValue	0..1	attr	This attribute specifies the period in seconds of this task in case of a cyclically activated task. Please note that this attribute is informative and not directly relevant for the AUTOSAR OS. But the attribute value can be mapped into the OS configuration to support configuration work flows using a fixed set of OsTasks.
preemptability	OsTaskPreemptability Enum	0..1	attr	This attribute defines the preemptability of the task.
priority	PositiveInteger	0..1	attr	This attribute defines the priority of a task as a relative value, i.e. the values show only the relative ordering of the tasks.

Table 5.13: OsTaskProxy

Enumeration	**OsTaskPreemptabilityEnum**
Note	Enumeration that defines the possible preemptability values for OsTask. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	OsTaskProxy.preemptability
Literal	***Description***
full	Task is preemptable. **Tags:** atp.EnumerationLiteralIndex=1
none	Task is not preemptable. **Tags:** atp.EnumerationLiteralIndex=0

Table 5.14: OsTaskPreemptabilityEnum

The software component specific `OsTaskProxy` definitions can be mapped to `OsTaskProxy` definitions that are defined for a specific `EcuInstance` with the `AppOsTaskProxyToEcuTaskProxyMapping`. `OsTaskProxy` elements that are related to an `EcuInstance` are referenced by the `EcuInstance` in the role `ecuTaskProxy`.

[TPS_SYST_02367] Execution Order of `RTEEvent`s on a `EcuInstance` ⌈Software component specific `OsTaskProxy` elements (`appTaskProxy`) that are mapped by `AppOsTaskProxyToEcuTaskProxyMapping`s to the same `ecuTaskProxy` shall be mapped together to the same OsTask. Optionally the execution order of software component related `OsTaskProxy` elements in the `ecuTaskProxy` can be defined with the `offset` attribute.⌋

[TPS_SYST_02438] Types of OsTaskProxy ⌈There are different ways to use an `OsTaskProxy`:

- Purely in the context of a `SwComponentType`. This is called `AppTaskProxy`. An `OsTaskProxy` is an `AppTaskProxy`, if it is only referenced by `RteEventInCompositionToOsTaskProxyMapping`s (in the role `osTaskProxy`), and

NOT referenced by any EcuInstance (in the role ecuTaskProxy). This property can (optionally) be declared by setting the OsTaskProxy's category to APP_TASK_PROXY.

- Purely in the context of an EcuInstance. This is called EcuTaskProxy. An OsTaskProxy is an EcuTaskProxy, if it is only referenced by an EcuInstance (in the role ecuTaskProxy), and NOT referenced by any RteEventInCompositionToOsTaskProxyMapping (in the role osTaskProxy). This property can (optionally) be declared by setting the OsTaskProxy's category to ECU_TASK_PROXY.
- Both in context of a SwComponentType and an EcuInstance. This is called AppAndEcuTaskProxy. An OsTaskProxy is an AppAndEcuTaskProxy, if it is referenced to by an EcuInstance (in the role ecuTaskProxy), AND referenced by at leat one RteEventInCompositionToOsTaskProxyMapping (in the role osTaskProxy). This property can (optionally) be declared by setting the OsTaskProxy's category to APP_AND_ECU_TASK_PROXY.

⌋

Note that the category is optional. If it is set, it is possible to check that the usage (the type) matches the declared category. If no category is declared, the OsTaskProxy can be used as any of the above types, the type is then defined by the usage.

By definition, one OsTaskProxy cannot have two types within a single model. In different models, however, it can be used in different ways, and therefore have a different type, if the category is not set.

[constr_9382] AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy shall only target AppTaskProxy

Imposition time: CP: IT_SysDesc

⌈The reference AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy shall only refer to an OsTaskProxy that fulfills the definition of an AppTaskProxy according to [TPS_SYST_02438]. It shall not target an EcuTaskProxy or an AppAndEcuTaskProxy.⌋

[constr_9383] AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy shall only target EcuTaskProxy

Imposition time: CP: IT_SysDesc

⌈The reference AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy shall only refer to an OsTaskProxy that fulfills the definition of an EcuTaskProxy according to [TPS_SYST_02438]. It shall not target an AppTaskProxy or an AppAndEcuTaskProxy.⌋

The two constraints above imply that it is not allowed that the same OsTaskProxy is referenced by an AppOsTaskProxyToEcuTaskProxyMapping in the role ecuTaskProxy, and by any (the same or another) AppOsTaskProxyToEcuTaskProxyMapping in the role appTaskProxy.

The constraints also show that the usage of AppOsTaskProxyToEcuTaskProxyMapping is optional - if an OsTaskProxy is used as a AppAndEcuTaskProxy it cannot be (and does not need to be) referenced by an AppOsTaskProxyToEcuTaskProxyMapping.

If the category is not set, one OsTaskProxy can, without any change to the element, be used as an AppTaskProxy in one model (with or without a reference from an AppOsTaskProxyToEcuTaskProxyMapping) and as an AppAndEcuTaskProxy in another model (referenced from an EcuInstance and with no reference from an AppOsTaskProxyToEcuTaskProxyMapping). This usecase is explained later in this section. In theory, the same is true for EcuTaskProxy and AppAndEcuTaskProxy, but there is no known usecase for this combination.

[TPS_SYST_02439] OsTaskProxy offset is optional ⌈The attributes AppOsTaskProxyToEcuTaskProxyMapping.offset and RteEventInCompositionToOsTaskProxyMapping.offset are optional. If the offset attribute value is not defined, the respective elements can be scheduled in any order.⌋

Class	**AppOsTaskProxyToEcuTaskProxyMapping**			
Note	This meta-class is used to map an OsTaskProxy that was created in the context of a SwComponent to an OsTaskProxy that was created in the context of an Ecu. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.appOsTaskProxyToEcuTaskProxyMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
appTaskProxy	OsTaskProxy	0..1	ref	Reference to an OsTaskProxy that is created in the context of a SwComponent. **Tags:** atp.Splitkey=appTaskProxy
ecuTaskProxy	OsTaskProxy	0..1	ref	Reference to an OsTaskProxy that is created in the context of an EcuInstance. **Tags:** atp.Splitkey=ecuTaskProxy
offset	Integer	0..1	attr	This attribute is used to describe the position of the app TaskProxy in an ecuTaskProxy as a relative value, i.e. the values show only the relative position of the appTask Proxy in the ecuTaskProxy.

Table 5.15: AppOsTaskProxyToEcuTaskProxyMapping

The following figure shows the mapping approach where RTEEvents of Software Component A are mapped to TaskProxy1 with offset 1 and 4 and RTEEvents of Software Component B are mapped to TaskProxy2 with offset 2 and 3.

Both Software Components are mapped to the same EcuInstance and AppOsTaskProxyToEcuTaskProxyMappings that are referencing the ecuTaskProxy are defining the execution order on the EcuInstance with the offset attribute. According to the definition the RTEEvents of Software Component A (with offset 1 and 4) are executed first. And the RTEEvents of Software Component B (with offset 2 and 3) are executed afterwards.

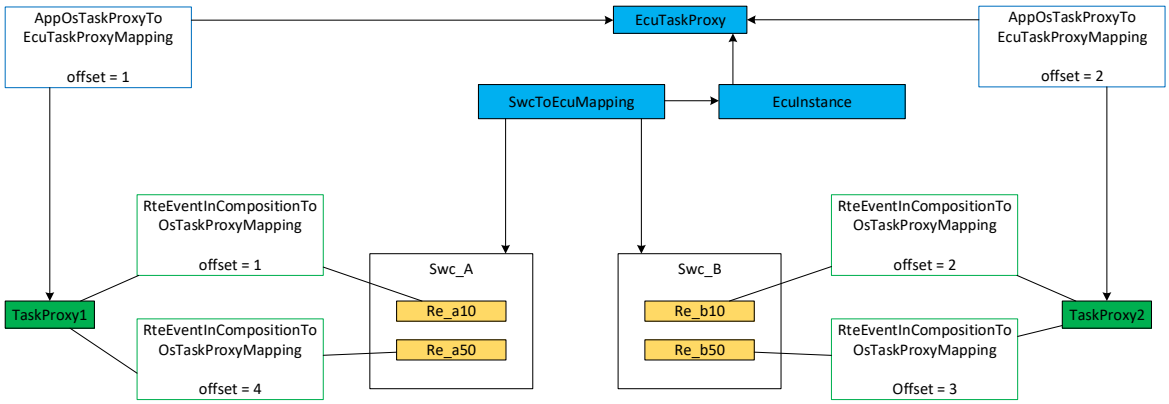

Figure 5.6: AppOsTaskProxyToEcuTaskProxyMapping example

Please note that it is allowed to have several AppOsTaskProxyToEcuTaskProxyMappings that are referencing the same OsTaskProxy in the role ecuTaskProxy that have the same offset value defined. In this case, the relative execution order between elements with the same offset on the EcuInstance is not specified by the system designer. Instead, it is determined by the Integrator (or can depend on the implementation of a tool that derives the configuration from the upstream mapping).

The same applies to RteEventInCompositionToOsTaskProxyMapping: If several RteEventInCompositionToOsTaskProxyMappings are defined that are referencing the same OsTaskProxy, and all those RteEventInCompositionToOsTaskProxyMappings have the same offset defined, then the order of the corresponding RTEEvents in the OsTaskProxy is not specified by the system designer.

Relation to TIMEX

The Timing Extensions (TIMEX, see [7]) also offer a way to design the relative order of RTEEvents, which can be used as an alternative to Affinity Constraints.

The two methods follow different approaches to scheduling: Affinity Constraints define an explicit order of elements via the offset attributes, while TIMEX defines an order as a chain of relationships between runnables or events. The former is similar to a fixed scheduling design, the latter similar to requirements that lead to a design.

Affinity Constraints offer a straightforward way to describe a desired scheduling order in a familiar format. In contrast, TIMEX is a complete toolset for timing design, allowing, for example, to also define effect chains on signal level, at the cost of a more abstract representation.

5.1.5.1 RteEvent to OsTaskProxy mapping constraints in the context of a Software Composition

This section describes constraints for the mapping of RTEEvents to OsTasks in the context of a Software Composition.

[TPS_SYST_02368] RTEEvent pairing constraint in Software Composition context ⌈RTEEvents defined in the context of a CompositionSwComponentType that are mapped by RteEventInCompositionToOsTaskProxyMappings to the same OsTaskProxy shall be mapped together to the same OsTask. Optionally an order of the RTEEvents in the OsTaskProxy can be defined with the offset attribute.⌋

[TPS_SYST_02369] RTEEvent separation constraint in Software Composition context ⌈RTEEvents defined in the context of a CompositionSwComponentType that are referenced by RteEventInCompositionSeparation are not allowed to be mapped to the same OsTask.⌋

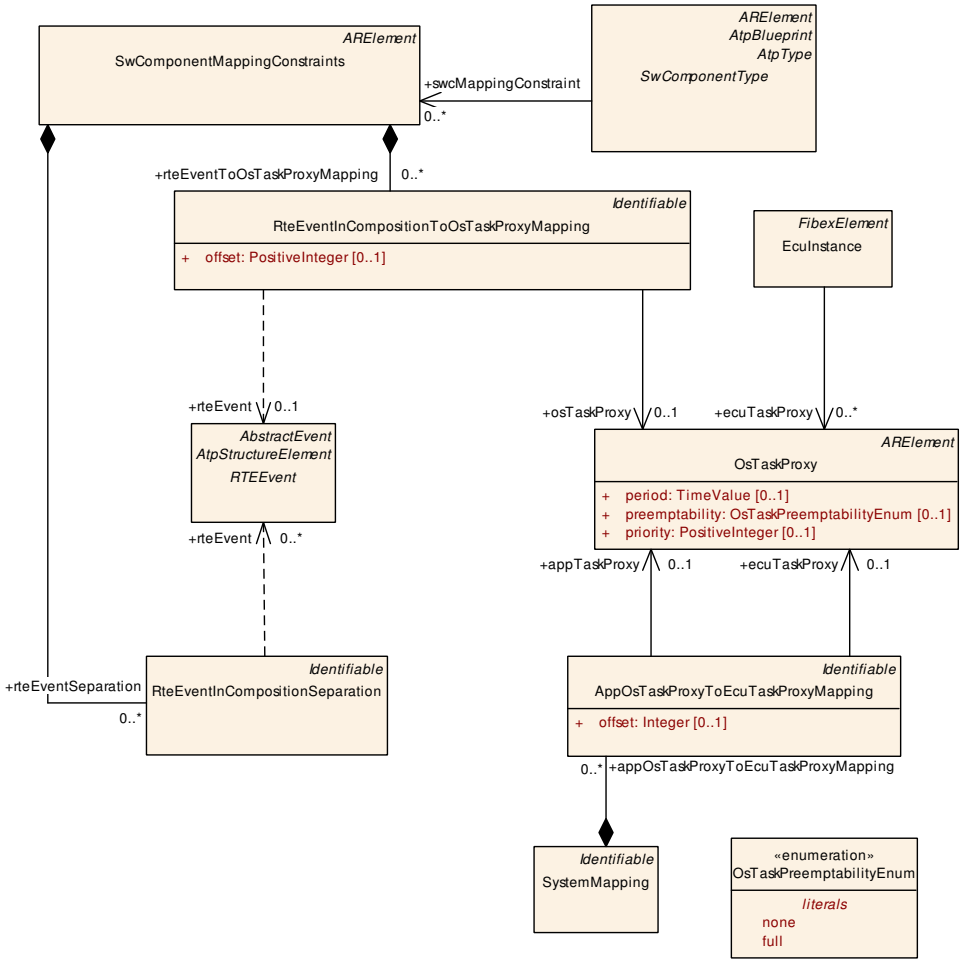

Figure 5.7: Mapping of Rte Events to Os Tasks in Software Composition context

Class	SwComponentMappingConstraints			
Note	Collection of constraints that are valid for particular SwComponentTypes. **Tags:** atp.recommendedPackage=SwComponentMappingConstraintCollection This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
rteEvent Separation	RteEventInComposition Separation	*	aggr	Separation constraint that limits the mapping freedom for the mapping of RteEvents to OsTasks in the context of a SwComponent. **Tags:** atp.Splitkey=rteEventSeparation, rteEvent Separation.shortName
rteEventToOs TaskProxy Mapping	RteEventInComposition ToOsTaskProxyMapping	*	aggr	Constraint that enforces a mapping of RteEvent to a particular OsTask in the context of a SwComponent. **Tags:** atp.Splitkey=rteEventToOsTaskProxyMapping, rte EventToOsTaskProxyMapping.shortName
swcTo Application Partition Mapping	SwcToApplication PartitionMapping	*	aggr	Constraint that enforces a mapping of a Swc to a particular Application Partition. **Tags:** atp.Splitkey=swcToApplicationPartitionMapping, swcToApplicationPartitionMapping.shortName

Table 5.16: SwComponentMappingConstraints

Class	RteEventInCompositionToOsTaskProxyMapping			
Note	This meta-class is used to map an RteEvent to an OsTaskProxy in the context of a SwComposition. Several RteEventInCompositionToOsTaskProxyMappings can be used to define a pairing constraint that describes which RteEvents shall be mapped together into an OsTask. Optionally the relative position of the RteEvents in the OsTask can be defined in the mapping. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SwComponentMappingConstraints.rteEventToOsTaskProxyMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
offset	PositiveInteger	0..1	attr	This attribute is used to describe the position of the Rte Event in the OsTask as a relative value, i.e. the values show only the relative position of the RteEvent in the Os Task.
osTaskProxy	OsTaskProxy	0..1	ref	Reference to OsTaskProxy to which the RteEvent is mapped. **Tags:** atp.Splitkey=osTaskProxy
rteEvent	RTEEvent	0..1	iref	Reference to RteEvent that is mapped to the OsTask Proxy. **Tags:** atp.Splitkey=rteEvent, rteEvent.contextSw Component, rteEvent.targetRteEvent **InstanceRef implemented by:** RteEventInComposition InstanceRef

Table 5.17: RteEventInCompositionToOsTaskProxyMapping

Class	RteEventInCompositionSeparation
Note	This meta-class is used to define a separation constraint in the context of a SwComposition. The referenced RteEvents are not allowed to be mapped into the same OsTask. This Class is only used by the AUTOSAR Classic Platform.



△

Class	RteEventInCompositionSeparation			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	SwComponentMappingConstraints.rteEventSeparation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
rteEvent	RTEEvent	*	iref	Reference to RteEvents that are not allowed to be mapped into the same OsTask. **Tags:** atp.Splitkey=rteEvent, rteEvent.contextSw Component, rteEvent.targetRteEvent **InstanceRef implemented by:** RteEventInComposition InstanceRef

Table 5.18: RteEventInCompositionSeparation

5.1.5.2 RteEvent to OsTaskProxy mapping constraints in the context of the System

[TPS_SYST_02370] **`RTEEvent` pairing constraint in `System` context** ⌈`RTEEvent`s defined in the context of a `System` that are mapped by `RteEventInSystemToOsTaskProxyMapping`s to the same `OsTaskProxy` shall be mapped together to the same OsTask. Optionally an order of the `RTEEvent`s in the `OsTaskProxy` can be defined with the `offset` attribute.⌋

[TPS_SYST_02371] **`RTEEvent` separation constraint in `System` context** ⌈`RTEEvent`s defined in the context of a `System` that are referenced by `RteEventInSystemSeparation` are not allowed to be mapped to the same OsTask.⌋

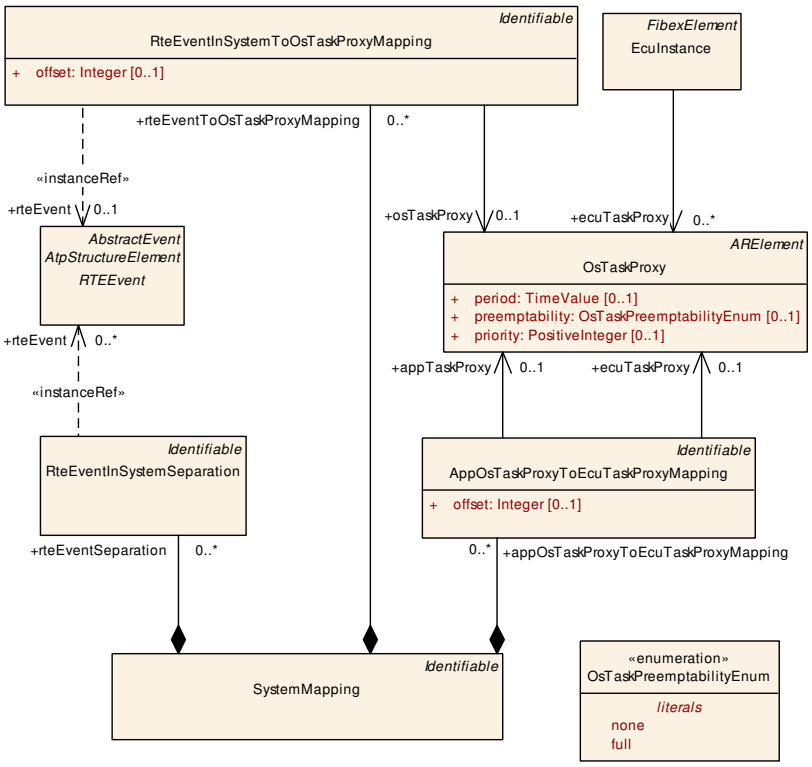

Figure 5.8: Mapping of Rte Events to Os Tasks in System context

Class	**RteEventInSystemToOsTaskProxyMapping**			
Note	This meta-class is used to map an RteEvent to an OsTaskProxy in the context of the System. Several RteEventToOsTaskProxyMappings can be used to define a pairing constraint that describes which RteEvents shall be mapped together into an OsTask. Optionally the position of the RteEvents in the OsTask can be defined. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.rteEventToOsTaskProxyMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
offset	Integer	0..1	attr	This attribute is used to describe the position of the RteEvent in the OsTask as a relative value, i.e. the values show only the relative position of the RteEvent in the OsTask.
osTaskProxy	OsTaskProxy	0..1	ref	Reference to OsTaskProxy to which the RteEvent is mapped. **Tags:** atp.Splitkey=osTaskProxy
rteEvent	RTEEvent	0..1	iref	Reference to RteEvent that is mapped to the OsTaskProxy. **Tags:** atp.Splitkey=rteEvent, rteEvent.contextRootComposition, rteEvent.contextSwComponent, rteEvent.targetRteEvent **InstanceRef implemented by:** RteEventInSystemInstanceRef

Table 5.19: RteEventInSystemToOsTaskProxyMapping

Class	**RteEventInSystemSeparation**			
Note	This meta-class is used to define a separation constraint in the context of the System. The referenced RteEvents are not allowed to be mapped into the same OsTask. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.rteEventSeparation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
rteEvent	RTEEvent	*	iref	Reference to RteEvents that are not allowed to be mapped into the same OsTask. **Tags:** atp.Splitkey=rteEvent, rteEvent.contextRootComposition, rteEvent.contextSwComponent, rteEvent.targetRteEvent **InstanceRef implemented by:** RteEventInSystemInstanceRef

Table 5.20: RteEventInSystemSeparation

The combined application of `AppTaskProxy` and `AppAndEcuTaskProxy` (according to [TPS_SYST_02438]) on the same `OsTaskProxy` is illustrated in figure 5.9 and figure 5.10.

First, the VFB centric definition of the `RteEventInCompositionToOsTaskProxyMapping`s is defined, as shown in figure 5.9. Here, the functional aspects of the Software Components and their Runnable Entities are in focus (VFB). There is no need for any specific `EcuInstance` in this approach. This model is valid, and can be used stand-alone for VFB purposes. In such a scenario, the `OsTaskProxy` is basically used in the role of an `AppTaskProxy`.

A further refinement can be specified, where the RootSwCompositionPrototype and the associated OsTaskProxys are allocated to an EcuInstance. This is shown in figure 5.10. The already defined RteEventInCompositionToOsTaskProxy-Mappings of figure 5.9 can stay as they are, there is just the additional definition of the EcuInstance.ecuTaskProxy. This assignment makes the OsTaskProxy effectively an AppAndEcuTaskProxy.

In such a usage the fixed setting of a OsTaskProxy.category does not make sense, as the same OsTaskProxy model element may be of category AppTaskProxy or AppAndEcuTaskProxy, depending on the model scope. Thus the dynamic categorization of OsTaskProxy according to [TPS_SYST_02438] provides more flexibility on model re-use.

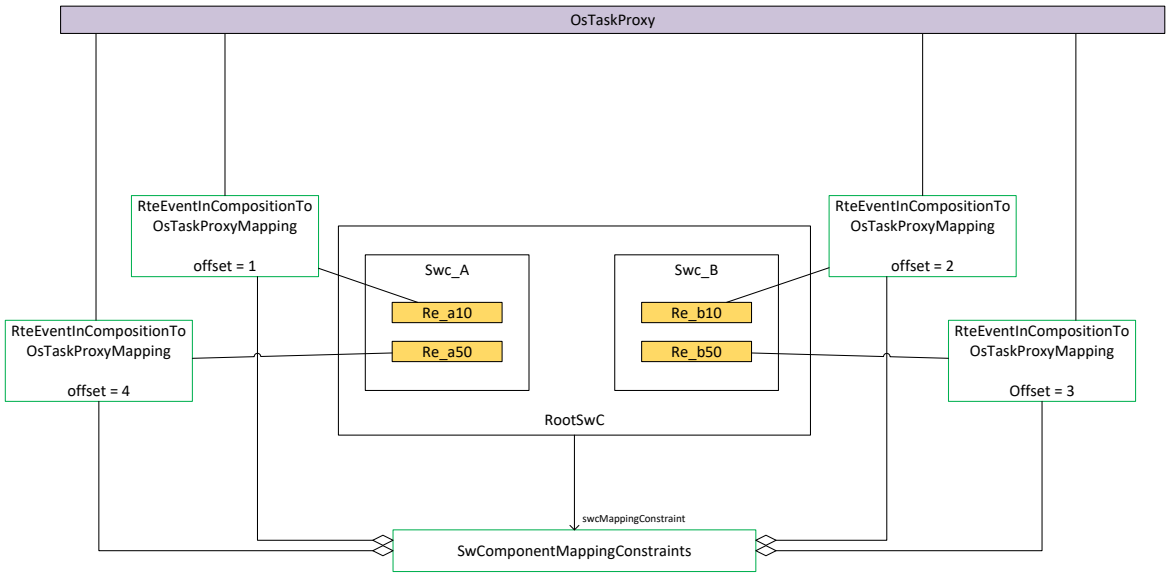

Figure 5.9: RteEventInSystemToOsTaskProxyMapping VFB example

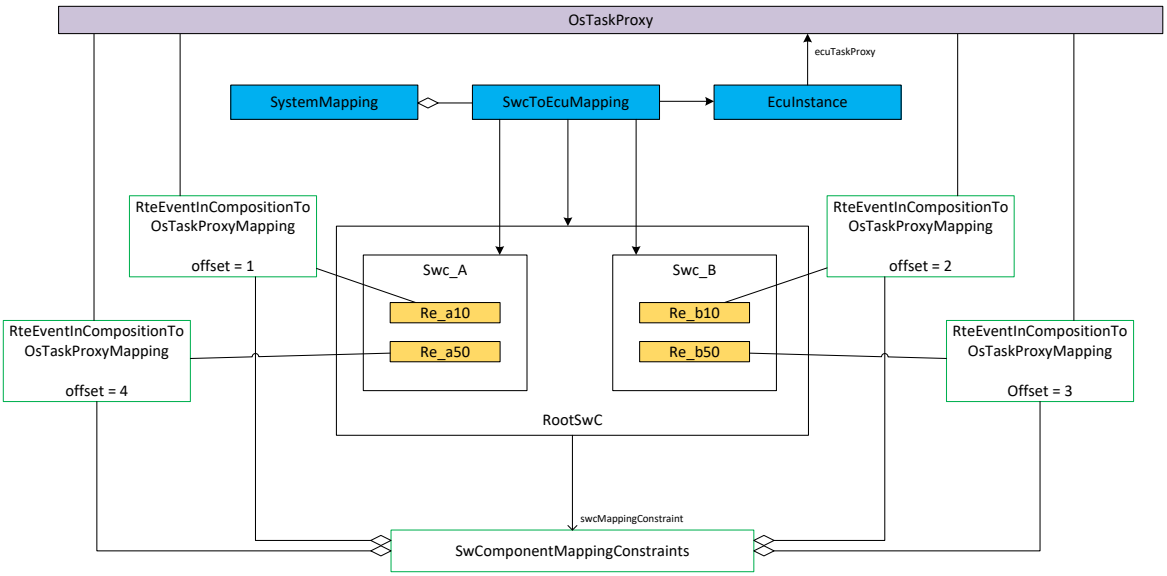

Figure 5.10: RteEventInSystemToOsTaskProxyMapping System example

5.2 Data Mapping

The data mapping description may either be mapping of client server communication or sender receiver communication (see Figure 5.11). It is used to map `VariableDataPrototype`s or `ClientServerOperation`s of SW Component Ports to `SystemSignal`s.

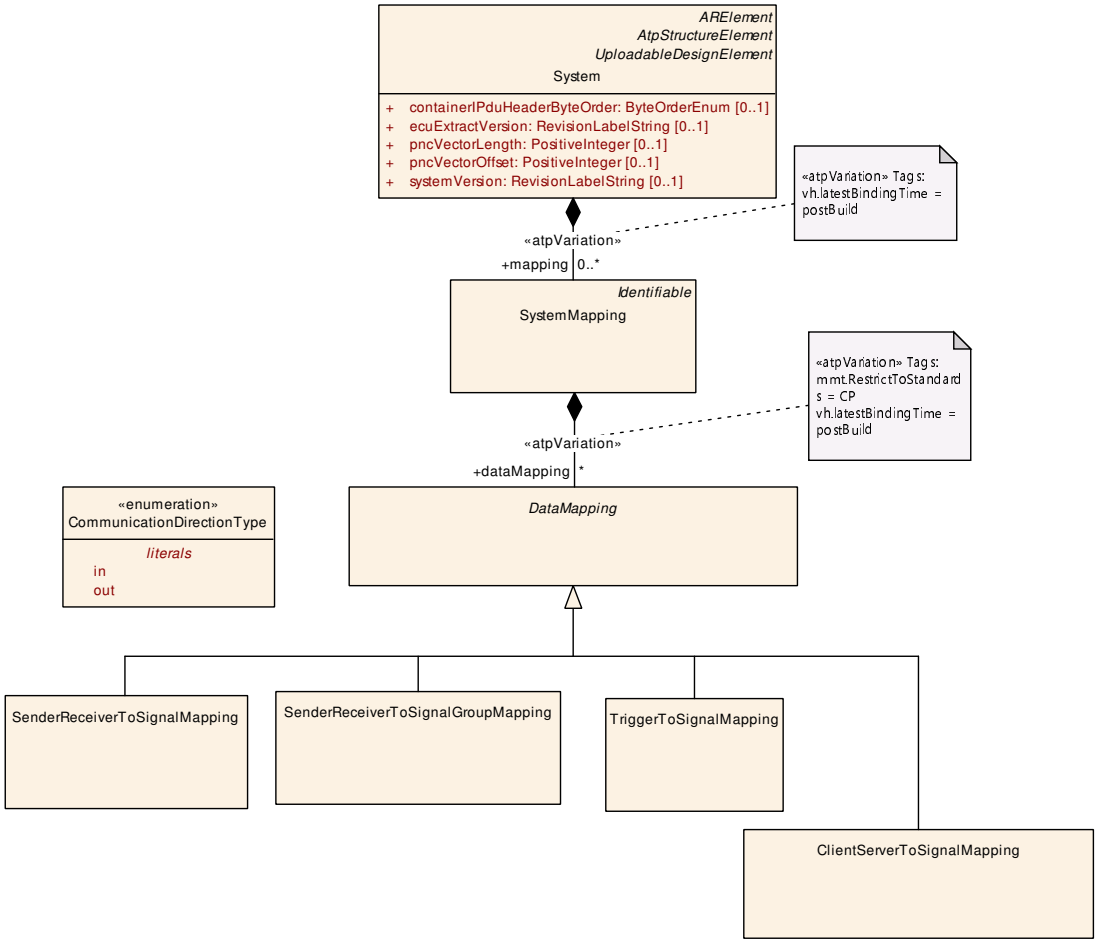

Figure 5.11: Overview: Data Mapping Description (DataMappingOverview)

[TPS_SYST_01030] Representation of VariableDataPrototypes and ClientServerOperations in System Description ⌈SystemSignals represent VariableDataPrototypes and ClientServerOperations in the communication description.⌋

[constr_5055] DataMapping of elements of PRPortPrototypes is not supported

Imposition time: CP: IT_EcuExt

⌈A DataMapping shall not map elements of PRPortPrototypes to SystemSignals⌋

In other words the usage of PRPortPrototypes for inter-Ecu communication is not supported by AUTOSAR.

[constr_9360] DataMapping not available for elements of PortInterfaces with isService = true

Imposition time: CP: IT_EcuExt

⌈VariableDataPrototypes, ClientServerOperations and Triggers within PortPrototypes typed by PortInterfaces in which the isService attribute is set to true shall not be referenced by a DataMapping.⌋

[constr_5266] `VariableDataPrototype` of `NvDataInterface` shall not be mapped to a `SystemSignal`

Imposition time: CP: `IT_EcuExt`

⌈A `VariableDataPrototype` that is aggregated by a `NvDataInterface` shall not be referenced by

- `SenderReceiverToSignalGroupMapping` in the role `dataElement` and
- `SenderReceiverToSignalMapping` in the role `dataElement`.

⌋

[constr_5267] `VariableDataPrototype` of `NvDataInterface` shall not be mapped to a `SystemSignal` via a delegation to a `PortPrototype` with a `SenderReceiverInterface`

Imposition time: CP: `IT_EcuExt`

⌈If a `VariableDataPrototype` that is aggregated by a

- `SenderReceiverInterface` and that `SenderReceiverInterface` is referenced by a `PortPrototype` of a Composition and
- that `PortPrototype` is connected by a delegation connector with an inner `PortPrototype` of a `NvBlockSwComponentType` and
- that `PortPrototype` is typed by a `NvDataInterface`

then this `PortPrototype` shall not be referenced by:

- `SenderReceiverToSignalGroupMapping` in the role `dataElement` and
- `SenderReceiverToSignalMapping` in the role `dataElement`.

⌋

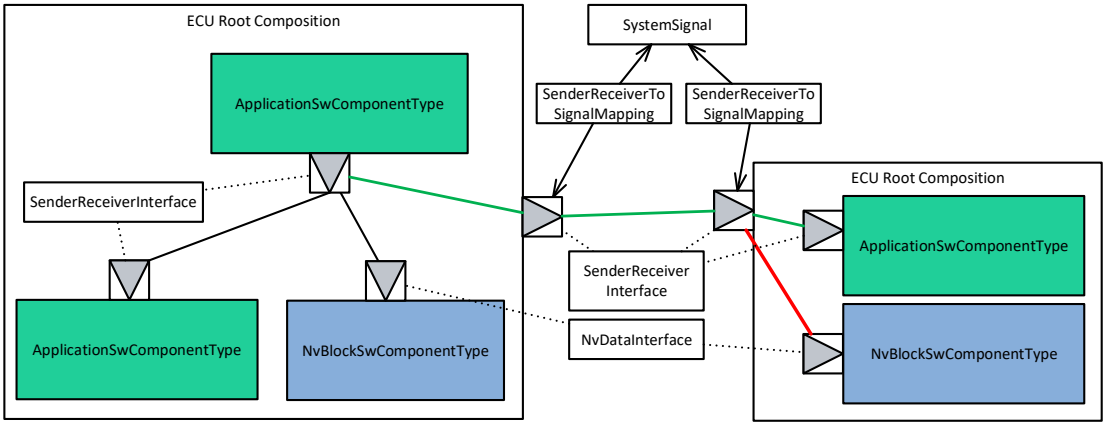

Figure 5.12: Mapping scenario that is forbidden by constr_5267

The figure 5.12 shows the scenario that is forbidden by [constr_5267]. The Port of the NvBlockSwComponentType is delegated to the Ecu Root Composition by the red connector. The VariableDataPrototype of the outer Port that is typed by a SenderReceiverInterface is mapped to a SystemSignal for communication over the network.

A scenario that would be allowed is shown in figure 5.13. Here the NvBlockSwComponentType is connected to an ApplicationSwComponentType in the same Ecu Root Composition. Data that is communicated over the network is contained in an additional Port of the ApplicationSwComponentType that is delegated to the outer Port of the Composition for the definition of the DataMapping. In this scenario [constr_5266] and [constr_5267] are not violated.

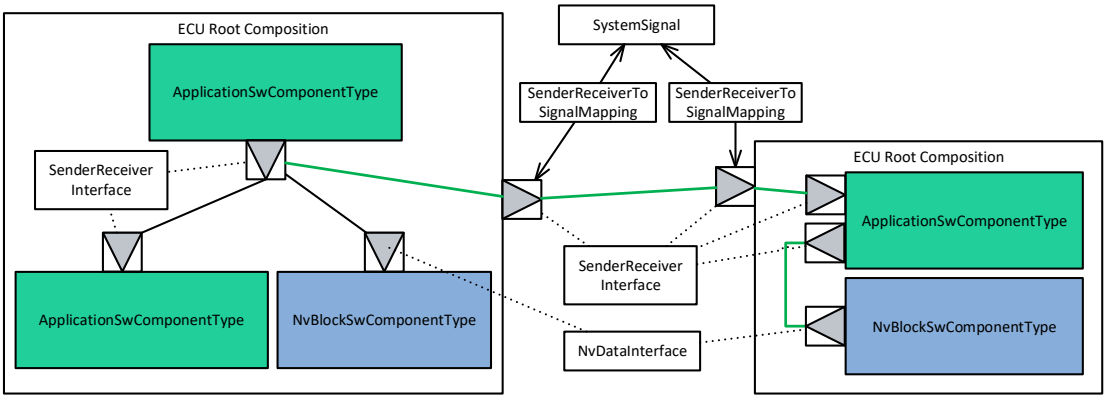

Figure 5.13: Supported mapping scenario

[constr_4000] Local communication of mode switches

Imposition time: CP: IT_EcuExt

⌈Ports with ModeSwitchInterfaces cannot be connected across ECU boundaries.⌋

In other words a DataMapping for ModeDeclarationGroupPrototypes is not supported.

[TPS_SYST_01032] Independence of SystemSignals from CommunicationClusters

⌈The SystemSignals can be defined independently of CommunicationClusters.⌋

This chapter describes how the VariableDataPrototypes and ClientServerOperations are mapped onto SystemSignals. The Communication chapter (6) describes how the SystemSignals are mapped into Pdus and Frames, implementing the actual inter-ECU communication.

Class	***DataMapping*** (abstract)			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	*ARObject*			
Subclasses	ClientServerToSignalMapping, SenderReceiverToSignalGroupMapping, SenderReceiverToSignal Mapping, TriggerToSignalMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping. **Tags:** atp.Splitkey=introduction

Table 5.21: DataMapping

Class	**SystemSignal**			
Note	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances. **Tags:** atp.recommendedPackage=SystemSignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicLength	Boolean	0..1	attr	The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).
physicalProps	SwDataDefProps	0..1	aggr	Specification of the physical representation. **Tags:** atp.Splitkey=physicalProps

Table 5.22: SystemSignal

A `SystemSignal` is used to represent `VariableDataPrototype`s for network transport.

The motivation for `SystemSignal`s is to represent (physical) data which is exchanged between ECUs as a design element. The `SystemSignal` gives data an identity and allows to refer to that information from different context.

`SystemSignal`s are part of the communication matrix and are used as the binding blocks between the Software Component data access and the communication stack infrastructure.

The `SystemSignal`s are mapped to the `PortPrototype`s of Software Components, and the `SystemSignal`s are referenced by `ISignal`s of the communication stack (see Chapter 6.2). `ISignal`s are placed in `ISignalIPdu`s and `Pdu`s are transported on networks.

The creation of the relation between elements of a `PortPrototype` and corresponding `SystemSignal`s has value on its own, both as the bus-independent communication matrix and as a milestone in the workflow towards further downstream processing, where the focus changes to bus-specific descriptions. In other words, if you throw away

the bus specific description, you don't have to start from scratch, thanks to the existence of a stable bus-independent baseline from which the bus-specific part could be derived again.

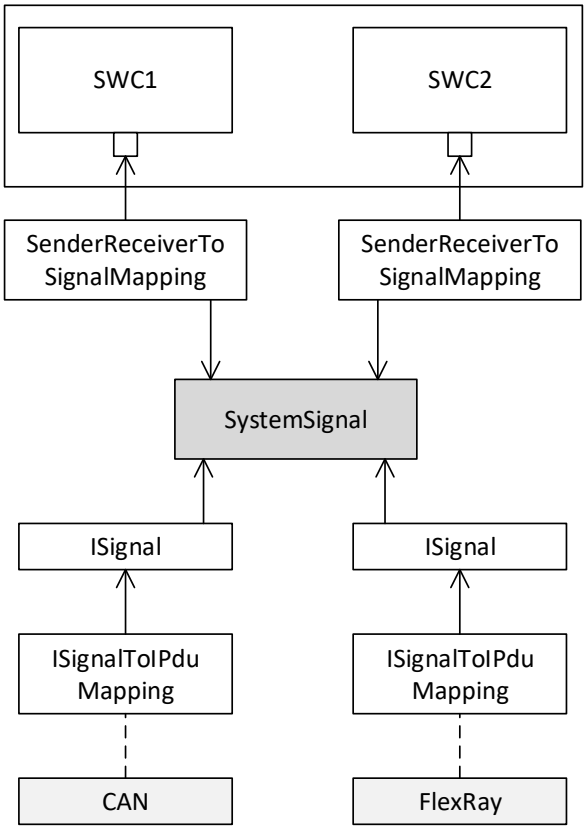

Figure 5.14: Example for a SystemSignal and a DataMapping

[TPS_SYST_01144] Physical properties of a `SystemSignal` ⌈With the aggregation of `SwDataDefProps` in the role `physicalProps` the physical properties of the `SystemSignal` can be specified.⌋

[TPS_SYST_05000] System Description doesn't use a complete Software Component Description ⌈If the System Description doesn't use a complete Software Component Description (VFB View) the data mapping of `VariableDataPrototype`s or `ArgumentDataPrototype`s owned by `ClientServerOperation`s on `SystemSignal`s does not need to be defined. This supports the inclusion of legacy signals.⌋

The data mapping may be defined in various ways to achieve the same behavior. The examples shown in figures 5.15, figure 5.16, and figure 5.17. All data mappings have the goal of mapping the data produced at a software component `PortPrototype` to two sets of `ISignal`s. This use-case is typically applied when data produced by the software component shall be transported on two different networks. One network may be able to transport all of the produced data, while the other network may have a

reduced bandwidth, and thus only a subset of the produced data is actually transmitted on that network.

In the example where *RootComp X* is involved (figure 5.15) the data of the PortPrototype is mapped to one SystemSignalGroup. This SystemSignalGroup uses one set of SystemSignals to define the transported data. Then two sets of ISignalGroups with dedicated ISignals are defined.

One aspect of the data mapping as defined for *RootComp X* is that there exists only one SystemSignalGroup. If the ISignalGroups are transported on different networks, then also the SystemSignalGroup is used on both networks. This leads to the situation that on the network where *ISignalGroup2* is used the corresponding SystemSignalGroup has the SystemSignal *b*, although on that network *b* is not transported.

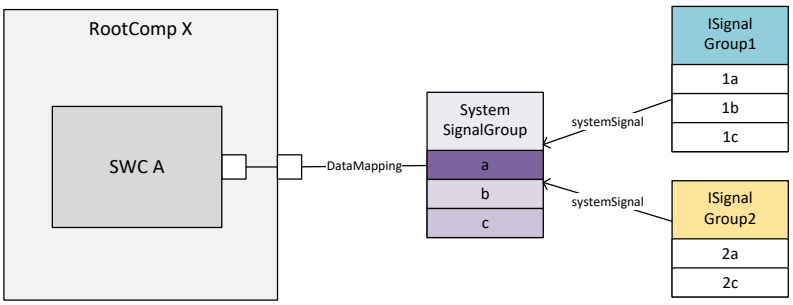

Figure 5.15: Example of possible data mapping definition

That is one motivation why the setup of *RootComp Y* (figure 5.16) may be favoured, as in this example two SystemSignalGroups are used. Note that some of the SystemSignals are actually shared for the definition of both SystemSignalGroups (i.e. SystemSignals *a* and *c* are used by both SystemSignalGroups).

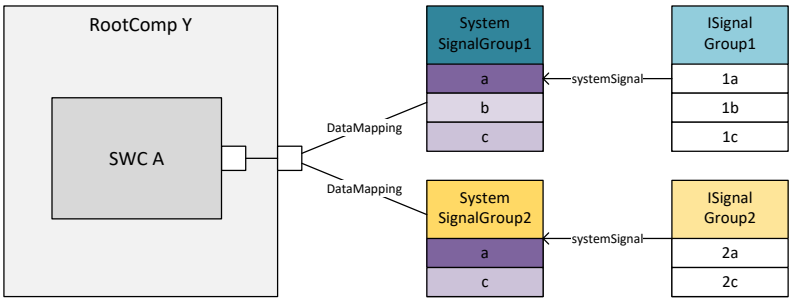

Figure 5.16: Example of possible data mapping definition

Another alternative modeling is shown in the illustration for *RootComp Z* (figure 5.17), where a delegation of the data is already modeled at the RootSwCompositionPrototype. The data mappings are applied in this example individually per delegated PortPrototype to individual SystemSignalGroups with again shared SystemSignals.

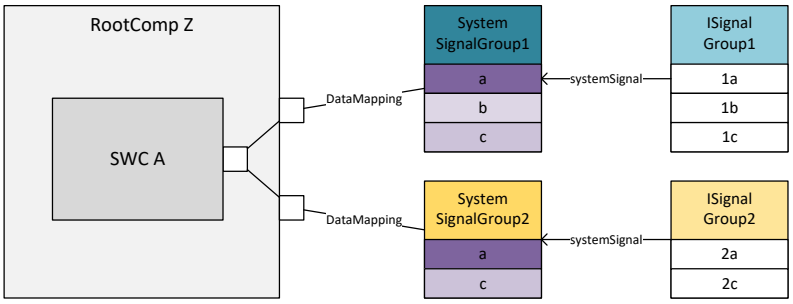

Figure 5.17: Example of possible data mapping definition

Which of the illustrated approaches is use depends on the modeling and export strategy of the system description designer. The RTE behavior shall be the same for all three modeling variants.

The effective behavior of the RTE for the possible data mappings illustrated in the previous figures is shown in figure 5.18.

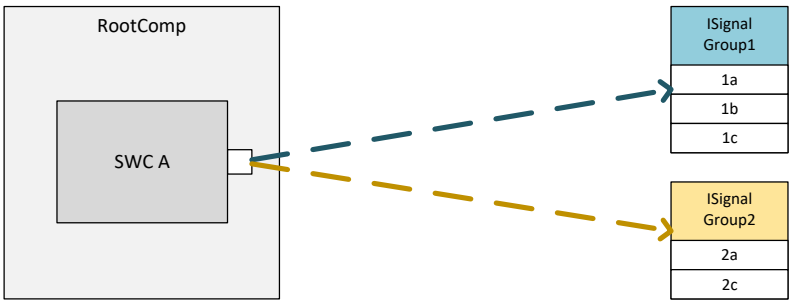

Figure 5.18: Example of resulting effective RTE behavior

[constr_3501] Role of `SystemSignal` in 1:n communication

Imposition time: CP: `IT_EcuExt`

⌈In case of 1:n communication the `VariableDataPrototype` in the `PPortPrototype` of the `SwComponentPrototype` shall be mapped to only one `SystemSignal`.⌋

Note that [constr_3501] is also fulfilled by the modeling approaches illustrated in figures 5.16 and 5.17 where several `SystemSignalGroup`s share some `SystemSignal`s. Such a setup is shown in figure 5.19 from a detailed model perspective.

SystemSignal b is only used in one `SystemSignalGroup`, thus only one data mapping is pointing to *SystemSignal b*.

But *SystemSignal a* and *SystemSignal c* are used in both `SystemSignalGroup`s, thus there are two sets of data mappings defined. But each of the data mappings refer to same pair of *RecordElement* and *SystemSignal*, thus [constr_3501] is still fulfilled.

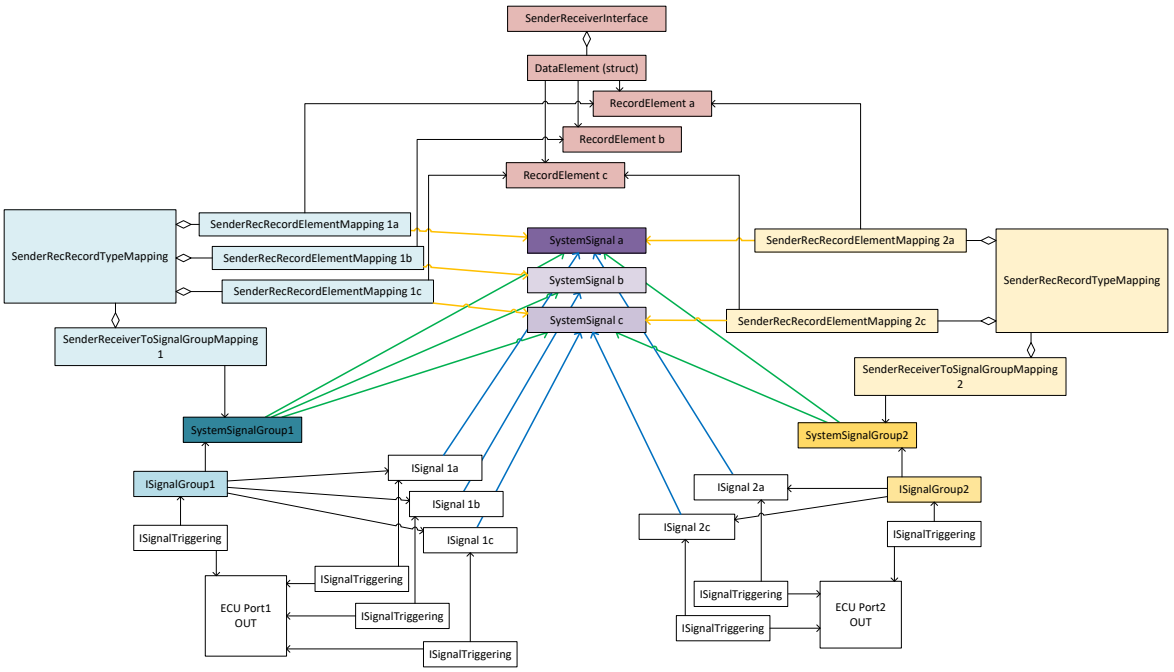

Figure 5.19: Detailed model of several data mappings

In case of figure 5.17 the delegation connectors need to be resolved, which results again in two data mappings with partial identical mapping pairs.

[constr_3086] Role of `SystemSignal` in n:1 sender-receiver communication

Imposition time: CP: `IT_EcuExt`

⌈In case of n:1 communications

- if `DataTransformation` is used each sender shall be mapped to the same `SystemSignal`
- if `DataTransformation` is not used each sender shall be mapped
  - to the same `SystemSignal` in case of a primitive DataType on the sender side,
  - to the same `SystemSignalGroup` in case of a composite DataType on the sender side.

⌋

[constr_5117] Client-Server communication over Ethernet

Imposition time: CP: `IT_SysDesc`

⌈A `SystemSignal` that is referenced by a `ClientServerToSignalMapping` in the role `callSignal` or `returnSignal` shall only be referenced by an `ISignal` that in turn is referenced by an `ISignalTriggering` aggregated by an `EthernetPhysicalChannel`.⌋

In other words, the client-server communication is only supported by AUTOSAR on the Ethernet communication channel.

[TPS_SYST_02150] Role of `SystemSignal` in inter-ECU client server communication over Ethernet with clients located on different ECUs ⌈In case of a n:1 inter-ECU client server communication over Ethernet with clients located on different ECUs exactly one `SystemSignal` per communication direction shall be used to define the client server interaction between the server and all clients since the relationship between the call and return is achieved by means of meta data items attached to the `Pdu`s by the Socket Adapter.⌋

[TPS_SYST_02151] MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs ⌈The modeling of client server interaction over Ethernet with clients located on different ECUs requires the support of COM Stack MetaData. The relationship between the call and return is achieved by means of meta data items attached to the `Pdu`s by the Socket Adapter.⌋

[TPS_SYST_01087] Role of `SystemSignal` in inter-ECU client server communication with clients located on the same ECU ⌈In case of n:1 inter-ECU client server communication it is allowed to use the same `SystemSignal` for several clients on the same Ecu, if the client identifier is used to distinguish the different clients.⌋

[constr_3112] Invalidation support for partial mapping of a data element typed by composite data type

Imposition time: CP: `IT_EcuExt`

⌈If a `VariableDataPrototype` with a composite data type in a `PPortPrototype` is mapped to a `SystemSignalGroup` and only a subset of elements of the composite data type that are primitives is mapped to separate `SystemSignal`s of the `SystemSignalGroup` then at least one mapped primitive shall have an `invalidValue` defined.⌋

[constr_3074] No TransmissionAcknowledgementRequest for multiple senders

Imposition time: CP: `IT_EcuExt`

⌈If more than one `SenderComSpec` exist (in different `PortPrototype`s on atomic level) that refer to data elements effectively mapped to the same `SystemSignal` it is not allowed that any `SenderComSpec` aggregates `transmissionAcknowledge`.⌋

Please note that the term “effectively mapped” refers to the fact that the `DataMapping` can refer to a `dataElement` in a "delegation" `PortPrototype` on the surface of a `rootSoftwareComposition` of an Ecu Extract OR to `PortPrototype`s inside the `rootSoftwareComposition`. Both ways shall be considered.

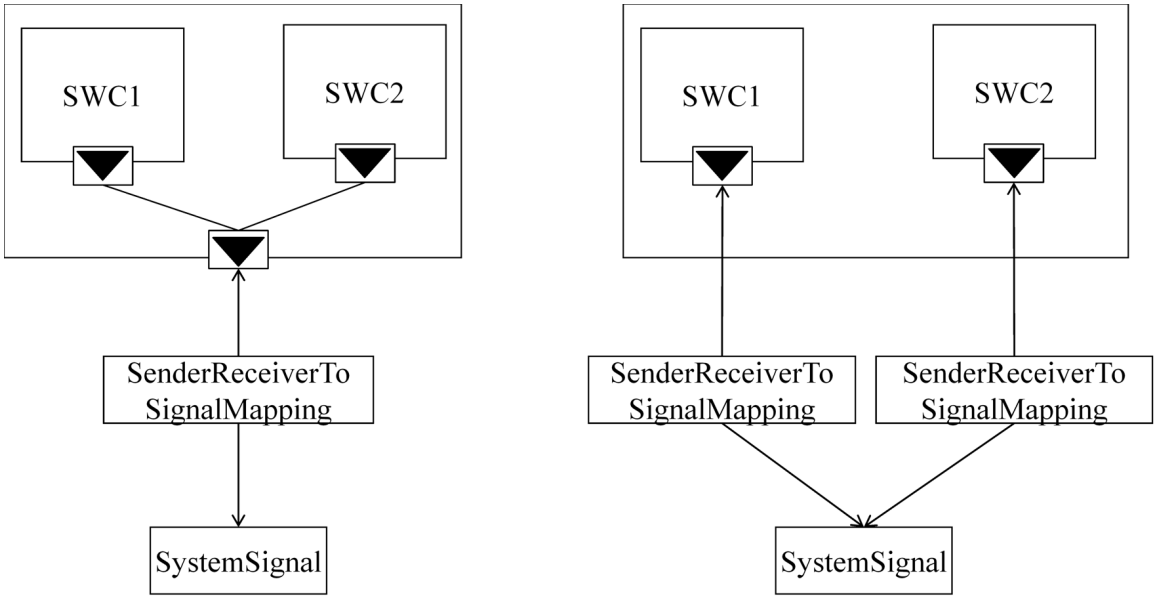

Figure 5.20: Example for data elements that are effectively mapped to the same SystemSignal

The different kinds of data mapping are described in the following sections in detail.

Please note that the usage of `ImplementationDataType`s within an `AnyInstanceRef` is described in detail in [1].

5.2.1 Mapping of Variable Data Prototypes on System Signals

This section describes how `VariableDataPrototype`s are mapped onto `SystemSignal`s. For a detailed description of the interconnection of software components refer to [4].

It is the task of system configuration to map `VariableDataPrototype`, `ClientServerOperation`, or `Trigger` contained in `PortPrototype`s referenced by the `SwConnector` onto a `SystemSignal`.

[TPS_SYST_01033] `DataMapping` and `SwConnector` ⌈For the purpose of creating `DataMapping`s `PortPrototype`s may or may not be connected by `SwConnector`s.⌋

The same `SystemSignal` may satisfy more than one `SwConnector` (1:n communication), and one `SwConnector` may be implemented by several `SystemSignal`s

(e.g. one per VariableDataPrototype in the PortInterface being connected), so there is no 1:1 mapping between SwConnectors and SystemSignals.

In the following sections, each reference to a VariableDataPrototype, ArgumentDataPrototype, or Trigger is of type AtpInstanceRef [1]. This means it not only references the actual VariableDataPrototype, but additionally contains contextual references to the PortPrototype and the hierarchy of SwComponentPrototypes forming the individual instance context of the VariableDataPrototype.

In a complete System with category SYSTEM_DESCRIPTION, it is sufficient to refer to the VariableDataPrototype in the PPortPrototype or the RPortPrototype to define the mapping of the communication between a provider and its receivers.

This is possible since the connectors implicitly define which RPortPrototype are connected to which PPortPrototype. In case the System with category SYSTEM_DESCRIPTION does not use a complete Software Component Description (VFB View) the data mapping needs not to be defined. This supports the inclusion of legacy signals.

[TPS_SYST_01137] Several DataMappings may be defined for the same SystemSignal ⌈For a SystemSignal which is

- part in several SystemSignalGroups
- part in at least one SystemSignalGroup and at the same time is transmitted additionally as standalone SystemSignal
- used in N:1 sender-receiver communication where the same SystemSignal is mapped to compatible VariableDataPrototypes with DataMappings on different SwComponentPrototypes. Please note that in this case [constr_9330] applies.
- used in a ClientServer communication where the same SystemSignal is used for a compatible ClientServerOperation in DataMappings on the different clients

several DataMappings may be defined.⌋

[constr_9330] Derivation of network representation in case that several DataMappings are defined that map the same SystemSignal to different VariableDataPrototypes

Imposition time: CP: IT_SysDesc

⌈If several DataMappings are defined that map the same SystemSignal to different VariableDataPrototypes then

- all ISignals that reference this SystemSignal shall define networkRepresentationProps or

- if networkRepresentationProps are not specified on the ISignal level (and are therefore derived from the ImplementationDataType) then the different DataMappings shall reference VariableDataPrototypes that in turn reference the identical ImplementationDataType.

⌋

Please note that [constr_9330] was introduced to forbid cases in which a serializing transformer will get contradicting requirements regarding the network representation. As an alternative the scenario sketched on the left side in Figure 5.20 may be used in which only one data mapping on the composition is defined.

As the SystemSignal represents a specific information this information may appear in different delivery units, but it is still the same information.

One example could be the tire pressure sensor value. At first each wheel's tire pressure value exists alone, as they are captured at each wheel individually. So there are 4 SystemSignals defined for a 4 wheel vehicle. If now some Software Component receives all 4 tire pressure values and puts them inside a SystemSignalGroup for consistent handling, then those 4 tire pressure values are still represented by the same 4 SystemSignals. Thus each individual tire pressure SystemSignal may appear as stand-alone as well as part of a SystemSignalGroup.

[TPS_SYST_01050] SystemSignal in the System Extract and ECU Extract ⌈In the System with category SYSTEM_EXTRACT or ECU_EXTRACT the missing DataMappings on the complementary Sender/Receiver side needs to be supplemented.⌋

In the System with category SYSTEM_EXTRACT or ECU_EXTRACT, where only the relevant parts of the rootSoftwareComposition are defined, it is necessary to utilize the information from the complementary PortPrototype if the corresponding PortPrototype is located on another ECU and thus is not part of the extract. This is described in more detail in chapter 13.2 and chapter 14.2.3.

Therefore in a System with category ECU_EXTRACT the DataMappings are provided on both, PPortPrototypes and RPortPrototypes.

[TPS_SYST_01034] Data Mappings can be applied to compositions and atomic software components ⌈DataMappings can be applied to CompositionSwComponentTypes and on AtomicSwComponentTypes in a System of any category other than ECU_EXTRACT.⌋

Please note that for a System of category ECU_EXTRACT [constr_5328] applies.

[TPS_SYST_01035] Transformation of Data Mappings during flattening ⌈During the creation of the System with category ECU_EXTRACT (flattening) each existing DataMapping that is referencing a PortPrototype of a SwComponentType that is not directly referenced by the RootSwCompositionPrototype in

the role softwareComposition shall be transformed to a DataMapping that directly refers by means of a DataPrototypeInSystemInstanceRef to a PortPrototype of the CompositionSwComponentType that is referenced by the RootSwCompositionPrototype in the role softwareComposition.⌋

In other words in the System with category ECU_EXTRACT only DataMappings defined on the outerPorts of the RootSwCompositionPrototype are allowed to exist.

[TPS_SYST_01036] No additional Data Mappings in composition substructure ⌈When a CompositionSwComponentType is refined by a supplier the already existing DataMappings that refer to the CompositionSwComponentType shall not be copied to the internal substructure.⌋

Suppliers who add substructure to a CompositionSwComponentType by adding SwComponentPrototypes and SwConnectors shall respect the predefined DataMappings on the CompositionSwComponentType.

The OEM/Supplier Collaboration Scenario is described in chapter 13.1.

[TPS_SYST_05034] DataMapping of ImplementationDataType of category UNION, DATA_REFERENCE, or FUNCTION_REFERENCE ⌈SenderReceiverInterface.dataElement that is typed by an ImplementationDataType of category UNION, DATA_REFERENCE, or FUNCTION_REFERENCE is not allowed to be mapped by a DataMapping.⌋

Please note that for unions there is [constr_1441] and [constr_1607] that are restricting such a modeling.

5.2.1.1 Mapping of Variable Data Prototypes with primitive datatypes on System Signals (Sender-Receiver Communication)

This section describes the relation between the VariableDataPrototype with primitive datatypes and the SystemSignal (see Figure 5.21).

[TPS_SYST_02082] SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE or BOOLEAN and a DataTypeMap exists ⌈If a SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE or BOOLEAN and a DataTypeMap exists that points to the ApplicationPrimitiveDataType and an ImplementationDataType of category VALUE or TYPE_REFERENCE that eventually references (via the SwDataDefProps.implementationDataType) an ImplementationDataType of category VALUE, then this SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02083] SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and a DataTypeMap exists ⌈If a SenderReceiverInterface.dataElement is typed by

an ApplicationPrimitiveDataType of category STRING and a DataTypeMap exists that points to the ApplicationPrimitiveDataType and an ImplementationDataType of category ARRAY or TYPE_REFERENCE that eventually references (via the SwDataDefProps.implementationDataType) an ImplementationDataType of category ARRAY with a subElement that either

- represents the platform type uint8 or
- references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to the value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE,

then this SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02084] SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and a DataTypeMap exists ⌈If a SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and a DataTypeMap exists that points to the ApplicationArrayDataType and an ImplementationDataType of category ARRAY or TYPE_REFERENCE that eventually references (via the SwDataDefProps.implementationDataType) an ImplementationDataType of category ARRAY with a subElement that either

- represents the platform type uint8 or
- references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to the value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE,

then this SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02085] SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category ARRAY ⌈If a SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category ARRAY with a subElement that either

- represents the platform type uint8 or
- references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to the value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE,

then this SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02086] SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category VALUE or TYPE_REFERENCE ⌈If a SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category VALUE or TYPE_REFERENCE that eventually references (via

the SwDataDefProps.implementationDataType) an ImplementationDataType of category VALUE then this SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02087] SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category BOOLEAN and no DataTypeMap exists ⌈The SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02088] SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and no DataTypeMap exists ⌈If a SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and no DataTypeMap exists and the ApplicationArrayDataType fulfills the following conditions:

- ApplicationPrimitiveDataType.swDataDefProps.dataConstr exists and refers to a PhysConstrs.
- ApplicationPrimitiveDataType.swDataDefProps.compuMethod exists and refers to a CompuMethod of category TEXTTABLE and CompuMethod.compuPhysToInternal exists.
- Application of ApplicationPrimitiveDataType.swDataDefProps.compuMethod to ApplicationPrimitiveDataType.swDataDefProps.dataConstr yields a numerical range in [0 .. 255]

then the SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02089] SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and no DataTypeMap exists ⌈If a SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and no DataTypeMap exists and the ApplicationPrimitiveDataType fulfills the following conditions:

- ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout exists and values of SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupFrom and SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupTo are both set to 1.
- An ApplicationPrimitiveDataType.swDataDefProps.swTextProps exists and refers to an SwBaseType where the SwBaseType.baseTypeDefinition.baseTypeEncoding is set to NONE and the value of SwBaseType.baseTypeDefinition.baseTypeSize is set to 8.

then the SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_02090] SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE and no DataTypeMap

exists ⌈There is no clear indication that the SenderReceiverInterface.dataElement is a candidate for a primitive Data Mapping.⌋

[TPS_SYST_01037] primitive Data Mapping of UINT8-Arrays ⌈The primitive Data Mapping may also be used for the Data Mapping of UINT8-Arrays. This supports an optimized definition of the Data Mapping.⌋

In other words it is allowed to map an array VariableDataPrototype consisting of UINT8 elements to exactly one SystemSignal in the context of one SenderReceiverToSignalMapping. A UINT8 element may be a String or an array that contains array elements of Integer type with range 0..255.

Background: In the ECU Configuration of the AUTOSAR COM module such a SystemSignal will be mapped to a COM Signal with the ComSignalType UINT8_N.

[TPS_SYST_02279] SenderReceiverInterface.dataElement is typed by a "new-world" variable-size ApplicationArrayDataType and a DataTypeMap exists ⌈A SenderReceiverInterface.dataElement is a candidate for a primitive SenderReceiverToSignalMapping to a single SystemSignal if all following conditions are fulfilled:

- a SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType that fulfills the conditions of a "new-world" dynamic-size array data type according to [TPS_SWCT_01644] (see definition in Software Component Template [4])
- the ApplicationArrayDataType has the dynamicArraySizeProfile = VSA_LINEAR
- a DataTypeMap exists that points to both the ApplicationArrayDataType and an ImplementationDataType that fulfills the conditions of a "new-world" dynamic size array data type according to [TPS_SWCT_01645] (see definition in Software Component Template [4]) and is of category STRUCTURE
- the referenced ImplementationDataType has the dynamicArraySizeProfile = VSA_LINEAR
- the referenced ImplementationDataType has two subElements where
  - one is a numerical value that represents the size indicator and
  - the other is an ImplementationDataTypeElement of category ARRAY that in turn contains a subElement that represents the platform type uint8.

⌋

[constr_5112] `ImplementationDataType` needs to be defined if a "new-world" variable-size `ApplicationArrayDataType` is mapped to a single `SystemSignal`

Imposition time: CP: `IT_EcuExt`

⌈A `SenderReceiverInterface.dataElement` that is typed by a "new-world" variable-size `ApplicationArrayDataType` according to [TPS_SWCT_01644] (see definition in Software Component Template [4]) is only allowed to be mapped to a single `SystemSignal` by the `SenderReceiverToSignalMapping` if a `DataTypeMap` exists that points to both the `ApplicationArrayDataType` and an `ImplementationDataType` that fulfills the conditions of a "new-world" dynamic size array data type according to [TPS_SWCT_01645] (see definition in Software Component Template [4]).⌋

[TPS_SYST_02280] `SenderReceiverInterface.dataElement` is typed by a "new-world" variable-size `ImplementationDataType` ⌈A `SenderReceiverInterface.dataElement` is a candidate for a primitive `SenderReceiverToSignalMapping` to a single `SystemSignal` if all following conditions are fulfilled:

- the `SenderReceiverInterface.dataElement` is typed by an `ImplementationDataType` that fulfills the conditions of a "new-world" dynamic-size array data type according to [TPS_SWCT_01645] (see definition in Software Component Template [4]).
- the referenced `ImplementationDataType` has the `dynamicArraySizeProfile` = `VSA_LINEAR`
- the `ImplementationDataType` is of category STRUCTURE with two `subElement`s where
  - one is a numerical value that represents the size indicator and
  - the other is an `ImplementationDataTypeElement` of category `ARRAY` that in turn contains a `subElement` that represents the platform type uint8.

⌋

With [TPS_SYST_02279] and [TPS_SYST_02280] it is possible to map a `dataElement` that represents a "new-world" variable-size array to a single `SystemSignal` without the usage of a data transformer.

Please note that the mapping of an "old-world" variable-size array (see definition in Software Component Template [4]) to a single `SystemSignal` is not supported by AUTOSAR since the Rte_Send call does not include the IN parameter [length] and the Rte_Receive API does not include the OUT parameter [length] any longer that was used in former releases to pass the number of elements in the data element.

[constr_5113] Mapping of "old-world" variable size arrays to a single `SystemSignal` is not supported.

Imposition time: CP: `IT_EcuExt`

⌈The `SenderReceiverToSignalMapping` is not allowed to map a `dataElement` that is typed by an "old-world" variable size array defined by [TPS_SWCT_01641] and [TPS_SWCT_01642] (see definition in Software Component Template [4]) to a single `SystemSignal`.⌋

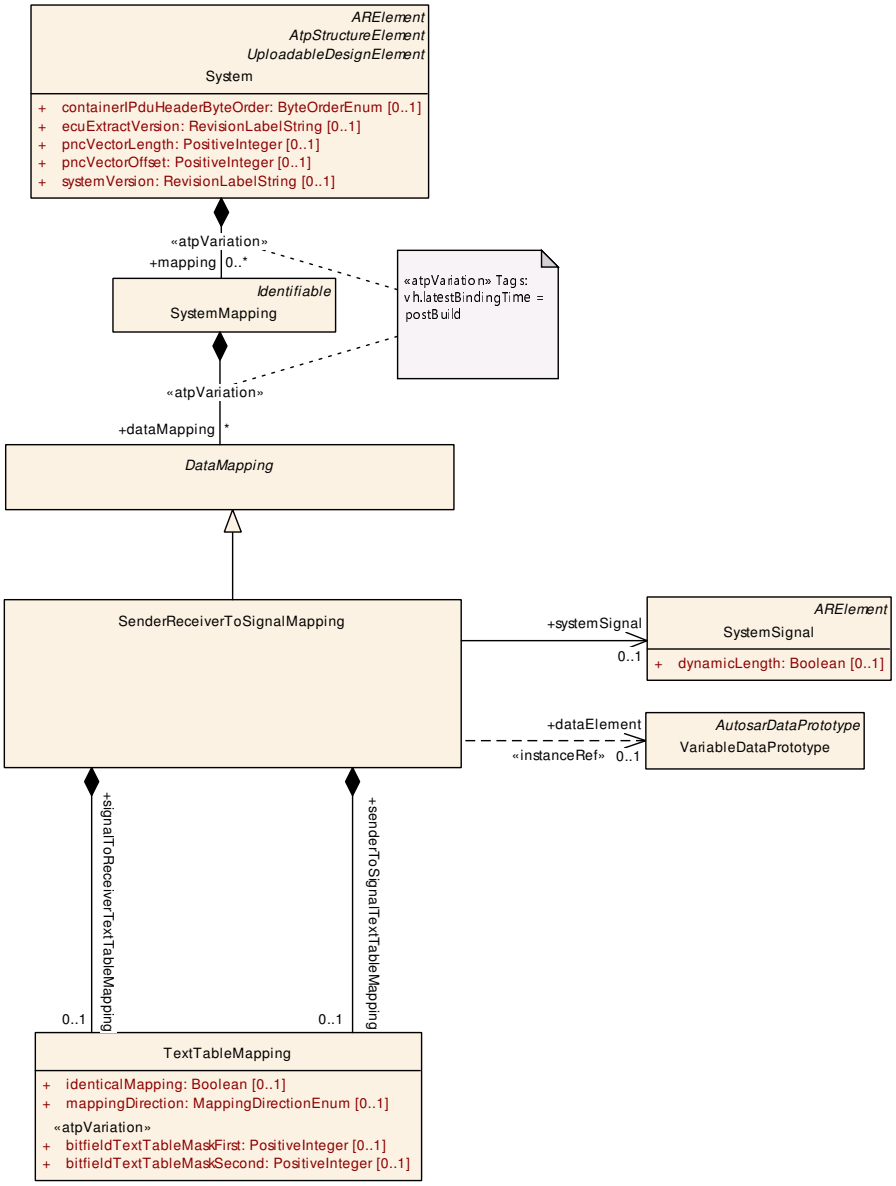

Figure 5.21: Mapping of data elements with primitive datatypes (SenderRecPrimitiveTypeMapping)

Class	**SenderReceiverToSignalMapping**			
Note	Mapping of a sender receiver communication data element to a signal.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	0..1	iref	Reference to the data element. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal. **Tags:** atp.Splitkey=senderToSignalTextTableMapping
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype. **Tags:** atp.Splitkey=signalToReceiverTextTableMapping
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the data element. **Tags:** atp.Splitkey=systemSignal

Table 5.23: SenderReceiverToSignalMapping

[constr_5466] Existence of SenderReceiverToSignalMapping.dataElement

Imposition time: CP: IT_EcuExt

⌈For each SenderReceiverToSignalMapping, the reference to VariableData-Prototype in the role dataElement shall exist.⌋

[constr_5467] Existence of SenderReceiverToSignalMapping.systemSignal

Imposition time: CP: IT_EcuExt

⌈For each SenderReceiverToSignalMapping, the reference to SystemSignal in the role systemSignal shall exist.⌋

[TPS_SYST_02304] Conversion of discrete parts of a CompuMethod on signal level in SenderReceiverToSignalMapping ⌈If a SystemSignal defines a CompuMethod of category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE, and BITFIELD_TEXTTABLE, a conversion of the texttable part of the CompuMethod of the AutosarDataType of the sending DataPrototype to the SystemSignal as well as from the SystemSignal to the CompuMethod associated with the Autosar-DataType of the receiving DataPrototype may be necessary.

For this purpose, meta-class SenderReceiverToSignalMapping aggregates the meta-class TextTableMapping in the roles senderToSignalTextTableMapping and signalToReceiverTextTableMapping.⌋

As explained in specification of the AUTOSAR Software Component Template [4], the TextTableMapping allows enumerated types to be connected when they have the

same or similar semantics but different numerical and/or symbolic representations of those semantics.

[TPS_SYST_02305] Relevance of attribute TextTableMapping.mapping-Direction in an aggregation by SenderReceiverToSignalMapping ⌈The value of attributes

- SenderReceiverToSignalMapping.senderToSignalTextTable-Mapping.mappingDirection
- SenderReceiverToSignalMapping.signalToReceiverTextTable-Mapping.mappingDirection

has no meaning and shall be ignored.⌋

Class	**TextTableMapping**			
Note	Defines the mapping of two DataPrototypes typed by AutosarDataTypes that refer to CompuMethods of category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE or BITFIELD_TEXTTABLE.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.textTableMapping, SenderRecArrayTypeMapping.senderToSignalTextTable Mapping, SenderRecArrayTypeMapping.signalToReceiverTextTableMapping, SenderReceiverToSignal Mapping.senderToSignalTextTableMapping, SenderReceiverToSignalMapping.signalToReceiverTextTable Mapping, SenderRecRecordElementMapping.senderToSignalTextTableMapping, SenderRecRecord ElementMapping.signalToReceiverTextTableMapping, SubElementMapping.textTableMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bitfieldTextTable MaskFirst	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the first element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
bitfieldTextTable MaskSecond	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the second element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
identical Mapping	Boolean	0..1	attr	If identicalMapping is set == true the values of the two referenced DataPrototypes do not need any conversion of the values.
mapping Direction	MappingDirectionEnum	0..1	attr	Specifies the conversion direction for which the TextTable Mapping is applicable.
valuePair	TextTableValuePair	*	aggr	Defines a pair of values which are translated into each other. **Tags:** atp.Splitkey=valuePair

Table 5.24: TextTableMapping

5.2.1.2 Mapping of Variable Data Prototypes with composite datatypes (Sender-Receiver Communication)

This section describes the mapping of VariableDataPrototypes typed by composite data types to SystemSignals.

It is not possible to map a VariableDataPrototype typed by composite data type directly (without any additional mechanisms) to one SystemSignal because The RTE is required to treat AUTOSAR signals transmitted using sender-receiver communication consistently. For this purpose, data transformation or SystemSignalGroups is used.

There are two ways to map a VariableDataPrototype typed by composite data type to SystemSignals/SystemSignalGroups:

1. Use data transformation and map it directly to a SystemSignal.
2. Map it to a SystemSignalGroup with SenderReceiverToSignalGroupMapping

[constr_9399] Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if no data transformation is used

Imposition time: CP: IT_EcuExt

⌈Either all or a subset of elements of a VariableDataPrototype typed by a composite data type shall be mapped with the SenderReceiverToSignalGroupMapping to SystemSignals of the same SystemSignalGroup if data transformation is not used.⌋

[constr_9400] Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if comBasedSignalGroupTransformation is used

Imposition time: CP: IT_EcuExt

⌈If a ISignalGroup has a reference to the DataTransformation element in the role comBasedSignalGroupTransformation then the ISignalGroup shall reference a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping that either maps all or a subset of elements of a VariableDataPrototype typed by a composite data type to SystemSignals of this SystemSignalGroup.⌋

[constr_9401] Mapping of VariableDataPrototype typed by a array data type consisting of UINT8 elements to exactly one SystemSignal

Imposition time: CP: IT_EcuExt

⌈The SenderReceiverToSignalGroupMapping is allowed to reference a VariableDataPrototype typed by an array data type if and only if the array data type consists of UINT8 elements.⌋

5.2.1.2.1 Data Transformation

If data transformation is used, the consistency of the composite data is assured by the transformation.

A VariableDataPrototype typed by composite data type can be mapped to one SystemSignal without any SystemSignalGroup if data transformation is used.

In that case any required mapping between the ApplicationCompositeElementDataPrototypes of the VariableDataPrototype of the connected PortPrototypes needs to be expressed by means of a PortInterfaceMapping attached to the SwConnector connecting the two PortPrototypes and not by means of two separated DataMappings (one referencing the VariableDataPrototype at the PPortPrototype and the other one referencing the VariableDataPrototype at the RPortPrototype).

During creation of a System Extract of the System Configuration Description or the creation of an ECU Extract of the System Configuration Description, this PortInterfaceMapping needs to be preserved in order to support proper deserializing transformation at the receiver side (see chapter 13.4 and 14.2).

See chapter 7 for details how to enable data transformation.

In case the COM Based Transformer [17] is used the mapping from section 7.3.3 is required.

5.2.1.2.2 Mapping via SystemSignalGroups

The VariableDataPrototype that is referenced by dataElement can be typed by an ApplicationDataType or by an ImplementationDataType. This type decides which reference is used within the SenderRecRecordElementMapping and SenderRecArrayElementMapping.

Composite DataPrototypes may nest within composite VariableDataPrototypes. Each element typed by a primitive data type of such nested composite VariableDataPrototype can be mapped to one SystemSignal of a SystemSignalGroup.

Please note that not every single element typed by a primitive data type needs to be mapped to a SystemSignal since a partial mapping is also supported that maps a subset of elements of the composite data type to separate SystemSignals of a SystemSignalGroup.

The mapping between the SystemSignal and the VariableDataPrototype is provided in the SenderReceiverToSignalGroupMapping (see Figure 5.22).

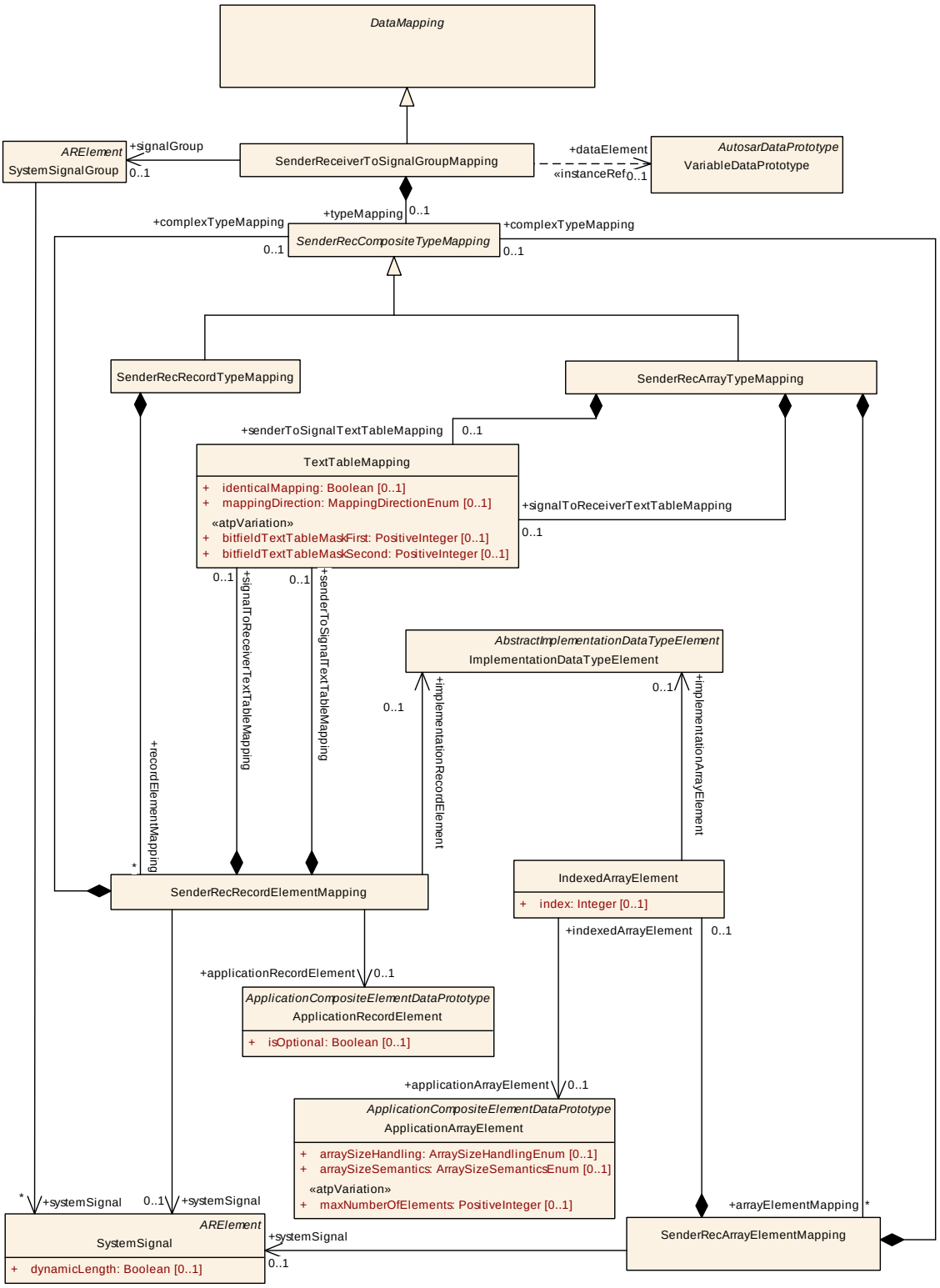

Figure 5.22: Mapping of data elements with composite data types (SenderRecCompositeTypeMapping)

[constr_3000] valid SenderRecCompositeTypeMappings

Imposition time: CP: IT_EcuExt

⌈All SenderRecRecordElementMappings or SenderRecArrayElement-Mappings aggregated in the context of a given SenderReceiverToSignal-GroupMapping shall reference a SystemSignal that is also referenced in the role systemSignal by the SystemSignalGroup that is referenced by the enclosing SenderReceiverToSignalGroupMapping in the role signalGroup.⌋

In other words: within the context of an SenderReceiverToSignalGroupMapping, it shall only be possible to refer to a SystemSignal that is a member of the System-SignalGroup referenced by the SenderReceiverToSignalGroupMapping.

Please note that [constr_3000] does not demand that all leaf elements of the composite data type are actually mapped to a SystemSignal.

[TPS_SYST_02278] Existence of SystemSignals in a SystemSignalGroup that are not referenced by a SenderRecCompositeTypeMapping ⌈There are use cases where not all SystemSignals of a SystemSignalGroup are referenced by a SenderRecRecordElementMapping or a SenderRecArrayElementMapping. One example is the ComBased Transformer use case where the SystemSignal-Group contains SystemSignals that are added by additional Transformers like the E2E Transformer (e.g. CRC and Alive Counter), but only the application data element signals are mapped by the SenderReceiverToSignalGroupMapping. One additional use case is the partial mapping of composite data types where only a subset of elements of the composite data type are mapped to SystemSignals of the System-SignalGroup.⌋

Class	**SenderReceiverToSignalGroupMapping**			
Note	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	0..1	iref	Reference to a data element with a composite datatype which is mapped to a signal group. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
signalGroup	SystemSignalGroup	0..1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type **Tags:** atp.Splitkey=signalGroup
typeMapping	SenderRecComposite TypeMapping	0..1	aggr	The CompositeTypeMapping maps the ApplicationArray Elements and ApplicationRecordElements to Signals of the SignalGroup. **Tags:** atp.Splitkey=typeMapping

Table 5.25: SenderReceiverToSignalGroupMapping

[constr_5468] Existence of SenderReceiverToSignalGroupMapping.dataElement

Imposition time: CP: IT_EcuExt

⌈For each SenderReceiverToSignalGroupMapping, the reference to VariableDataPrototype in the role dataElement shall exist.⌋

[constr_5469] Existence of SenderReceiverToSignalGroupMapping.signalGroup

Imposition time: CP: IT_EcuExt

⌈For each SenderReceiverToSignalGroupMapping, the reference to SystemSignalGroup in the role signalGroup shall exist.⌋

[constr_5470] Existence of SenderReceiverToSignalGroupMapping.typeMapping

Imposition time: CP: IT_EcuExt

⌈For each SenderReceiverToSignalGroupMapping, the aggregation of SenderRecCompositeTypeMapping in the role typeMapping shall exist.⌋

Class	***SenderRecCompositeTypeMapping*** (abstract)			
Note	Two mappings exist for the composite data types: "ArrayTypeMapping" and "RecordTypeMapping". In both, a primitive datatype will be mapped to a system signal. But it is also possible to combine the arrays and the records, so that an "array" could be an element of a "record" and in the same manner a "record" could be an element of an "array". Nesting these data types is also possible. If an element of a composite data type is again a composite one, the "CompositeTypeMapping" element will be used one more time (aggregation between the ArrayElementMapping and CompositeType Mapping or aggregation between the RecordElementMapping and CompositeTypeMapping).			
Base	*ARObject*			
Subclasses	SenderRecArrayTypeMapping, SenderRecRecordTypeMapping			
Aggregated by	SenderRecArrayElementMapping.complexTypeMapping, SenderReceiverToSignalGroupMapping.typeMapping, SenderRecRecordElementMapping.complexTypeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 5.26: SenderRecCompositeTypeMapping

Class	**SenderRecArrayTypeMapping**			
Note	If the ApplicationCompositeDataType is an Array, the "ArrayTypeMapping" will be used.			
Base	*ARObject*, *SenderRecCompositeTypeMapping*			
Aggregated by	SenderRecArrayElementMapping.complexTypeMapping, SenderReceiverToSignalGroupMapping.typeMapping, SenderRecRecordElementMapping.complexTypeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arrayElement Mapping	SenderRecArray ElementMapping	*	aggr	Each ApplicationArrayElement shall be mapped on a SystemSignal. **Tags:** atp.Splitkey=arrayElementMapping

▽

△

Class	**SenderRecArrayTypeMapping**			
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal. **Tags:** atp.Splitkey=senderToSignalTextTableMapping
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype. **Tags:** atp.Splitkey=signalToReceiverTextTableMapping

Table 5.27: SenderRecArrayTypeMapping

Class	**SenderRecRecordTypeMapping**			
Note	If the ApplicationCompositeDataType is a Record, the "RecordTypeMapping" will be used.			
Base	*ARObject*, *SenderRecCompositeTypeMapping*			
Aggregated by	SenderRecArrayElementMapping.complexTypeMapping, SenderReceiverToSignalGroupMapping.typeMapping, SenderRecRecordElementMapping.complexTypeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
recordElement Mapping	SenderRecRecord ElementMapping	*	aggr	Each ApplicationRecordElement shall be mapped on a SystemSignal. **Tags:** atp.Splitkey=recordElementMapping

Table 5.28: SenderRecRecordTypeMapping

Class	**SenderRecRecordElementMapping**			
Note	Mapping of a primitive record element to a SystemSignal. If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference applicationRecordElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference implementationRecordElement shall be used. Either the implementationRecordElement or applicationRecordElement reference shall be used. If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the RecordElementMapping element will aggregate the complexTypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals.			
Base	*ARObject*			
Aggregated by	SenderRecRecordTypeMapping.recordElementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application RecordElement	ApplicationRecord Element	0..1	ref	Reference to an ApplicationRecordElement in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=applicationRecordElement
complexType Mapping	SenderRecComposite TypeMapping	0..1	aggr	This aggregation will be used if the element is composite. **Tags:** atp.Splitkey=complexTypeMapping
implementation RecordElement	ImplementationData TypeElement	0..1	ref	Reference to an ImplementationRecordElement in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=implementationRecordElement

▽

△

Class	SenderRecRecordElementMapping			
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal. **Tags:** atp.Splitkey=senderToSignalTextTableMapping
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype. **Tags:** atp.Splitkey=signalToReceiverTextTableMapping
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement. **Tags:** atp.Splitkey=systemSignal

Table 5.29: SenderRecRecordElementMapping

[constr_3230] Usage of SenderRecRecordElementMapping.applicationRecordElement

Imposition time: CP: IT_EcuExt

⌈SenderRecRecordElementMapping.applicationRecordElement shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement) is typed by an ApplicationDataType.⌋

[constr_3244] Usage of SenderRecRecordElementMapping.implementationRecordElement

Imposition time: CP: IT_EcuExt

⌈SenderRecRecordElementMapping.implementationRecordElement shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement) is typed by an ImplementationDataType.⌋

Class	SenderRecArrayElementMapping			
Note	The SenderRecArrayElement may be a primitive one or a composite one. If the element is primitive, it will be mapped to the SystemSignal (multiplicity 1). If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference to the ApplicationArrayElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference to the ImplementationArrayElement shall be used. If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the ArrayElementMapping element will aggregate the TypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals. Regardless whether composite or primitive array element is mapped the indexed element always needs to be specified.			
Base	*ARObject*			
Aggregated by	SenderRecArrayTypeMapping.arrayElementMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**SenderRecArrayElementMapping**			
complexType Mapping	SenderRecComposite TypeMapping	0..1	aggr	This aggregation will be used if the element is composite. **Tags:** atp.Splitkey=complexTypeMapping
indexedArray Element	IndexedArrayElement	0..1	aggr	Reference to an indexed array element in the context of the dataElement or in the context of a composite element. **Tags:** atp.Splitkey=indexedArrayElement
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationArrayElement. **Tags:** atp.Splitkey=systemSignal

Table 5.30: SenderRecArrayElementMapping

Class	**IndexedArrayElement**			
Note	This element represents exactly one indexed element in the array. Either the applicationArrayElement or implementationArrayElement reference shall be used.			
Base	*ARObject*			
Aggregated by	SenderRecArrayElementMapping.indexedArrayElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationArray Element	ApplicationArray Element	0..1	ref	Reference to an ApplicationArrayElement in an array. **Tags:** atp.Splitkey=applicationArrayElement
implementation ArrayElement	ImplementationData TypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array. **Tags:** atp.Splitkey=implementationArrayElement
index	Integer	0..1	attr	Position of an element in an array. Starting position is 0.

Table 5.31: IndexedArrayElement

[constr_5471] Existence of SenderRecArrayElementMapping.indexedArrayElement

Imposition time: CP: IT_EcuExt

⌈For each SenderRecArrayElementMapping, the aggregation in the role indexedArrayElement shall exist.⌋

[constr_5472] Existence of IndexedArrayElement.index

Imposition time: CP: IT_EcuExt

⌈For each IndexedArrayElement, the attribute index shall exist.⌋

[constr_3231] Usage of IndexedArrayElement.applicationArrayElement

Imposition time: CP: IT_EcuExt

⌈IndexedArrayElement.applicationArrayElement shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement) is typed by an ApplicationDataType.⌋

[constr_3245] Usage of IndexedArrayElement.implementationArrayElement

Imposition time: CP: IT_EcuExt

⌈IndexedArrayElement.implementationArrayElement shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement) is typed by an ImplementationDataType.⌋

Figure 5.23 shows a mapping example for nested composite data types.

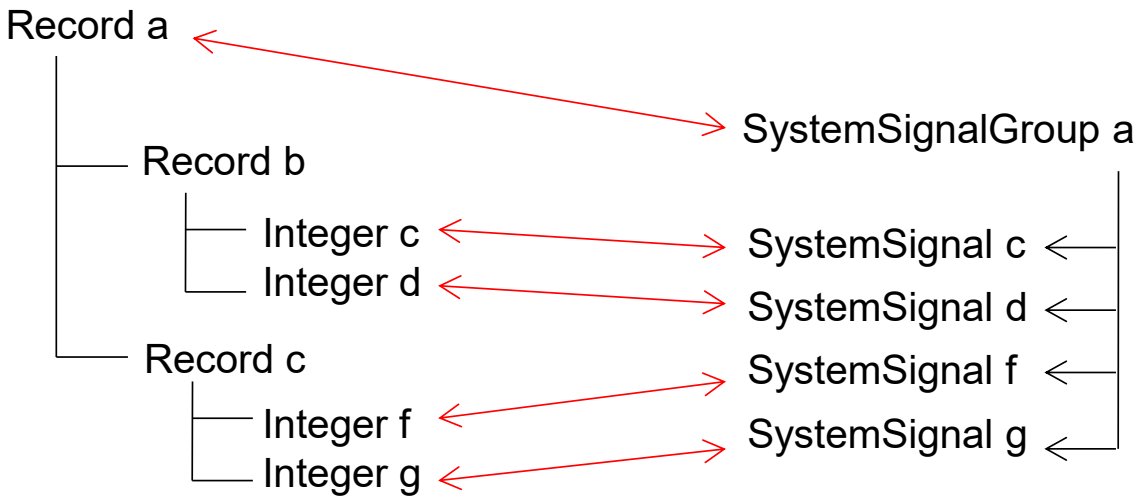

Figure 5.23: Mapping example for nested composite data types

Record a is mapped with SenderReceiverToSignalGroupMapping to a SystemSignalGroup. The content of Record a is mapped with the SenderRecRecordTypeMapping. Since the first element of *Record a* is *Record b* the SenderRecRecordElementMapping does not contain a reference to a SystemSignal because signals apply only to atomic data items. Instead it contains a complexTypeMapping with two SenderRecRecordElementMappings for *Integer c* and *Integer d*. These two elements are mapped to SystemSignals.

Please note that a partial mapping of a data element typed by composite data type in a PPortPrototype is also supported. If a VariableDataPrototype with a composite data type in a PPortPrototype is mapped to a SystemSignalGroup then it is allowed to map only a subset of elements of the composite data type that are primitives to separate SystemSignals of the SystemSignalGroup. This means that it is possible to transmit a subset of a composite data element in a ISignalGroup over the network. Figure 5.24 shows a partial mapping example for nested composite data types.

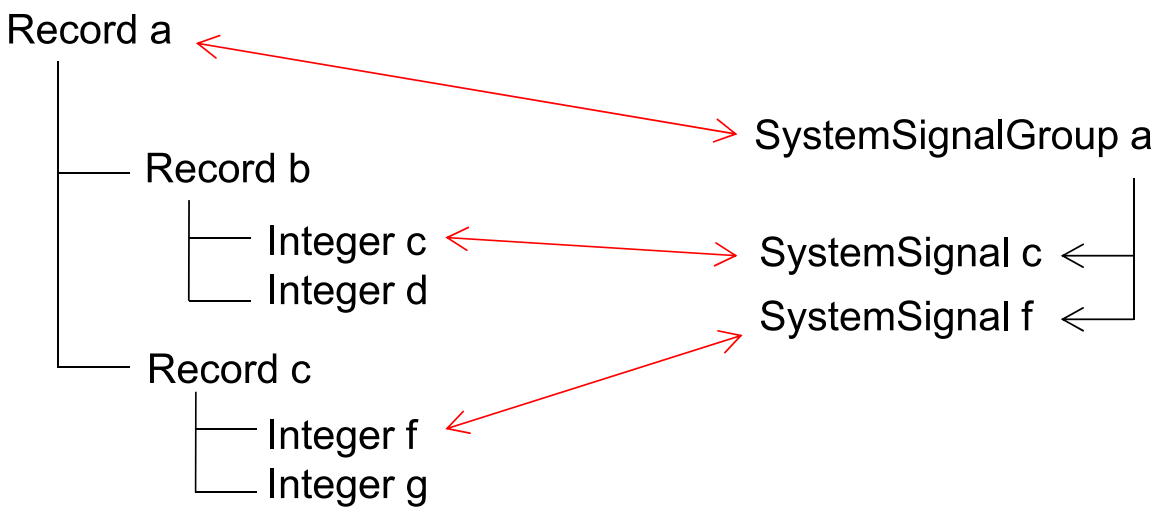

Figure 5.24: Partial mapping example for nested composite data types

[TPS_SYST_02306] Conversion of discrete parts of a CompuMethod on signal level in `SenderRecRecordElementMapping` and `SenderRecArrayTypeMapping` ⌈If a `SystemSignal` defines a `CompuMethod` of category `TEXTTABLE`, `SCALE_LINEAR_AND_TEXTTABLE`, and `BITFIELD_TEXTTABLE`, a conversion of the texttable part of the `CompuMethod` of the `AutosarDataType` of the sending `DataPrototype` to the `SystemSignal` as well as from the `SystemSignal` to the `CompuMethod` associated with the `AutosarDataType` of the receiving `DataPrototype` may be necessary.

For this purpose,

- meta-class `SenderRecRecordElementMapping` aggregates the meta-class `TextTableMapping` in the roles `SenderRecRecordElementMapping.senderToSignalTextTableMapping` and `SenderRecRecordElementMapping.signalToReceiverTextTableMapping`.
- meta-class `SenderRecArrayTypeMapping` aggregates the meta-class `TextTableMapping` in the roles `SenderRecArrayTypeMapping.senderToSignalTextTableMapping` and `SenderRecArrayTypeMapping.signalToReceiverTextTableMapping`.

⌋

As explained in specification of the AUTOSAR Software Component Template [4], the `TextTableMapping` allows enumerated types to be connected when they have the same or similar semantics but different numerical and/or symbolic representations of those semantics.

[TPS_SYST_02307] Relevance of attribute `TextTableMapping.mappingDirection` in an aggregation by `SenderRecRecordElementMapping` or `SenderRecArrayTypeMapping` ⌈The value of attributes

- SenderRecRecordElementMapping.senderToSignalTextTable-Mapping.mappingDirection
- SenderRecRecordElementMapping.signalToReceiverTextTable-Mapping.mappingDirection
- SenderRecArrayTypeMapping.senderToSignalTextTable-Mapping.mappingDirection
- SenderRecArrayTypeMapping.signalToReceiverTextTable-Mapping.mappingDirection

has no meaning and shall be ignored.⌋

[constr_5162] Valid TextTableMapping in the context of SenderRecRecord-ElementMapping

Imposition time: CP: IT_EcuExt

⌈The aggregation of a TextTableMapping at SenderRecRecordElement-Mapping is only valid if the SenderRecRecordElementMapping also references a SystemSignal in the role systemSignal.⌋

Rationale: SenderRecRecordElementMapping could also be used on record elements that itself need to be broken down further. In other words if the SenderRec-RecordElementMapping aggregates a complexTypeMapping it shall not aggregate a TextTableMapping.

[TPS_SYST_02308] TextTableMapping defined in the context of SenderRec-ArrayTypeMapping ⌈The aggregation of a TextTableMapping at SenderRec-ArrayTypeMapping allows for the text-table translation between all array elements of an array data type that is used in a PortPrototype of the application software and the physicalProps defined for the mapped SystemSignals.⌋

Please note that the TextTableMapping is aggregated by the SenderRecArray-TypeMapping because the same mapping rule is valid for all array elements.

The following example shows a case where an ApplicationRecordDataType is defined that contains two ApplicationRecordElements. Each Application-RecordElement defines a TEXTTABLE CompuMethod. One of the SystemSignals to which one of the ApplicationRecordElements is mapped has a different CompuMethod defined and therefore a TextTableMapping is created that defines the conversion of the texttable part of the CompuMethod of the AutosarDataType of the sending DataPrototype and the CompuMethod of the SystemSignal.

```
<SENDER-REC-RECORD-TYPE-MAPPING>
  <RECORD-ELEMENT-MAPPINGS>
    <SENDER-REC-RECORD-ELEMENT-MAPPING>
      <APPLICATION-RECORD-ELEMENT-REF DEST="APPLICATION-RECORD-ELEMENT">/
          Systems/Types/R1/enum1</APPLICATION-RECORD-ELEMENT-REF>
        <SENDER-TO-SIGNAL-TEXT-TABLE-MAPPING>
          <VALUE-PAIRS>
            <TEXT-TABLE-VALUE-PAIR>
```

```
            <FIRST-VALUE>0</FIRST-VALUE>
            <SECOND-VALUE>0</SECOND-VALUE>
          </TEXT-TABLE-VALUE-PAIR>
          <TEXT-TABLE-VALUE-PAIR>
            <FIRST-VALUE>1</FIRST-VALUE>
            <SECOND-VALUE>1</SECOND-VALUE>
          </TEXT-TABLE-VALUE-PAIR>
          <TEXT-TABLE-VALUE-PAIR>
            <FIRST-VALUE>2</FIRST-VALUE>
            <SECOND-VALUE>1</SECOND-VALUE>
          </TEXT-TABLE-VALUE-PAIR>
        </VALUE-PAIRS>
      </SENDER-TO-SIGNAL-TEXT-TABLE-MAPPING>
      <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/Systems/SystemSignals/
          sig_a</SYSTEM-SIGNAL-REF>
    </SENDER-REC-RECORD-ELEMENT-MAPPING>
    <SENDER-REC-RECORD-ELEMENT-MAPPING>
      <APPLICATION-RECORD-ELEMENT-REF DEST="APPLICATION-RECORD-ELEMENT">/
          Systems/Types/R1/enum2</APPLICATION-RECORD-ELEMENT-REF>
      <SENDER-TO-SIGNAL-TEXT-TABLE-MAPPING>
        <IDENTICAL-MAPPING>true</IDENTICAL-MAPPING>
      </SENDER-TO-SIGNAL-TEXT-TABLE-MAPPING>
      <SYSTEM-SIGNAL-REF DEST="SYSTEM-SIGNAL">/Systems/SystemSignals/sig_b<
          /SYSTEM-SIGNAL-REF>
    </SENDER-REC-RECORD-ELEMENT-MAPPING>
  </RECORD-ELEMENT-MAPPINGS>
</SENDER-REC-RECORD-TYPE-MAPPING>
```

Listing 5.1: Example for the definition of a `TextTableMapping` aggregated by the `SenderRecRecordElementMapping`

5.2.1.3 Mapping of Client Server Operations to System Signals

This section describes the mapping of `ClientServerOperation`s to `SystemSignal`s (see Figure 5.25).

[TPS_SYST_01148] Mapping of IN and INOUT `ArgumentDataPrototype`s to `callSignal`s ⌈The `ArgumentDataPrototype`s that are passed to the operation (i.e. the `direction` is “in”) and the `ArgumentDataPrototype`s that are passed to and returned from the operation (i.e. the `direction` is “inout”) are expected to be mapped to the `callSignal` by the serializer.⌋

[TPS_SYST_01149] Mapping of OUT and INOUT `ArgumentDataPrototype`s to `returnSignal`s ⌈The `ArgumentDataPrototype`s that are returned from the operation (i.e. the `direction` is “out”) and the `ArgumentDataPrototype`s that are passed to and returned from the operation (i.e. the `direction` is “inout”) are expected to be mapped to the `returnSignal` by the serializer.⌋

Please note that due to `DataMapping` restrictions the client-server communication is only supported in AUTOSAR if the SOME/IP Transformer or DDS Transformer is used as serializer.

In SOME/IP the ApplicationErrors are part of the SOME/IP Header as Return Code as described by [PRS_SOMEIP_00030] in [18]. This is the reason why the ApplicationErrors are not considered in the ClientServerToSignalMapping.

In DDS, the ApplicationError is transmitted over the network through the error_code field, as specified in [FO_PRS_DDS_00303] in [19], and is mapped to the returnValue field of the DDS Transformer Infrastructure Header (see [CP_SWS_DdsXf_00751] in [20], and [CP_SWS_Dds_00930], [CP_SWS_Dds_00931] in [21]).

[TPS_SYST_01150] Mapping of returnSignal and callSignal to COM Signal
⌈In the ECU Configuration of the AUTOSAR COM module the returnSignal and the callSignal are expected to be mapped to COM Signals with the ComSignalType UINT8_N or UINT8_DYN.⌋

The ClientServerToSignalMapping can only map transformed data to SystemSignals because it contains no information how data shall be serialized, it only references the primitive SystemSignal which shall contain the serialized data. How to define the necessary information which serialization algorithm shall be applied can be found in chapter 7. The implementation of this algorithm is provided via a BSW module.

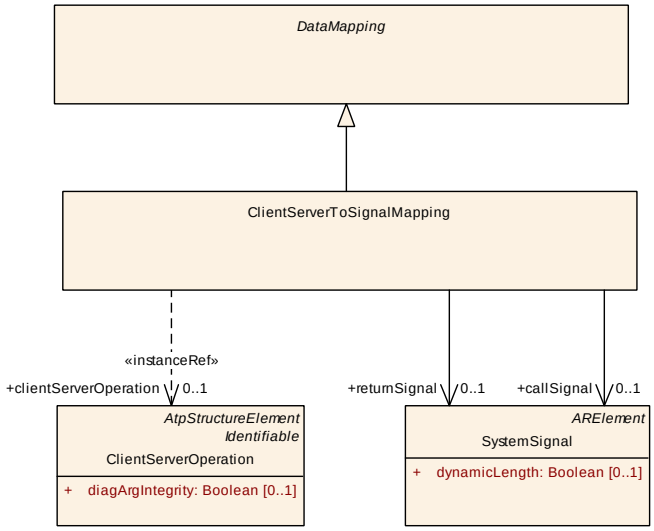

Figure 5.25: Mapping of a ClientServerOperation to a callSignal and a returnSignal

Class	**ClientServerToSignalMapping**			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
callSignal	SystemSignal	0..1	ref	Reference to the callSignal to which the IN and INOUT ArgumentDataPrototypes are mapped. **Tags:** atp.Splitkey=callSignal

△

Class	**ClientServerToSignalMapping**			
clientServer Operation	ClientServerOperation	0..1	iref	Reference to a ClientServerOperation, which is mapped to a call SystemSignal and a return SystemSignal. **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped. **Tags:** atp.Splitkey=returnSignal

Table 5.32: ClientServerToSignalMapping

[constr_5473] Existence of ClientServerToSignalMapping.callSignal

Imposition time: CP: IT_EcuExt

⌈For each ClientServerToSignalMapping, the reference to SystemSignal in the role callSignal shall exist.⌋

[constr_5474] Existence of ClientServerToSignalMapping.clientServer-Operation

Imposition time: CP: IT_EcuExt

⌈For each ClientServerToSignalMapping, the reference to ClientServer-Operation in the role clientServerOperation shall exist.⌋

[constr_3111] returnSignal in ClientServerToSignalMapping is mandatory

Imposition time: CP: IT_EcuExt

⌈A ClientServerToSignalMapping shall always have a returnSignal defined.⌋

[constr_3215] TransformationTechnology.version and Transformation-Technology.protocol settings for request and response of a client/server communication on the same TransformationTechnology

Imposition time: CP: IT_SysDesc

⌈TransformationTechnology.version and Transformation-Technology.protocol shall be identical for ISignals that are derived from the same ClientServerOperation and TransformationTechnology. This means that all ISignals that refer to ClientServerToSignalMapping.call-Signal or to ClientServerToSignalMapping.returnSignal of the same ClientServerToSignalMapping and TransformationTechnology shall have the same TransformationTechnology.protocol and Transformation-Technology.version defined.⌋

The ClientServerToSignalMapping (as any other DataMapping) defines the mapping on the level of SystemSignals (see also section 5.2). For the communication on actual PhysicalChannels (could be actual VLAN or dedicated EthernetCluster) an ISignal and a corresponding ISignalTriggering needs to be defined (see section 6.1).

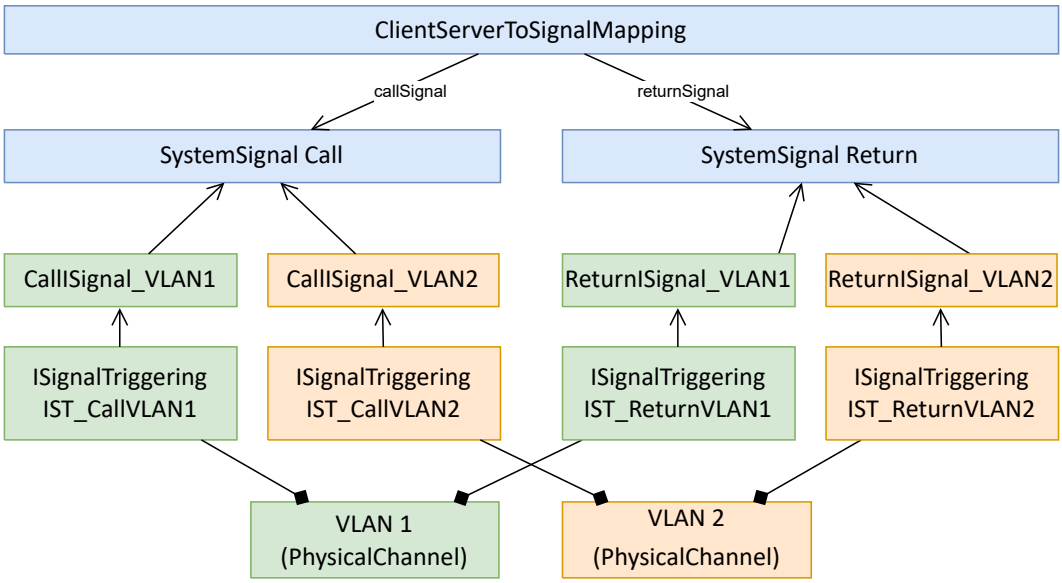

Figure 5.26: Scenario in which a SOME/IP Service with a ClientServerOperation is offered on several VLANs

In the example in figure 5.26 a server offers his service on two VLANs (could be actual VLAN or dedicated EthernetCluster). When a call arrives at the server from VLAN1 it will be propagated in the "CallISignal_VLAN1" to the RTE. The RTE takes the call message and de-serializes the payload and calls the Server. The result of the server is serialized and needs to be put into a return ISignal. But from just looking at the available ISignals for the return it is not possible to determine whether the result message has to be put into the result ISignal "ReturnISignal_VLAN1" or "ReturnISignal_VLAN2".

Thus the RTE needs to determine to which PhysicalChannel the call ISignal belongs and choose the return ISignal which is defined on the same PhysicalChannel. The ISignalTriggering defines on which PhysicalChannel an ISignal is transported, thus by determining which ISignalTriggering refers to the ISignal (in the call example ISignal: "CallISignal_VLAN1", ISignalTriggering: "IST_CallVLAN1") it is clear that the PhysicalChannel "VLAN 1" is the source of the call and thus the return message needs to go to an ISignal which is transported on that PhysicalChannel.

If a Server offers a service on several PhysicalChannels then there are some constraints to be respected in order for the RTE and Com to work together properly:

- Define an own ISignal-Pair (call and return) for each PhysicalChannel the service shall be offered on. These ISignals refer to the respective call and return SystemSignals (in case of SOME/IP TP segmenter however a concurrent processing might be defined as specified in [TPS_SYST_02442] which leads to several ISignal-Pairs for a single PhysicalChannel) being defined.
- Only one ISignalTriggering per ISignal: each ISignal shall only be referenced by up to one ISignalTriggering.
- Whether an ISignalTriggering is relevant for a specific RTE, Com, or Large Data Path in COM is determined by the ISignalTriggering referring to an ISignalPort, and that ISignalPort is member of a CommunicationConnector which belongs to the respective EcuInstance.

[constr_5273] One ISignalTriggering pair allowed per EthernetPhysicalChannel for a ClientServerOperation

Imposition time: CP: IT_SysDesc

⌈For each EthernetPhysicalChannel at most one pair of

- ISignalTriggering that refers to an ISignal that in turn refers to a SystemSignal that is referenced by a specific ClientServerToSignalMapping in the role callSignal
- ISignalTriggering that refers to an ISignal that in turn refers to a SystemSignal that is referenced by the same ClientServerToSignalMapping in the role returnSignal

shall exist, if the respective ISignalTriggerings are not referenced by a SomeipTpConnection in the role tpConcurrentProcessingSdu⌋

A concurrent processing of SOME/IP TP segmented IPdus may be defined using the SomeipTpConnection, see also [TPS_SYST_02442]. In this case there may be more than one ISignal-Pair (call and return) for a dedicated PhysicalChannel. This is illustrated in figure 5.27. This is used to support concurrent processing of server operation calls from clients that are located on different ECUs and the request of the clients are segmented by SOME/IP TP. The number of available ISignal-Pairs (call and return) on the same PhysicalChannel that reference ISignals that are in turn reference SystemSignals which are referenced by the same ClientServerToSignalMapping reflect the maximum amount of parallel processable server operation calls segmented by SOME/IP TP for that ClientServerToSignalMapping (see also figure 5.28).

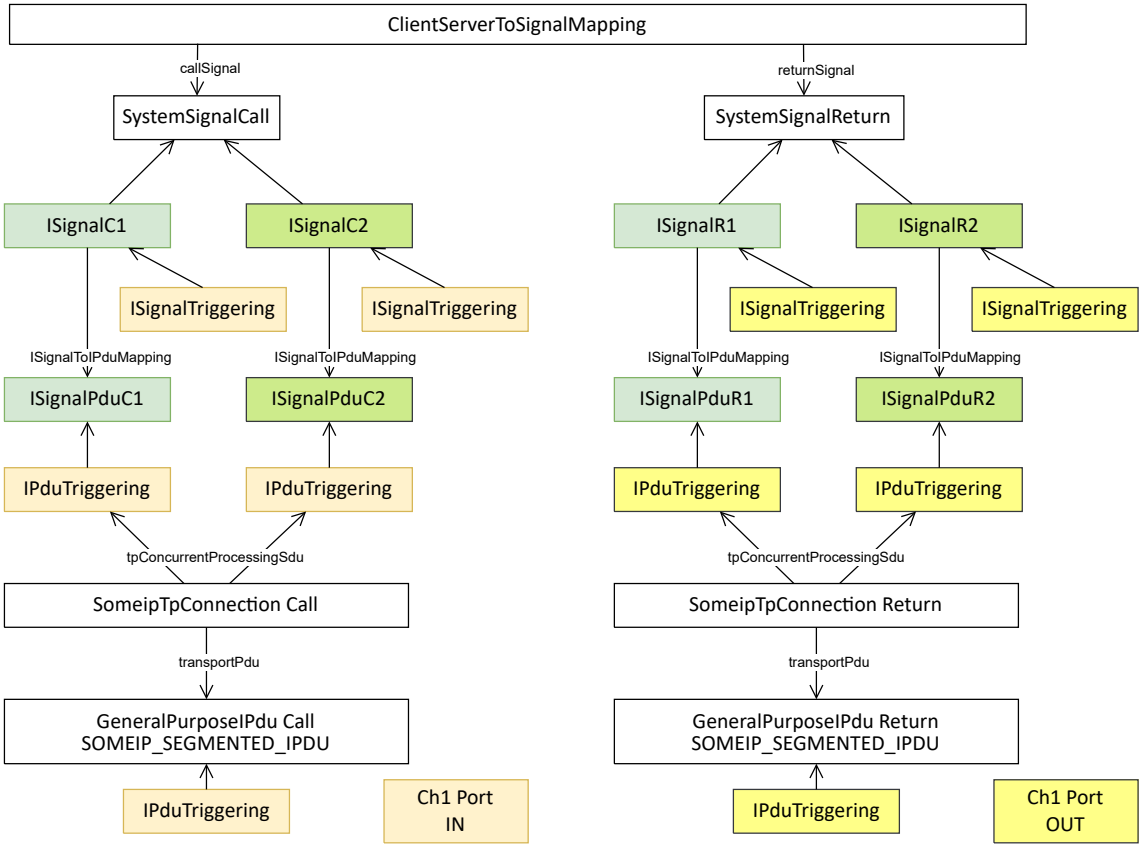

Figure 5.27: Concurrent SOME/IP TP segment example

For a SOME/IP TP segmented call message that defines more than one `SomeipTpConnection.tpConcurrentProcessingSdu` reception path (according to figure 5.27 *ISignalPduC1* and *ISignalPduC2*) the referenced `PduTriggering`s (via the `tpConcurrentProcessingSdu` reference) are a pool of upper layer reception path. The SOME/IP TP module will take one available path (currently not used reception path) to start the assembly of the newly arrived segmented call message.

This assembly process will take some time, and when during the ongoing assembly there is another segmented call message arriving from the same VLAN, this call message will arrive at the same `SomeipTpConnection` (because these call messages are coming from the same VLAN, i.e. `PhysicalChannel`). As there already is an assembly ongoing, the SOME/IP TP module will check if there is another reception path available and start a second, concurrent, assembly. If no free reception path is available (because there are already all defined `SomeipTpConnection.tpConcurrentProcessingSdu` path used) the newly arrived reception will be aborted, but the ongoing receptions will be processed further (unless they are aborted due to a timeout based on `SomeipTpChannel.rxTimeoutTime`).

When the assembly is successful or timed out, the occupied reception path is freed again, and can be used for the next call message arriving on that VLAN.

The RTE has a similar behavior for the return messages, as shown in figure 5.27.

[constr_9395] Each call ISignal shall have exactly one corresponding return ISignal that is associated with the same ClientServerOperation and PhysicalChannel

Imposition time: IT_SysDesc

⌈For each ClientServerToSignalMapping the number of call ISignals

- ISignals referring to a SystemSignal in the role systemSignal and
- that SystemSignal is referenced from the ClientServerToSignalMapping in role callSignal and
- the ISignal is referenced from an ISignalTriggering in role iSignal and
- that ISignalTriggering is aggregated by a PhysicalChannel in role iSignalTriggering

shall be exactly the same as the number of return ISignals

- ISignals referring to a SystemSignal in the role systemSignal and
- that SystemSignal is referenced from the ClientServerToSignalMapping in role returnSignal and
- the ISignal is referenced from an ISignalTriggering in role iSignal and
- that ISignalTriggering is aggregated by the same PhysicalChannel in role iSignalTriggering

on that PhysicalChannel.⌋

The rational for [constr_9395] is that, in case of concurrent processing of SOME/IP TP segmented IPdus (see also [TPS_SYST_02442]) the number of call and the return ISignals on a dedicated PhysicalChannel shall be identical.

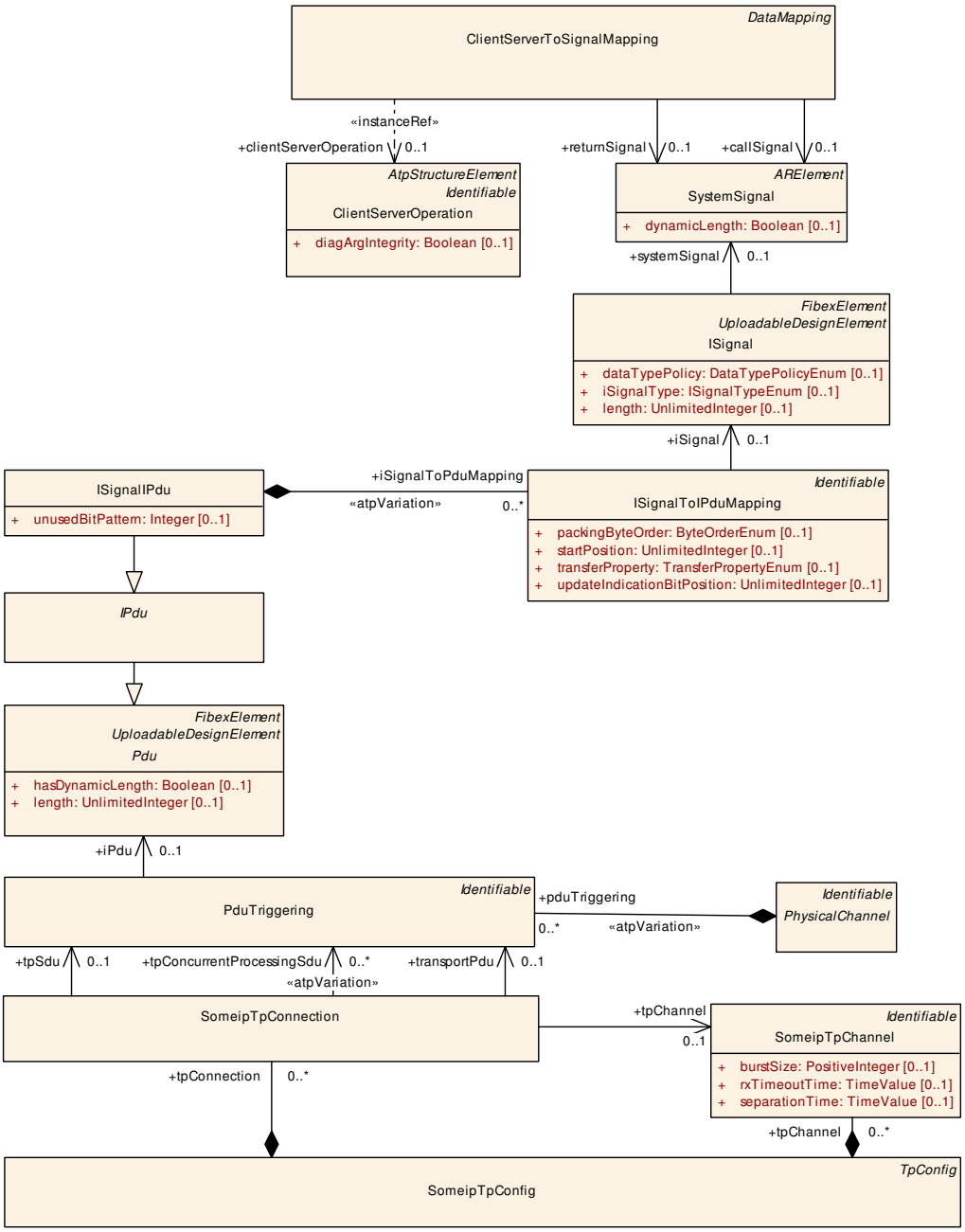

Figure 5.28: Mapping of a ClientServerOperation to a callSignal and a returnSignal using SOME/IP TP segmentation

Also the combination of the use-cases as defined in figure 5.26 and figure 5.27 is possible where a service is offered on several VLANs and concurrent processing on one (as indicated by figure 5.29 on *Ch1*) or even several `PhysicalChannel`s is configured.

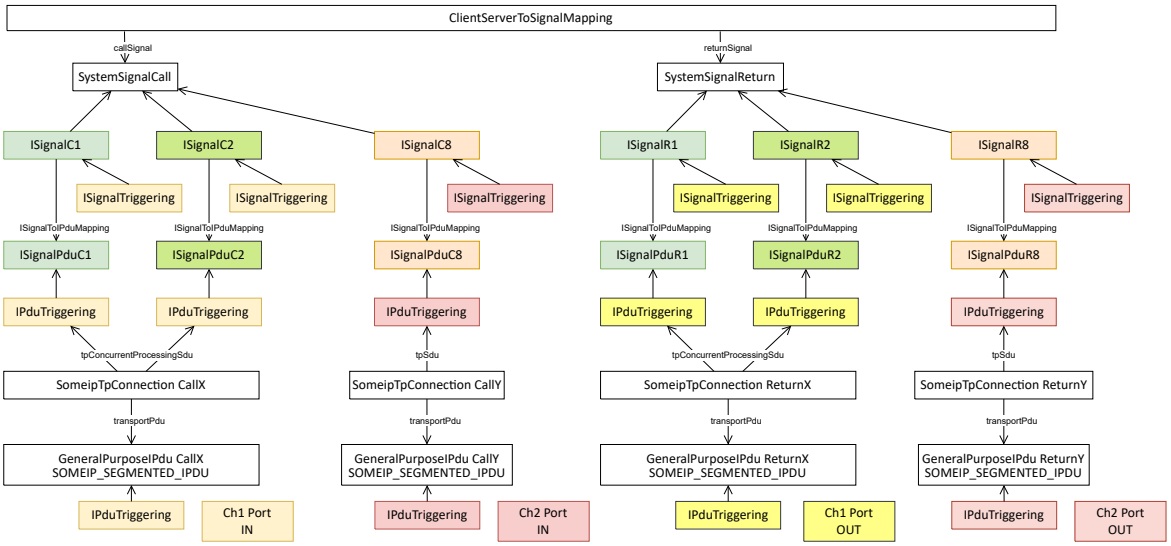

Figure 5.29: Concurrent SOME/IP TP segment and several VLANs example

Also it is required that a Client/Server interaction is fully provided on each PhysicalChannel, i.e. both callSignal and returnSignal shall be put onto the PhysicalChannel.

[constr_5274] ISignalTriggerings that represent the callSignal and returnSignal of the same ClientServerOperation on a PhysicalChannel shall be referenced by the same ClientServerToSignalMapping

Imposition time: CP: IT_SysDesc

⌈If on an EthernetPhysicalChannel an ISignalTriggering that refers to an ISignal that in turn refers to a SystemSignal that is referenced by a specific ClientServerToSignalMapping in the role callSignal is defined, then another ISignalTriggering shall be aggregated by the same EthernetPhysicalChannel and that ISignalTriggering shall refer to an ISignal that in turn refers to a SystemSignal that is referenced by the same ClientServerToSignalMapping in the role returnSignal, and vice versa.⌋

5.2.1.4 Mapping of Trigger to SystemSignal

[TPS_SYST_05001] Send a Trigger across a network ⌈In order to be able to send a Trigger across a network to trigger a RunnableEntity deployed to a different EcuInstance it is possible to define a TriggerToSignalMapping that maps a Trigger to a SystemSignal in the role systemSignal.⌋

[constr_1198] `TriggerToSignalMapping.systemSignal`s eligible for a `TriggerToSignalMapping` in case no `DataTransformation` is used

Imposition time: CP: `IT_EcuExt`, AP: `IT_Mani`

⌈The `ISignal` that is referenced by a `SystemSignal` that in turn is referenced by a `TriggerToSignalMapping` in the role `systemSignal` shall have the length attribute set to 0 if the `ISignal` does not reference a `DataTransformation` in the role `dataTransformation`.⌋

[constr_1199] `ISignal`s relating to `systemSignal`s eligible for a `TriggerToSignalMapping` shall use update bit in case no `DataTransformation` is used

Imposition time: CP: `IT_SysDesc`, AP: `IT_Mani`

⌈An `ISignal`

- that is used to reference a `systemSignal` that in turn is referenced by a `TriggerToSignalMapping` and
- does not reference a `DataTransformation` in the role `dataTransformation`

shall be referenced by an `ISignalToIPduMapping` where the attribute `updateIndicationBitPosition` is defined.⌋

Please note that according to [TPS_SYST_02021] the `updateIndicationBitPosition` shall not be defined if Large Data is enabled for the PDU in Com.

[constr_5258] `TriggerToSignalMapping.systemSignal`s eligible for a `TriggerToSignalMapping` in case `DataTransformation` is used

Imposition time: CP: `IT_SysDesc`

⌈The `ISignal` that is referenced by a `SystemSignal` that in turn is referenced by a `TriggerToSignalMapping` in the role `systemSignal` shall have its `length` attribute set to the value of `BufferProperties.headerLength` attribute of the respective `TransformationTechnology` if the `ISignal` references a `DataTransformation` in the role `dataTransformation` that in turn references the `TransformationTechnology`.⌋

For example, in case of the SOME/IP Transformer the 64 bit length covers the SOME/IP header defined by the SOME/IP Transformer (i.e. Request ID, Protocol Version, InterfaceVersion, Message Type, and Return Code) (as defined by [constr_5258] and [constr_3128]). The actual payload of the `ISignal` representing a trigger is 0.

[constr_5262] `SystemSignal` used for Trigger communication shall not be part of any `SystemSignalGroup`

Imposition time: CP: `IT_EcuExt`

⌈A `SystemSignal` that is target of a `TriggerToSignalMapping` in the role `systemSignal` shall not be referenced by a `SystemSignalGroup` in the role `systemSignal`.⌋

[TPS_SYST_02365] No support of Com Based Transformer for Trigger communication ⌈Due to [constr_5262] it is not possible to define a `SystemSignal` which is representing a `Trigger` to be part of a `SystemSignalGroup`, thus it is not possible to define a `Trigger` to be processed by a Com Based Transformer (as Com Based Transformer is enabled with the `ISignalGroup.comBasedSignalGroupTransformation`).⌋

[TPS_SYST_05002] The value of `startPosition` is irrelevant ⌈The value of `startPosition` shall not be considered inside an `ISignalToIPduMapping` that references an `ISignal` used to reference a `TriggerToSignalMapping.systemSignal` that in turn is referenced by a `TriggerToSignalMapping`.⌋

Please note that in case of a `TriggerToSignalMapping` for transmission of a `Trigger` over the network that has the `swImplPolicy` set to `queued` the sender will not get any indication that the receiver queue is full.

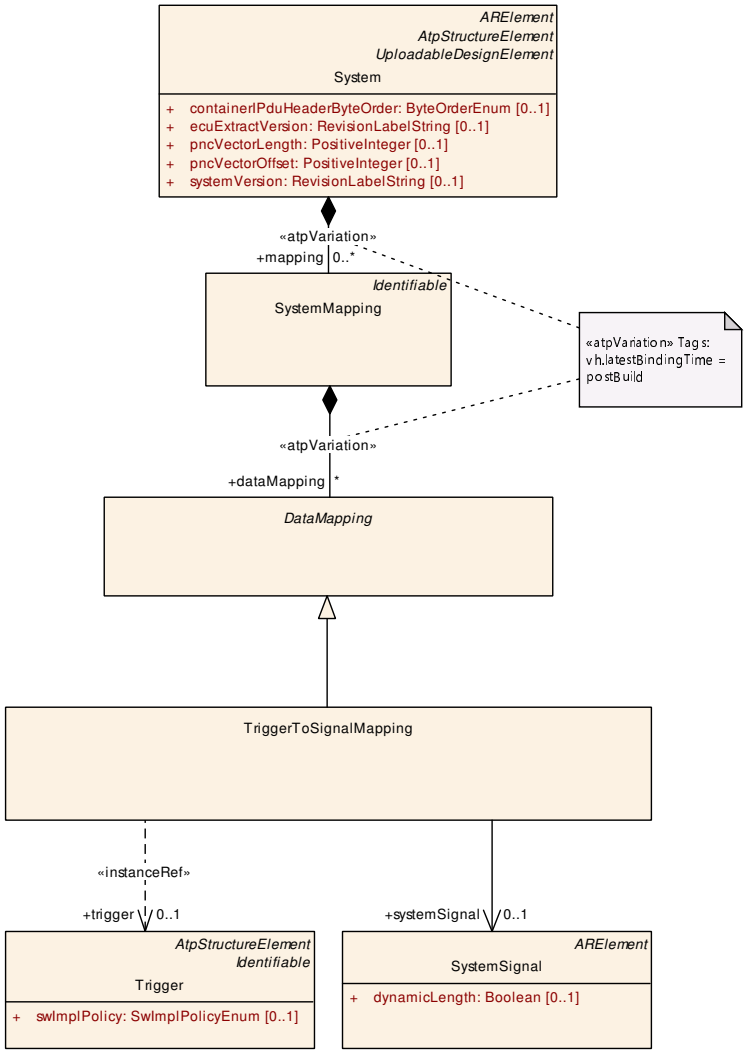

Figure 5.30: Structure of a `TriggerToSignalMapping`

Class	**TriggerToSignalMapping**			
Note	This meta-class represents the ability to map a trigger to a SystemSignal of size 0. The Trigger does not transport any other information than its existence, therefore the limitation in terms of signal length.			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
systemSignal	SystemSignal	0..1	ref	This is the SystemSignal taken to transport the Trigger over the network. **Tags:** atp.Splitkey=systemSignal xml.sequenceOffset=20
trigger	Trigger	0..1	iref	This represents the Trigger that shall be used to trigger RunnableEntities deployed to a remote ECU. **Tags:** atp.Splitkey=trigger, trigger.contextComposition, trigger.contextComponent, trigger.contextPort, trigger.targetTrigger **InstanceRef implemented by:** TriggerInSystemInstanceRef

Table 5.33: TriggerToSignalMapping

[constr_5477] Existence of `TriggerToSignalMapping.systemSignal`

Imposition time: CP: `IT_EcuExt`

⌈For each `TriggerToSignalMapping`, the reference to `SystemSignal` in the role `systemSignal` shall exist.⌋

[constr_5478] Existence of `TriggerToSignalMapping.trigger`

Imposition time: CP: `IT_EcuExt`

⌈For each `TriggerToSignalMapping`, the reference to `Trigger` in the role `trigger` shall exist.⌋

5.2.2 Signal Path Constraint

One task of the System Generator is to define the needed communication infrastructure (e.g. `ISignal`s, `Pdu`s, `Frame`s) between ECUs. The System Generator often has the choice between alternative paths through the topology. In the example shown in Figure 5.31 the System Generator would have the choice between two paths (Path1: CAN3 or Path2: CAN1-GW-CAN2) for a signal that is send by ECU2 and is received by ECU4. If no further information is given the decision will be made e.g. by means of boundary conditions like busload, transmissions speed, etc.

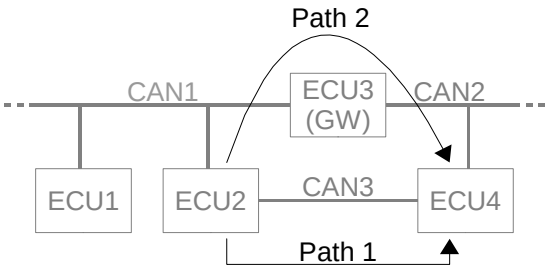

Figure 5.31: Example for a Communication Path

Signal Mapping Constraints allow to further restrict or specify the path(s) a signal is allowed to be transmitted over. A path is specified by an list of PhysicalChannels.

There exist four different constraints for signals regarding the signal path (see Figure 5.32):

[TPS_SYST_01041] CommonSignalPath definition ⌈The CommonSignalPath describes that two or more signals shall take the same path in the topology.⌋

[TPS_SYST_01042] ForbiddenSignalPath definition ⌈The ForbiddenSignalPath describes the path that one or more signals shall not take in the topology, e.g. in case of safety critical transmission.⌋

[TPS_SYST_01043] PermissibleSignalPath definition ⌈The PermissibleSignalPath describes the path one or more signals may take in the topology. If more than one PermissibleSignalPath is defined for the same signal/operation attributes, any of them may be chosen.⌋

[TPS_SYST_01044] SeparateSignalPath definition ⌈The SeparateSignalPath describes that two or more signals shall take separate paths in the topology e.g. in case of redundant transmission.⌋

It is also possible that the same signal is aggregated two times by the SeparateSignalPath element to indicate that this signal should be transmitted redundantly over two different paths.

The meta-model part, which describes the Communication Path constraints, will be explained in the following sections.

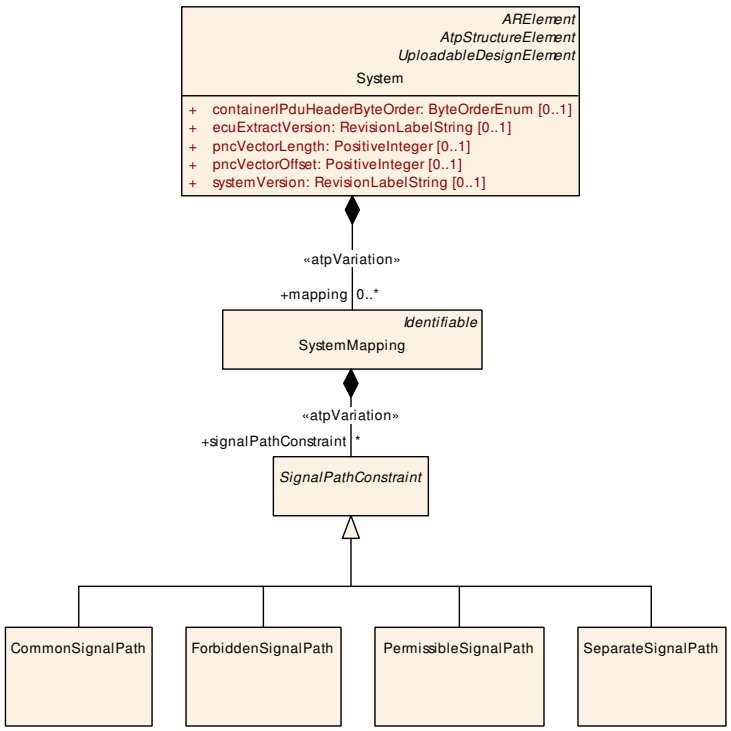

Figure 5.32: Communication Path Description (SignalPathConstraints)

5.2.2.1 CommonSignalPath

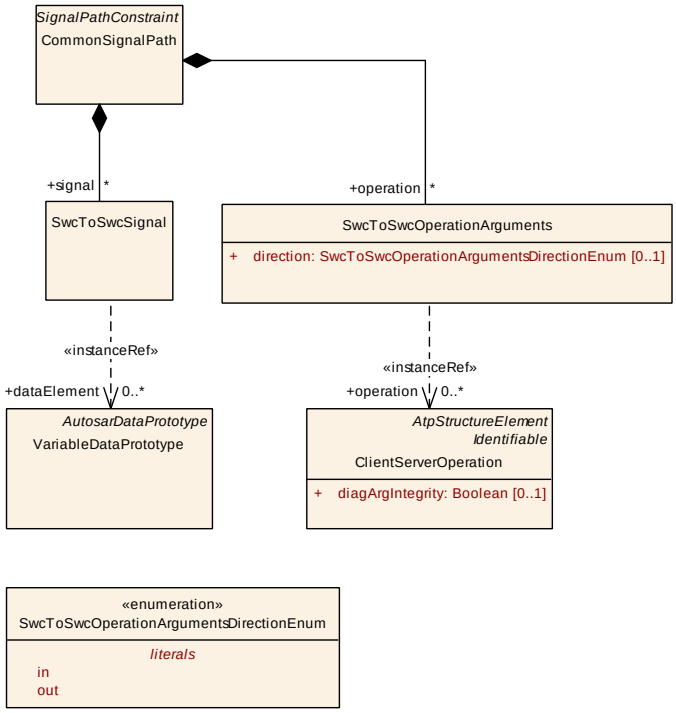

Figure 5.33: Description of signals that shall take the same way in the topology (CommonSignalPath)

Class	CommonSignalPath			
Note	The CommonSignalPath describes that two or more SwcToSwcSignals and/or SwcToSwcOperation Arguments shall take the same way (Signal Path) in the topology.			
Base	*ARObject*, *SignalPathConstraint*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation	SwcToSwcOperation Arguments	*	aggr	The arguments sent in one direction (either from client to server or server to client) of the operations that shall take the same signal path. **Tags:** atp.Splitkey=operation
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that shall take the same way (Signal Path) in the topology. **Tags:** atp.Splitkey=signal

Table 5.34: CommonSignalPath

Class	SwcToSwcSignal			
Note	The SwcToSwcSignal describes the information (data element) that is exchanged between two SW Components. On the SWC Level it is possible that a SW Component sends one data element from one P-Port to two different SW Components (1:n Communication). The SwcToSwcSignal describes exactly the information which is exchanged between one P-Port of a SW Component and one R-Port of another SW Component.			
Base	*ARObject*			
Aggregated by	CommonSignalPath.signal, ForbiddenSignalPath.signal, PermissibleSignalPath.signal, SeparateSignal Path.signal			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	VariableDataPrototype	*	iref	Reference to a data element on the PPortPrototype and to the same data element on the RPortPrototype. **Tags:** atp.Splitkey=dataElement, dataElement.context Component, dataElement.contextComposition, data Element.contextPort, dataElement.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef

Table 5.35: SwcToSwcSignal

[constr_5481] Existence of `SwcToSwcSignal.dataElement`

Imposition time: CP: `IT_SysDesc`

⌈Each `SwcToSwcSignal` shall reference exactly two `VariableDataPrototype`s in the role `dataElement`.⌋

Class	SwcToSwcOperationArguments			
Note	The SwcToSwcOperationArguments describes the information (client server operation arguments, plus the operation identification, if required) that are exchanged between two SW Components from exactly one client to one server, or from one server back to one client. The direction attribute defines which direction is described. If direction == IN, all arguments sent from the client to the server are described by the SwcToSwcOperationArguments, in direction == OUT, it's the arguments sent back from server to client.			
Base	*ARObject*			
Aggregated by	CommonSignalPath.operation, ForbiddenSignalPath.operation, PermissibleSignalPath.operation, SeparateSignalPath.operation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**SwcToSwcOperationArguments**			
direction	SwcToSwcOperation ArgumentsDirection Enum	0..1	attr	Direction addressed by this SwcToSwcClientServer Operation element.
operation	ClientServerOperation	*	iref	Reference to the operation at the client and at the server side whose arguments are described by SwcToSwc OperationArguments. The two ports referenced shall be connected by a connector in the software component description. **Tags:** atp.Splitkey=operation, operation.context Composition, operation.contextComponent, operation.contextPort, operation.targetOperation **InstanceRef implemented by:** OperationInSystem InstanceRef

Table 5.36: SwcToSwcOperationArguments

[constr_5482] Existence of SwcToSwcOperationArguments.direction

Imposition time: CP: IT_SysDesc

⌈For each SwcToSwcOperationArguments, the attribute direction shall exist.⌋

[constr_5483] Existence of SwcToSwcOperationArguments.operation

Imposition time: CP: IT_SysDesc

⌈Each SwcToSwcOperationArguments element shall reference exactly two ClientServerOperations in the role operation.⌋

Enumeration	**SwcToSwcOperationArgumentsDirectionEnum**
Note	Direction addressed by this element.
Aggregated by	SwcToSwcOperationArguments.direction
Literal	*Description*
in	IN (all IN and INOUT arguments) **Tags:** atp.EnumerationLiteralIndex=0
out	OUT (all OUT and INOUT arguments) . **Tags:** atp.EnumerationLiteralIndex=1

Table 5.37: SwcToSwcOperationArgumentsDirectionEnum

5.2.2.2 ForbiddenSignalPath

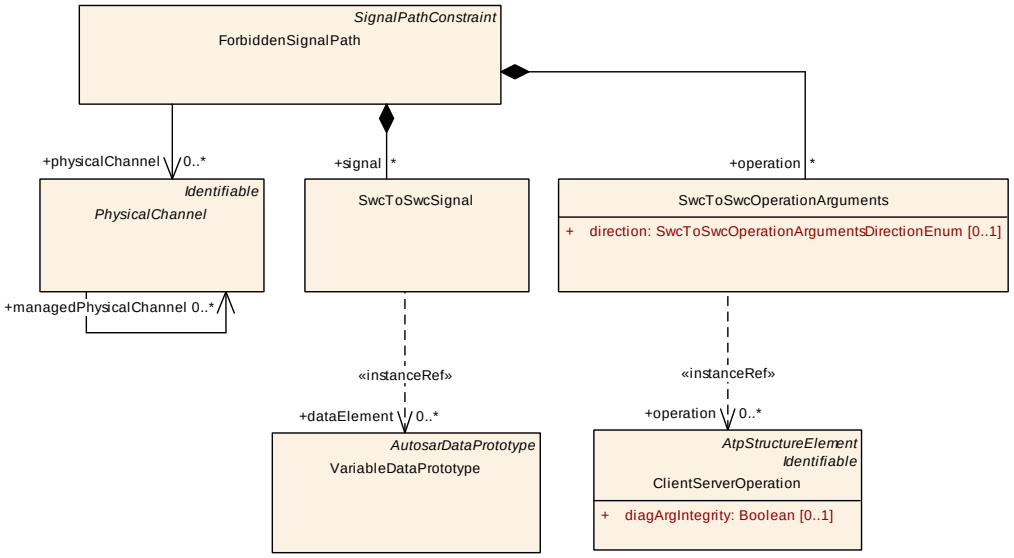

Figure 5.34: Description of the signal path that a signal shall not take in the topology (ForbiddenSignalPath)

Class	**ForbiddenSignalPath**			
Note	The ForbiddenSignalPath describes the physical channels which an element shall not take in the topology. Such a signal path can be a constraint for the communication matrix, because such a path has an effect on the frame generation and the frame path.			
Base	*ARObject*, *SignalPathConstraint*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	SwcToSwcOperation Arguments	*	aggr	Reference to the operation arguments of one operation which shall not take the predefined way in the topology. **Tags:** atp.Splitkey=operation
physical Channel	PhysicalChannel	*	ref	The SwcToSwcSignal shall not be transmitted on one of these physical channels. **Tags:** atp.Splitkey=physicalChannel
signal	SwcToSwcSignal	*	aggr	The data element which shall not take the predefined way in the topology. **Tags:** atp.Splitkey=signal

Table 5.38: ForbiddenSignalPath

[constr_5484] Existence of `ForbiddenSignalPath.physicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈For each `ForbiddenSignalPath`, at least one reference to `PhysicalChannel` in the role `physicalChannel` shall exist.⌋

5.2.2.3 PermissibleSignalPath

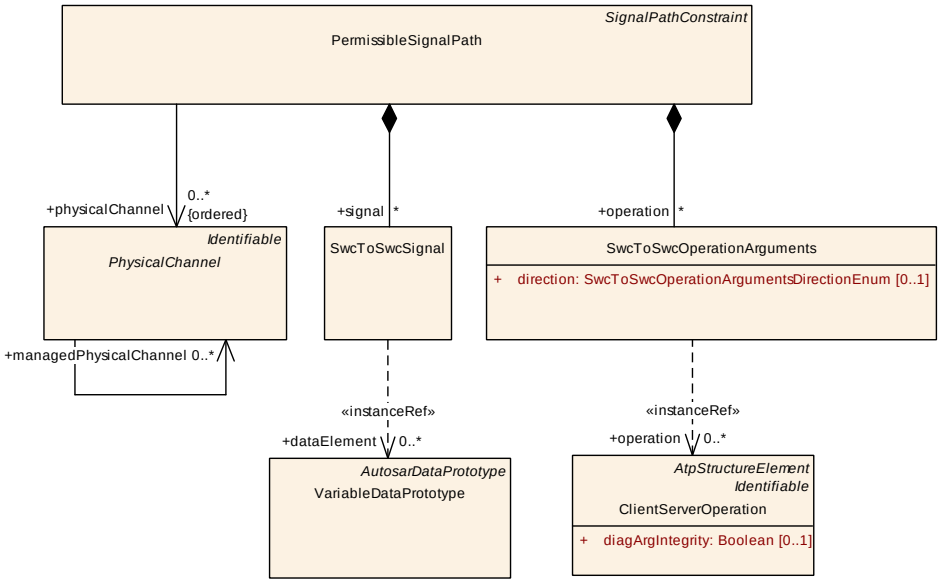

Figure 5.35: Description of the signal path that a signal shall take in the topology (PermissibleSignalPath)

Class	**PermissibleSignalPath**			
Note	The PermissibleSignalPath describes the way a data element shall take in the topology. The path is described by ordered references to PhysicalChannels. If more than one PermissibleSignalPath is defined for the same signal/operation attributes, any of them can be chosen. Such a signal path can be a constraint for the communication matrix . This path describes that one data element should take path A (e.g. 1. CAN channel, 2. LIN channel) and not path B (1. CAN channel, FlexRay channel A). This has an effect on the frame generation and the frame path.			
Base	*ARObject*, *SignalPathConstraint*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	SwcToSwcOperation Arguments	*	aggr	The arguments of an operation that can take the predefined way in the topology. **Tags:** atp.Splitkey=operation
physical Channel (ordered)	PhysicalChannel	*	ref	The SwcToSwcSignal can be transmitted on one of these physical channels.
signal	SwcToSwcSignal	*	aggr	The data element which can take the predefined way in the topology. **Tags:** atp.Splitkey=signal

Table 5.39: PermissibleSignalPath

[constr_5485] Existence of `PermissibleSignalPath.physicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈For each `PermissibleSignalPath`, at least one reference to `PhysicalChannel` in the role `physicalChannel` shall exist.⌋

5.2.2.4 SeparateSignalPath

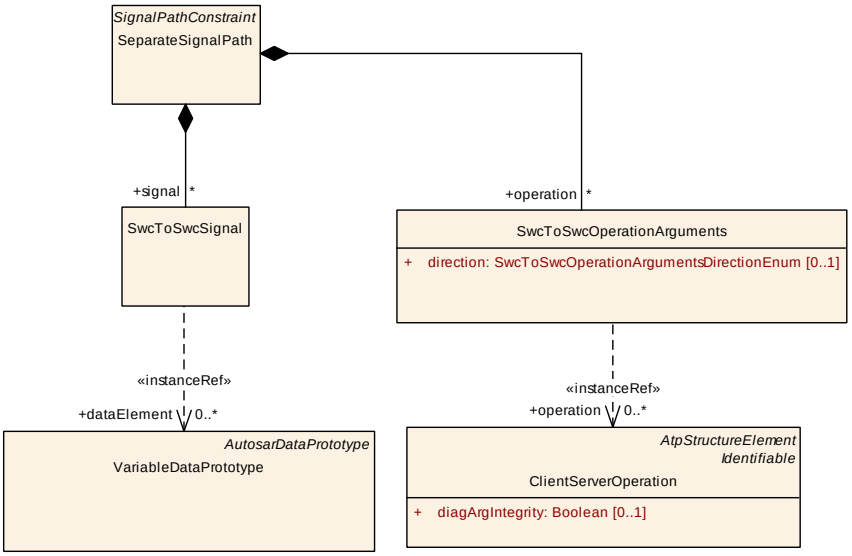

Figure 5.36: Description of signals that shall not take the same way in the topology (SeparateSignalPath)

Class	**SeparateSignalPath**			
Note	The SeparateSignalPath describes that two SwcToSwcSignals and/or SwcToSwcOperationArguments shall not take the same way (Signal Path) in the topology (e.g. Redundancy). This means that the signals are not allowed to share even a single physical channel in their path.			
Base	*ARObject*, *SignalPathConstraint*			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
operation	SwcToSwcOperation Arguments	*	aggr	The SwcToSwcOperationArguments that shall not take the same way (Signal Path) in the topology. **Tags:** atp.Splitkey=operation
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that shall not take the same way (Signal Path) in the topology. **Tags:** atp.Splitkey=signal

Table 5.40: SeparateSignalPath

5.3 Compatibility Rules for DataMapping between a DataPrototype and a SystemSignal

This section defines the compatibility rules for the DataMapping between a DataPrototype that is typed by an ApplicationDataType and a SystemSignal. The goal of those compatibility rules is to help the System Designer to map software component data to SystemSignals so that the information defined by Software Component developers is respected during the network design and inconsistent system descriptions are prevented.

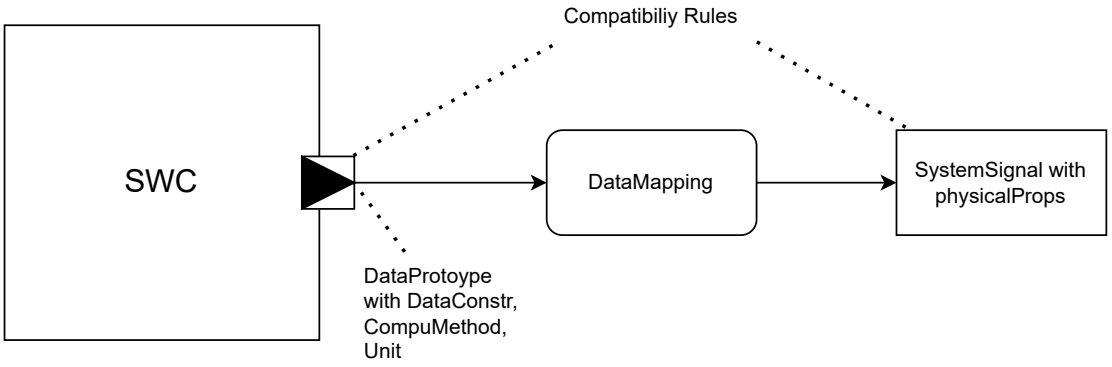

Figure 5.37: Compatibiliy checks for DataMapping between DataPrototypes typed by an ApplicationDataTypeand a SystemSignal

For a DataPrototype that is only typed by an ImplementationDataType and for which a DataTypeMap to an ApplicationDataType is not defined the compatibility rules in this chapter are not applicable. The reason for this restriction is that a SystemSignal defines physicalProps that can be checked against properties that are defined on ApplicationDataType level.

Please also note that the specification items and constraints in this section are only valid for data for which the DataTransformation is not used or in which the data is transformed by the COM Based Transformer (see [TPS_SYST_02058]). Another restriction is that those rules only cover the Sender-Receiver Communication.

5.3.1 DataPrototype typed by category VALUE

[TPS_SYST_02423] Compatibility of VariableDataPrototype typed by primitive data type of category VALUE in a PPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping ⌈If a VariableDataPrototype typed by an ApplicationPrimitiveDataType of category VALUE in a PPortPrototype is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9372].
- the units of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the units of SystemSignal.physicalProps according to [constr_1052] defined in [4].

⌋

[TPS_SYST_02424] Compatibility of VariableDataPrototype typed by primitive data type of category VALUE in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping ⌈If a VariableDataPrototype typed by an ApplicationPrimitiveDataType of category VALUE in a RPortPrototype is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9371].
- the units of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the units of SystemSignal.physicalProps according to [constr_1052] defined in [4].

⌋

[TPS_SYST_02425] Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category VALUE in a PPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping ⌈If a VariableDataPrototype in a PPortPrototype typed by ApplicationCompositeDataType is mapped by a SenderReceiverToSignalGroupMapping to a SystemSignalGroup then the following conditions shall be fulfilled by an ApplicationCompositeElementDataPrototype that is aggregated

by the ApplicationCompositeDataType and the SystemSignal of the SystemSignalGroup to which the ApplicationCompositeElementDataPrototype is mapped in case that the ApplicationCompositeElementDataPrototype is typed by an ApplicationPrimitiveDataType of category VALUE:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9372].
- the units of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype is compatible to the units of SystemSignal.physicalProps according to [constr_1052] defined in [4].

⌋

[TPS_SYST_02426] Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category VALUE in a RPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping ⌈If a VariableDataPrototype in a RPortPrototype typed by ApplicationCompositeDataType is mapped by a SenderReceiverToSignalGroupMapping to a SystemSignalGroup then the following conditions shall be fulfilled by an ApplicationCompositeElementDataPrototype that is aggregated by the ApplicationCompositeDataType and the SystemSignal of the SystemSignalGroup to which the ApplicationCompositeElementDataPrototype is mapped in case that the ApplicationCompositeElementDataPrototype is typed by an ApplicationPrimitiveDataType of category VALUE:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9371].
- the units of AutosarDataType.swDataDefProps defined on ApplicationDataType of the ApplicationCompositeElementDataPrototype is compatible to the units of SystemSignal.physicalProps according to [constr_1052] defined in [4].

⌋

5.3.2 DataPrototype typed by category BOOLEAN

[TPS_SYST_02427] Compatibility of VariableDataPrototype typed by primitive data type of category BOOLEAN in a PPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping ⌈If a VariableDataPrototype typed by an ApplicationPrimitiveDataType of category BOOLEAN in a PPortPrototype is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9372].

⌋

[TPS_SYST_02428] Compatibility of VariableDataPrototype typed by primitive data type of category BOOLEAN in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping ⌈If a VariableDataPrototype typed by an ApplicationPrimitiveDataType of category BOOLEAN in a RPortPrototype is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9371].

⌋

[TPS_SYST_02429] Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category BOOLEAN in a PPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping ⌈If a VariableDataPrototype in a PPortPrototype typed by ApplicationCompositeDataType is mapped by a SenderReceiverToSignalGroupMapping to a SystemSignalGroup then the following conditions shall be fulfilled by an ApplicationCompositeElementDataPrototype that is aggregated by the ApplicationCompositeDataType and the SystemSignal of the SystemSignalGroup to which the ApplicationCompositeElementDataPrototype is mapped in case that the ApplicationCompositeElementDataPrototype is typed by an ApplicationPrimitiveDataType of category BOOLEAN:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9372].

⌋

[TPS_SYST_02430] Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category BOOLEAN in a RPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping ⌈If a VariableDataPrototype in a RPortPrototype typed by ApplicationCompositeDataType is mapped by a SenderReceiverToSignalGroupMapping to a SystemSignalGroup then the following conditions shall be fulfilled by an ApplicationCompositeElementDataPrototype that is aggregated by the ApplicationCompositeDataType and the SystemSignal of the SystemSignalGroup to which the ApplicationCompositeElementDataPrototype is mapped in case that the ApplicationCompositeElementDataPrototype is typed by an ApplicationPrimitiveDataType of category BOOLEAN:

- the dataConstrs of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].
- the compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype is compatible to the compuMethod of SystemSignal.physicalProps according to [constr_9371].

⌋

5.3.3 DataPrototype typed by category STRING

[TPS_SYST_02431] Compatibility of `VariableDataPrototype` typed by primitive data type of `category` STRING in a `PortPrototype` and a `SystemSignal` to which the `VariableDataPrototype` is mapped by `SenderReceiverToSignalMapping` ⌈If a VariableDataPrototype typed by an ApplicationPrimitiveDataType of category STRING in a PortPrototype is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- the swTextPropss of AutosarDataType.swDataDefProps defined on ApplicationDataType of the VariableDataPrototype are compatible to the swTextPropss of SystemSignal.physicalProps according to [constr_10632] (see [4]).

⌋

[TPS_SYST_02432] Compatibility of `ApplicationCompositeElementDataPrototype` typed by primitive data type of `category` STRING in a `PortPrototype` and a `SystemSignal` to which the `ApplicationCompositeElementDataPrototype` is mapped by `SenderReceiverToSignalGroupMapping` ⌈If a VariableDataPrototype in a PortPrototype typed by ApplicationCompositeDataType is mapped by a SenderReceiverToSignalGroupMapping to a SystemSignalGroup then the following conditions shall be fulfilled by an ApplicationCompositeElementDataPrototype that is aggregated by the ApplicationCompositeDataType and the SystemSignal of the SystemSignalGroup to which the ApplicationCompositeElementDataPrototype is mapped in case that the ApplicationCompositeElementDataPrototype is typed by an ApplicationPrimitiveDataType of category STRING:

- the swTextPropss of AutosarDataType.swDataDefProps defined on ApplicationPrimitiveDataType of the ApplicationCompositeElementDataPrototype are compatible to the swTextPropss of SystemSignal.physicalProps according to [constr_10632] (see [4]).

⌋

5.3.4 DataPrototype typed by an ARRAY

[TPS_SYST_02433] Compatibility of `VariableDataPrototype` typed by an ARRAY in a `PPortPrototype` and a `SystemSignal` to which the `VariableDataPrototype` is mapped by `SenderReceiverToSignalMapping` ⌈If a VariableDataPrototype in a PPortPrototype typed by ApplicationArrayDataType that fulfills the rules for a primitive Data Mapping according to [TPS_SYST_02084] is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- if the VariableDataPrototype defines dataConstrs in DataPrototype.swDataDefProps then the dataConstrs shall be compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].
- if the VariableDataPrototype is not defining dataConstrs in DataPrototype.swDataDefProps then dataConstrs of DataPrototype.swDataDefProps defined on ApplicationArrayElement that is aggregated by the ApplicationArrayDataType shall be compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9373].

⌋

[TPS_SYST_02434] Compatibility of VariableDataPrototype typed by an ARRAY in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping ⌈If a VariableDataPrototype in a RPortPrototype typed by ApplicationArrayDataType that fulfills the rules for a primitive Data Mapping according to [TPS_SYST_02084] is mapped by a SenderReceiverToSignalMapping to a SystemSignal then the following conditions shall be fulfilled by the VariableDataPrototype and the SystemSignal:

- if the VariableDataPrototype defines dataConstrs in DataPrototype.swDataDefProps then the dataConstrs shall be compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].
- if the VariableDataPrototype is not defining dataConstrs in DataPrototype.swDataDefProps then dataConstrs of DataPrototype.swDataDefProps defined on ApplicationArrayElement that is aggregated by the ApplicationArrayDataType shall be compatible to the dataConstrs of SystemSignal.physicalProps according to [constr_9374].

⌋

Please note that [TPS_SYST_02433] an [TPS_SYST_02434] are not considering compuMethods and units since these are not applicable for DataPrototypes of category ARRAY according to [constr_1289].

5.3.5 Compatibility of SwDataDefProps

[constr_9371] Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a RPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the DataPrototype in a RPortPrototype that is mapped to a SystemSignal is compatible to the compuMethods of physicalProps of the SystemSignal if the following conditions apply:

- [constr_1163] (see [4]) that defines the compatibility of CompuMethods is fulfilled,
- [constr_1153] (see [4]) that defines the compatibility of CompuScales of the enclosing CompuMethod is fulfilled,
- [constr_1156] (see [4]) that defines the compatibility of CompuScales which contribute to tabular conversion by having a compuConst,
- [constr_1176] (see [4]) that defines the compatibility of CompuScaless of category LINEAR and RAT_FUNC is fulfilled,
- [constr_1192] (see [4]) that defines the compatibility of CompuScaless of category IDENTICAL to RAT_FUNC or LINEAR is fulfilled,
- [constr_9376] is fulfilled.

⌋

[constr_9372] Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a PPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The compuMethods of AutosarDataType.swDataDefProps defined on ApplicationDataType of the DataPrototype in a PPortPrototype that is mapped to a SystemSignal is compatible to the compuMethods of physicalProps of the SystemSignal if the following conditions apply:

- [constr_1163] (see [4]) that defines the compatibility of CompuMethods is fulfilled,
- [constr_1153] (see [4]) that defines the compatibility of CompuScales of the enclosing CompuMethod is fulfilled,
- [constr_1156] (see [4]) that defines the compatibility of CompuScales which contribute to tabular conversion by having a compuConst,
- [constr_1176] (see [4]) that defines the compatibility of CompuScaless of category LINEAR and RAT_FUNC is fulfilled,
- [constr_1192] (see [4]) that defines the compatibility of CompuScaless of category IDENTICAL to RAT_FUNC or LINEAR is fulfilled,
- [constr_9375] is fulfilled.

⌋

[constr_9373] Compatibility of DataConstr between SystemSignal and DataPrototype in a PPortPrototype

Imposition time: CP: IT_SysDesc

⌈The DataConstrs (e.g. the limits) defined in the ApplicationDataType of the DataPrototype in a PPortPrototype shall be within the DataConstrs defined

in the SystemSignal.physicalProps of the SystemSignal to which the DataPrototype is mapped.⌋

[constr_9374] Compatibility of DataConstr between SystemSignal and DataPrototype in a RPortPrototype

Imposition time: CP: IT_SysDesc

⌈The DataConstrs (e.g. the limits) defined in the SystemSignal.physicalProps of the SystemSignal to which the DataPrototype in a RPortPrototype is mapped shall be within the DataConstrs defined in the ApplicationDataType of the DataPrototype.⌋

[constr_9375] Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a PPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The set of CompuScales in the CompuMethod defined in the ApplicationDataType of the DataPrototype in a PPortPrototype that is mapped to the SystemSignal shall be a subset of the set of CompuScales defined in the CompuMethod of the SystemSignal.⌋

[constr_9376] Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a RPortPrototype mapped to the SystemSignal

Imposition time: CP: IT_SysDesc

⌈The set of CompuScales in the CompuMethod of the SystemSignal shall be a subset of the set of CompuScales defined in the ApplicationDataType of the DataPrototype in a RPortPrototype.⌋

5.4 RTE and basic software resource estimations

Important constraints for system partitioning are the available resources on the ECUs in the system. For SW components, the resource estimations can be stated in SW component descriptions. It is however not only SW components that require resources. AUTOSAR RTE and basic software running on the ECU have resource needs as well.

The realization of the RTE and the kind of basic software to be run on a certain ECU depend on the implicit and explicit usage of all basic software by the software components. The software components need to communicate internally and with software components on other ECUs. Furthermore, they have different needs with respect to scheduling. This results in implicit use of e.g. communication and operating system software. In addition, the software components make explicit use of basic software when they e.g. utilize system services (e.g. diagnostics) and access sensors/actuators via the I/O abstraction layer or the Complex Driver abstraction layer. Thus, the

resource consumption of the RTE and the basic software depend on the SW Components mapped to the ECU, since this determines the exact configuration of the RTE and the basic software.

[TPS_SYST_01126] Resource Consumption for RTE and basic software ⌈The resource consumption for RTE and basic software may be specified using class `EcuResourceEstimation`. Each estimation is performed for a specific ECU and for a specific set of SW mapped to that ECU (reference from `EcuResourceEstimation` to `EcuInstance` and `SwcToEcuMapping`).⌋

Different resource estimations for a specific ECU, but with different mappings may exist, e.g. for different variants of the system, or to show the difference of resource needs for different mappings. The `EcuResourceEstimation` aggregates the meta-class `ResourceConsumption` from the GenericStructure package each for RTE and basic software, which specifies stack and heap usage and execution time.

`ExecutionTime` and `StackUsage` are used to provide information on the implementation specific resource usage of the `ExecutableEntity` defined in the `InternalBehavior` of SW-Component respectively in the `BswInternalBehavior` of BSW Module. `MemorySection` documents the resources needed to load the object file containing the implementation on the ECU. `HeapUsage` describes the dynamic memory usage of the software.

Figure 5.38 shows the meta-model for resource estimations for RTE and basic SW.

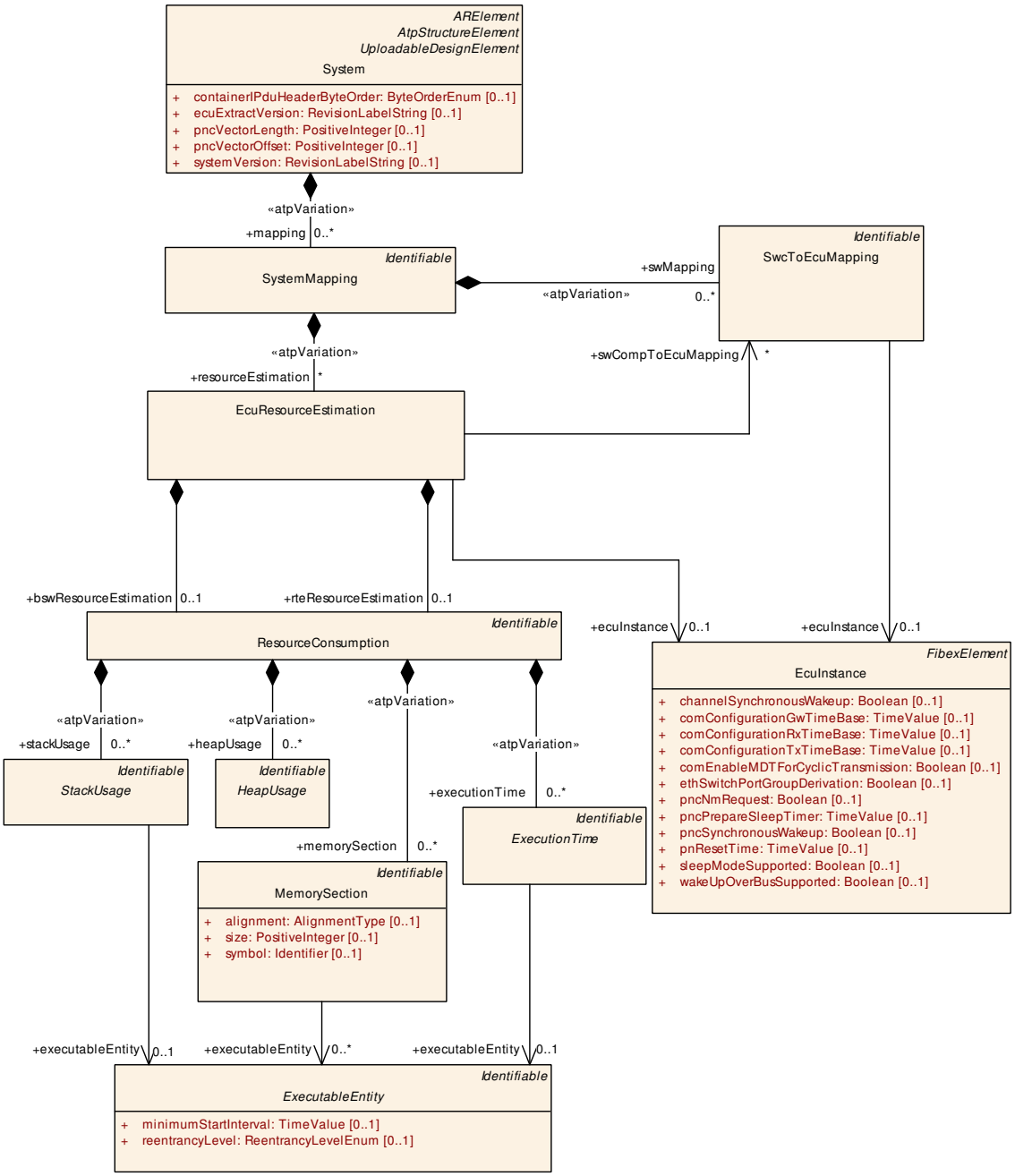

Figure 5.38: ECU resource estimations (ResourceEstimation)

[constr_3005] valid `EcuResourceEstimation`

Imposition time: CP: `IT_SysDesc`

⌈The same `EcuInstance` shall be referenced directly from the `EcuResourceEstimation` and from the `SwcToEcuMapping`:

EcuResourceEstimation.swCompToEcuMapping.ecuInstance == EcuResourceEstimation.ecuInstance⌋

Class	EcuResourceEstimation			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	*ARObject*			
Aggregated by	SystemMapping.resourceEstimation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the basic software. **Tags:** atp.Splitkey=bswResourceEstimation, bsw ResourceEstimation.shortName
ecuInstance	EcuInstance	0..1	ref	Reference to the ECU this estimation is done for. **Tags:** atp.Splitkey=ecuInstance
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ecu resource estimation **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-10
rteResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the run time environment. **Tags:** atp.Splitkey=rteResourceEstimation, rteResource Estimation.shortName
swCompToEcu Mapping	SwcToEcuMapping	*	ref	References to SwcToEcuMappings that have been taken into account for the resource estimations. This way it is possible to define dfferent EcuResourceEstimations with diifferent mappings, e.g. before and after mapping an additional SW component. **Tags:** atp.Splitkey=swCompToEcuMapping This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.41: EcuResourceEstimation

[constr_5480] Existence of `EcuResourceEstimation.ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `EcuResourceEstimation`, the reference to `EcuInstance` in the role `ecuInstance` shall exist.⌋

Class	ResourceConsumption			
Note	Description of consumed resources by one implementation of a software.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuResourceEstimation.bswResourceEstimation, EcuResourceEstimation.rteResourceEstimation, *Implementation*.resourceConsumption, StateDependentStartupConfig.resourceConsumption			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessCount Set	AccessCountSet	*	aggr	Set of access count values **Stereotypes:** atpVariation **Tags:** atp.Splitkey=accessCountSet, accessCountSet.variation Point.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	ResourceConsumption			
executionTime	ExecutionTime	*	aggr	Collection of the execution time descriptions for this implementation. The aggregation of executionTime is subject to variability with the purpose to support the conditional existence of runnable entities. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=executionTime, executionTime.shortName, executionTime.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
heapUsage	HeapUsage	*	aggr	Collection of the heap memory allocated by this implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=heapUsage, heapUsage.shortName, heapUsage.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
memorySection	MemorySection	*	aggr	An abstract memory section required by this Implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=memorySection, memorySection.shortName, memorySection.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
sectionNamePrefix	SectionNamePrefix	*	aggr	A prefix to be used for the memory section symbol in the code. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sectionNamePrefix, sectionNamePrefix.shortName, sectionNamePrefix.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
stackUsage	StackUsage	*	aggr	Collection of the stack memory usage for each runnable entity of this implementation. The aggregation of Stack Usage is subject to variability with the purpose to support the conditional existence of runnable entities. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=stackUsage, stackUsage.shortName, stackUsage.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.42: ResourceConsumption

The element `ResourceConsumption` and the sub-elements `HeapUsage`, `MemorySection`, `StackUsage` and `ExecutionTime` are described in more detail in the BSW Module Description [22].

5.5 Communication Management Mapping

The Communication Management Mapping is used to define the link between the application's communication request and the actual network topology.

Some applications may have the need to control the communication infrastructure during their run-time. So they can control at which point in time certain communication shall be enabled (requested) and when the communication can be disabled (release request). How the control interaction is to be setup is specified in the Software Component Template[4].

Which parts of the communication relies on such control is expressed using PortGroups. The PortGroup collect the set of PortPrototypes which shall be available when the application requests communication.

The handling of communication management comes in two flavors, defined by the category of the PortGroup:

- PARTIAL_NETWORKING
- MODE_MANAGEMENT

PortGroups of category PARTIAL_NETWORKING determine to control the communication via Partial Network Clusters (see Chapter 5.5.1).

PortGroups of category MODE_MANAGEMENT determine to control the communication directly via ComManager (see Chapter 5.5.2). This is specifically required in cases where no partial networking is used or on network which do not support partial networking (like Lin).

5.5.1 Partial Networking

The AUTOSAR BSW stack supports power saving during vehicle operation time with the partial networking mechanism. This mechanism allows to shut down and startup the bus communication interfaces of groups of ECUs (Partial Network Cluster) during normal bus communication.

On the VFB Level Partial Networks are represented by Virtual Function Clusters and are described with PortGroups. The Virtual Function Cluster groups the communication necessary to realize one or more vehicle functions that can become activated/deactivated during normal vehicle operation. Virtual Function Clusters are described in more detail in [4]. The Virtual Function Clusters are mapped onto Partial Network Clusters.

There are two variants of the partial networking mechanism:

- The *static* partial networking mechanism where the mapping of Partial Network Clusters (PNCs) to Virtual Function Clusters (and thus ECUs) is defined statically during vehicle development and does not change afterwards.

- The *dynamic* partial networking mechanism that enables changes of the mapping between PNCs and ECUs during life time of a vehicle by introducing a learning algorithm (for details, see chapter 5.5.1.4).

The use of partial networking in general is defined based on the element PncMapping. If partial networking is used, the *dynamic* partial networking is then further defined by the attribute dynamicPncToChannelMappingEnabled of the element CommunicationConnector.

[TPS_SYST_01133] Partial Network Clusters ⌈Partial Network Clusters are defined using PncMapping and may impact:

- ISignalIPduGroup via PncMapping.pncGroup
- PdurIPduGroup via PncMapping.pncPdurGroup
- ConsumedProvidedServiceInstanceGroup via PncMapping.pncConsumedProvidedServiceInstanceGroup
- PhysicalChannel via PncMapping.physicalChannel

⌋

Each PncMapping has the ability to define relations to ISignalIPduGroups, PdurIPduGroups, ConsumedProvidedServiceInstanceGroups, and PhysicalChannels. These relations are used to describe the impact of a PNC on the respective networking artifacts. The realization of those relationships is typically to be implemented in the Basic Software Mode Manager (BswM).

- PncMapping.pncGroup (referring to ISignalIPduGroup) is used to define which Com ISignalIPduGroups shall be enabled when this PNC is active. The use-case is to enable/disable Com ISignalIPdus, especially periodic behavior like cyclic sending and time-out monitoring.
- PncMapping.pncPdurGroup (referring to PdurIPduGroup) is used to define which PdurIPduGroups shall be enabled when this PNC is active. The use-case is to enable/disable Pdus which typically are not passing the Com module (Diagnostic IPdus, Gatewayed IPdus).
- PncMapping.pncConsumedProvidedServiceInstanceGroup (referring to ConsumedProvidedServiceInstanceGroup) is used to define which ConsumedProvidedServiceInstanceGroups shall be enabled when this PNC is active. The use-case is to define which ProvidedServiceInstances and ConsumedServiceInstances are active when this PNC is active.
- PncMapping.physicalChannel (referring to PhysicalChannel) is used to define which PhysicalChannels shall be enabled when this PNC is active. The use-case is to enable/disable the whole PhysicalChannel and allows to also cover Pdus which neither pass the Com module nor the Pdu Router module (e.g. NM).

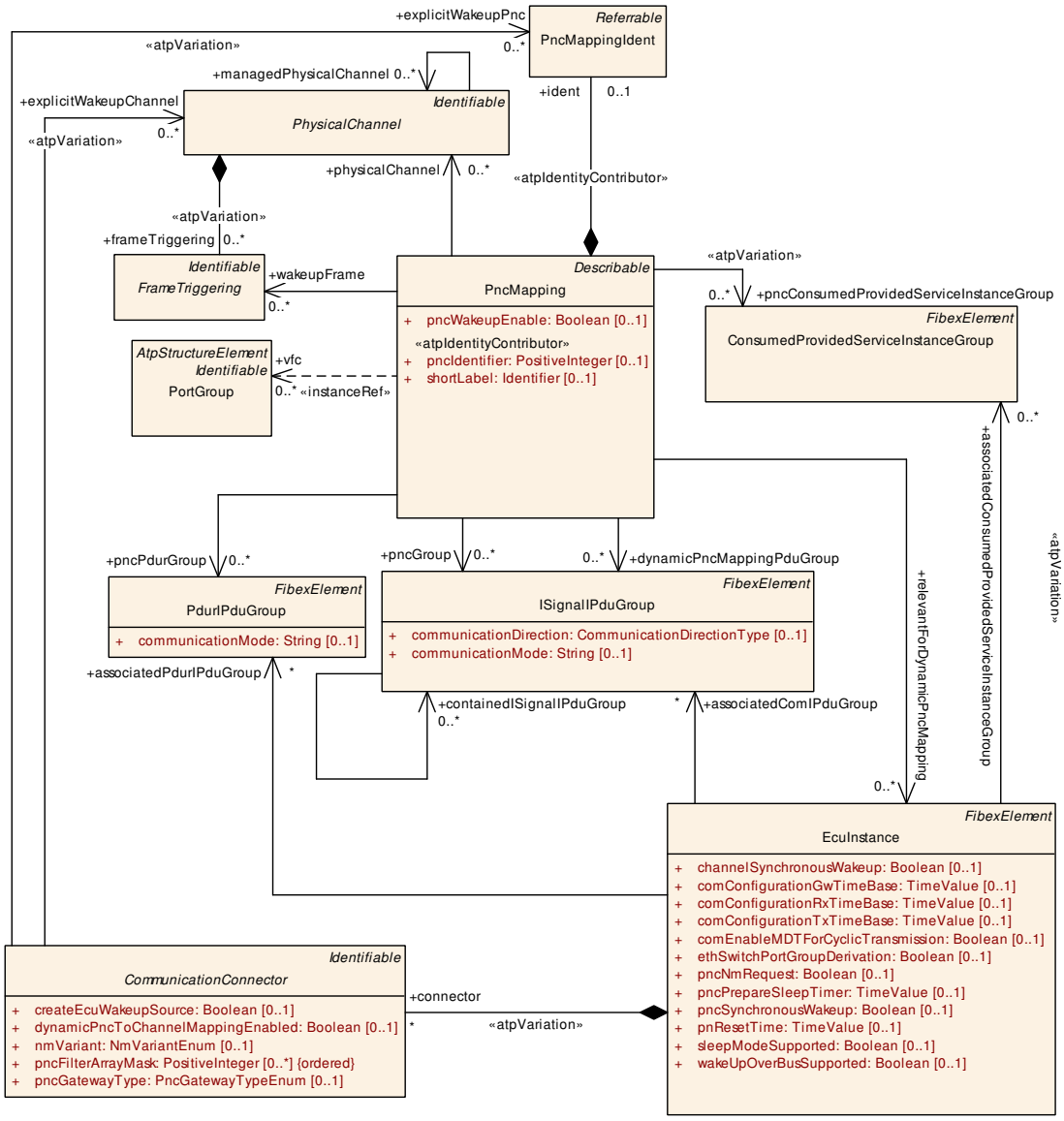

Figure 5.39: Mapping of Virtual Function Clusters onto Partial Network Clusters

Class	**PncMapping**			
Note	Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.			
Base	*ARObject*, *Describable*			
Aggregated by	SystemMapping.pncMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	**PncMapping**			
dynamicPnc MappingPdu Group	ISignalIPduGroup	*	ref	Reference to an ISignalIPduGroup that allows mapping of this PNC without statically mapping this PNC directly to a channel. This is needed to describe dynamic PNCs that can be learned only at run-time and which have also a relation to an ISignalIPduGroup. **Tags:** atp.Splitkey=dynamicPncMappingPduGroup atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
ident	PncMappingIdent	0..1	aggr	This adds the ability to become referrable to PncMapping. **Stereotypes:** atpIdentityContributor
physical Channel	PhysicalChannel	*	ref	This reference maps the partial network to a communication channel. **Tags:** atp.Splitkey=physicalChannel
pncConsumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	ConsumedProvidedServiceInstanceGroup used in a Partial Network Cluster. This reference is optional, since this could be used for starting and stopping Consumed ProvidedServiceInstanceGroup according the requested partial network, but is not necessarily needed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pncConsumedProvidedServiceInstance Group, pncConsumedProvidedServiceInstance Group.consumedProvidedServiceInstanceGroup, pnc ConsumedProvidedServiceInstanceGroup.variation Point.shortLabel vh.latestBindingTime=postBuild
pncGroup	ISignalIPduGroup	*	ref	IPduGroup participating in a Partial Network Cluster. This reference is optional in case an ecu extract has only indirect pnc access, i.e. ecu is not directly connected to a network which supports partial network. **Tags:** atp.Splitkey=pncGroup This Attribute is only used by the AUTOSAR Classic Platform.
pncIdentifier	PositiveInteger	0..1	attr	Identifer of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu. **Stereotypes:** atpIdentityContributor
pncPdurGroup	PdurIPduGroup	*	ref	This reference maps the Partial Network Cluster to a set of PdurIpduGroups. **Tags:** atp.Splitkey=pncPdurGroup This Attribute is only used by the AUTOSAR Classic Platform.
pncWakeup Enable	Boolean	0..1	attr	If this parameter is available and set to true then this PNC will be woken up as soon as a channel wakeup occurs on a channel where this PNC is assigned to. This is ensured by adding this PNC to the corresponding channel wakeup sources during upstream mapping. **Tags:** atp.Status=obsolete

▽

△

Class	**PncMapping**			
relevantFor DynamicPnc Mapping	EcuInstance	*	ref	Reference to a PNC Gateway ECU for PNCs which do not have a static channel mapping. This is needed to describe dynamic PNCs that can be learned only at run-time and which have no relation to an ISignalIPdu Group. **Tags:** atp.Splitkey=relevantForDynamicPncMapping atp.Status=draft This Attribute is only used by the AUTOSAR Classic Platform.
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the PncMapping. It shall be unique in the System scope. **Stereotypes:** atpIdentityContributor
vfc	PortGroup	*	iref	Virtual Function Cluster to be mapped onto a Partial Network Cluster. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems. **Tags:** atp.Splitkey=vfc, vfc.contextComposition, vfc.contextComponent, vfc.target **InstanceRef implemented by:** PortGroupInSystem InstanceRef
wakeupFrame	FrameTriggering	*	ref	Reference to collection of FrameTriggerings that are used for the wakeup of this PNC (Application Frames or Nm Frames can be used). This reference is only valid if this EcuExtract represents an ECU which has direct PNC access, i.e. ECU is directly connected to a network which supports partial network. **Tags:** atp.Splitkey=wakeupFrame This Attribute is only used by the AUTOSAR Classic Platform.

Table 5.43: PncMapping

[constr_5479] Existence of `PncMapping.pncIdentifier`

Imposition time: CP: `IT_SysDesc`

⌈For each `PncMapping`, the attribute `pncIdentifier` shall exist.⌋

[constr_3039] `pncIdentifier` range

Imposition time: CP: `IT_SysDesc`

⌈The `pncIdentifier` value shall be in the range of 8..63 for normal CAN and in the range of 8..511 for CAN FD, FlexRay and Ethernet.⌋

Note: Older ECUs with a smaller PNC range can be used together with new ECUs with a larger PNC range (e.g. when NM PDU length and PNC Range increased in the next model year and some ECUs are overtaken). But take care that in that case the older ECUs cannot react on PNC identifiers that are beyond their smaller range as they will be ignored by those older ECUs.

[TPS_SYST_03067] Definition of `pncVectorOffset` ⌈The attribute `System.pncVectorOffset` shall define the common start byte position of the PNC Request Bit Vector for all NM messages in the `System`.⌋

[TPS_SYST_03068] Definition of pncVectorLength ⌈The attribute System.pncVectorLength shall define the maximum PNC Request Bit Vector length from all NM messages of the System.⌋

Partial Networks are considered System wide. If a specific PhysicalChannel has a limited payload size and thus is not able to transport the entire PNC Request Bit Vector information in the payload of the NM messages it is possible to define a shortened PNC Request Bit Vector on that specific NmCluster (where that NmCluster configures the NM behavior on that CommunicationCluster or in case of Ethernet the PhysicalChannel representing a VLAN).

[constr_3687] Limited value range for NmCluster.pncClusterVectorLength

Imposition time: CP: IT_SysDesc

⌈The value of NmCluster.pncClusterVectorLength shall be equal or smaller than System.pncVectorLength.⌋

[constr_3198] Uniqueness of PncMapping.shortLabel

Imposition time: CP: IT_SysDesc

⌈If the optional shortLabel attribute is used it shall be unique in the System scope.⌋

The runtime information that is used to coordinate the request/release information of all partial networks is called PNC Request Bit Vector. The size and position of the PNC Request Bit Vector inside the NmPdu is globally defined in the System class in chapter 1.3. The size might be reduced using the optional NmCluster.pncClusterVectorLength attribute.

In the system description the NmPdus are described based on the actual network interaction (i.e. an ECU sends one NmPdu per network and receives a set of NmPdus).

NmPdus that define the existence of NM user data via the existence of the attribute iSignalToIPduMapping shall be referenced by corresponding PduTriggerings where the attribute iPduPort exists accordingly. This is also reflected by [TPS_SYST_01057].

NmPdus that define the existence of NM user data via the definition of NmPdu.nmDataInformation shall not be referenced by PduTriggerings because neither Com nor PduR are involved in the transmission (which lets the Nm module talk to the bus interface directly).

Please note that a PNC Request Bit Vector is transmitted as byte array, where each bit of the byte array represents one particular partial network cluster. The AUTOSAR stack handles a PNC Request Bit Vector as byte array data type.

[constr_3040] Restriction of pncIdentifier values

Imposition time: CP: IT_SysDesc

⌈The pncIdentifier value shall be within the range described by pncVectorOffset and pncVectorLength.⌋

[constr_3146] Partial Networking timing constraint

Imposition time: CP: IT_SysDesc

⌈For Partial Networking the following timing constraints shall be ensured:

- CAN / Ethernet: (pnResetTime + pncPrepareSleepTimer) < nmNetwork-Timeout
- FlexRay: (pnResetTime + pncPrepareSleepTimer) < nmReadySleepTime

⌋

[TPS_SYST_02145] Default behavior for not defined nmPncParticipation
⌈When NmCluster.nmPncParticipation is set to *true* or is not defined this Nm-Cluster shall contribute to the partial network mechanism.⌋

[constr_3323] Relation between NmCluster.nmPncParticipation and Pnc-Mapping.pncGroup

Imposition time: CP: IT_SysDesc

⌈If a PncMapping references an ISignalIPduGroup in role pncGroup which in turn

- contains (either directly or via one of its subordinate ISignalIPduGroups referenced in role containedISignalIPduGroup) ISignalIPdus that are referenced by a PduTriggering in role iPdu which in turn
- is composed by a PhysicalChannel in role pduTriggering which in turn
- is composed by CommunicationCluster in role physicalChannel which in turn
- is referenced by an NmCluster in role communicationCluster,

then this NmCluster shall have its nmPncParticipation attribute set to TRUE unless the PhysicalChannel is referenced in the role managedPhysicalChannel.⌋

[constr_3484] PncMapping that refers a managedPhysicalChannel shall also refer the managing PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈If a PncMapping refers to a PhysicalChannel (either directly in the role physicalChannel or indirectly by referencing an ISignalIPduGroup in the role pncGroup) and this PhysicalChannel is referenced in the role managed-PhysicalChannel, then the according managing PhysicalChannel (the source of the managedPhysicalChannel reference) shall also be referenced by the Pnc-Mapping (either directly in the role physicalChannel or indirectly by referencing an ISignalIPduGroup in the role pncGroup).⌋

Note that [constr_3484] ensures that the managing PhysicalChannel is part of the same PNC as the managedPhysicalChannels.

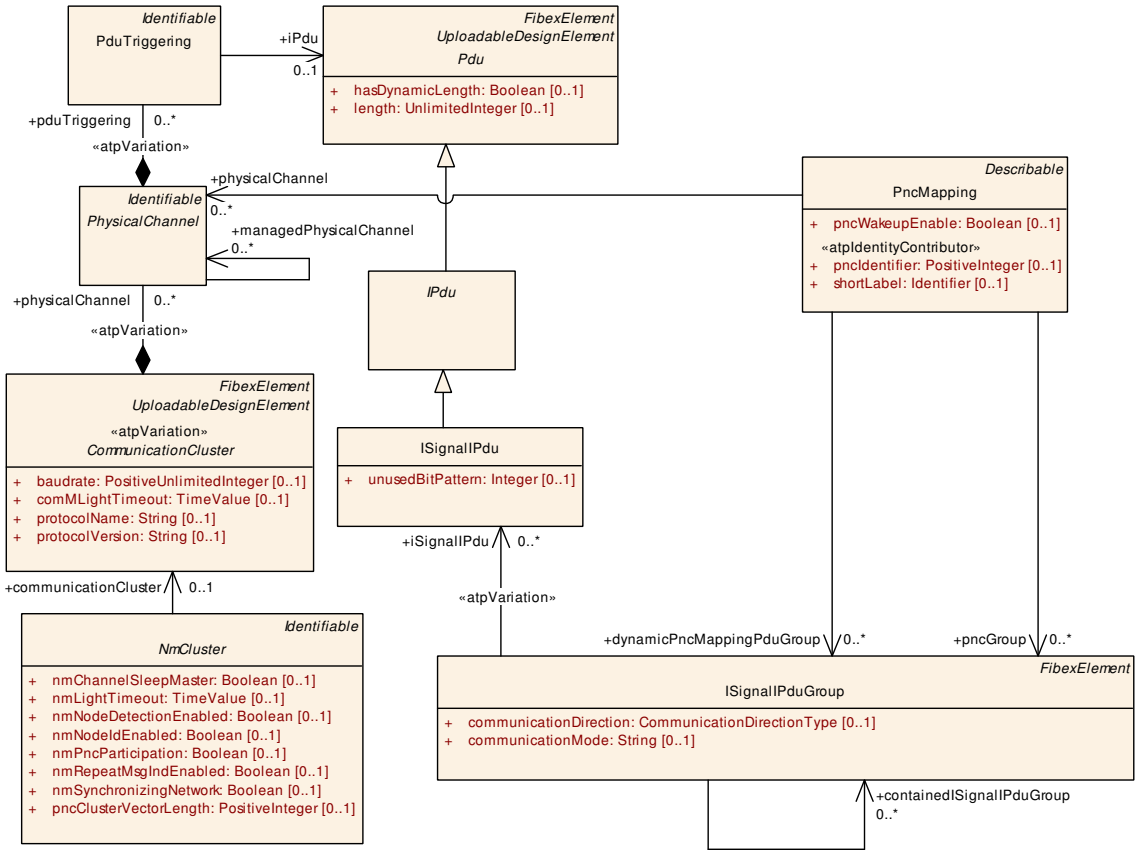

Figure 5.40: Relation between `NmCluster.nmPncParticipation` and `PncMapping.pncGroup`

[TPS_SYST_02146] Explicit definition of `PNC Request Bit Vector` at `NmPdu` ⌈If there is an `ISignalToIPduMapping` aggregated by `NmPdu` that fully matches the interval defined by `pncVectorOffset` and `pncVectorLength` then the corresponding `ISignal` represents the `PNC Request Bit Vector`.⌋

Attributes used to configure the Partial Network Wakeup of one specific Ecu are described in chapter 3.3.1.4.

[TPS_SYST_03073] Derivation of NmPnFilterMaskByte

Status: DRAFT

⌈The `CommunicationConnector.pncFilterArrayMask` is configured per `CommunicationConnector`. This data mask is covering the PNC Request Bit Vector payload of the `NmPdu`. Therefore `pncVectorLength` needs to be considered when deriving the NmPnFilterMaskByte configuration. Also the order of `pncFilterArrayMask` shall be preserved during derivation.⌋

Example: For `pncFilterArrayMask` = `[0x00, 0x00, 0x00, 0x00, 0x00, 0x80]` and `pncVectorOffset` = 2, `pncIdentifier` 63 in a `NmPdu` will be masked (see Figure 5.41).

NmPdu	Byte 0								Byte 1								Byte 2								Byte 3							
Absolute bit position	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
PNC identifiers				N	O	T					U	S	E	D			23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

NmPdu	Byte 4								Byte 5								Byte 6								Byte 7							
Absolute bit position	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
PNC identifiers	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56

pncFilterArrayMask	NOT								USED								Byte 0								Byte 1							
Mask bit position																	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
PNC identifiers				N	O	T					U	S	E	D			23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

pncFilterArrayMask	Byte 2								Byte 3								Byte 4								Byte 5							
Mask bit position	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
PNC identifiers	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56

Figure 5.41: Example of masked pncIdentifiers in a NmPdu

Note that only pncIdentifier =< 63 can be used for the Filter mask, even if NM PDUs are configured to be larger than 8 Byte and the PNC Request Bit Vector defined in NM user data exceeds the first 8 Byte.

5.5.1.1 Partial Networking and managed Ethernet switch

On switched Ethernet networks it is possible to let the Ethernet switch be managed by an AUTOSAR ECU. In this case the configuration and the behavior of the CouplingElement with couplingType=switch are controlled by the host ECU.

For the usage of Partial Networking on switched Ethernet networks with managed CouplingElements the following use-case applies:

Depending on the requested Partial Networks it is possible to switch off Ethernet switch ports (CouplingPorts) which are not involved in any currently active communication ([TPS_SYST_03055]). The time delay to switch off the CouplingPorts shall be in line with the used Network Management timing to avoid switching off the CouplingPorts before the Network Management has decided to go to sleep ([constr_3616]).

[TPS_SYST_03055] Semantics of EthernetCluster.couplingPortSwitchoffDelay ⌈EthernetCluster.couplingPortSwitchoffDelay, if defined, is used to configure a host ECU to delay a switch off of a CouplingPort.⌋

If using Partial Networks and the possibility to switch off Ethernet switch ports (CouplingPorts), it is possible to switch on CouplingPorts for a certain amount of time, as defined in EthernetCluster.couplingPortStartupActiveTime ([TPS_SYST_03054]). This is used to enable the host ECU (ECU that maintains an Ethernet switch) to listen to the network and wait for couplingPortStartupActiveTime to receive Partial Network information. If Partial Network information is received within couplingPortStartupActiveTime, then the according PNCs and the Ethernet switch will be requested. With this Partial Network related request the controlling of the Ethernet switch ports will be done as described in [TPS_SYST_03055]). The remaining Ethernet switch ports will be switched off after couplingPortStartupActiveTime expires ([TPS_SYST_03054])).

[TPS_SYST_03054] Semantics of EthernetCluster.couplingPortStartupActiveTime ⌈EthernetCluster.couplingPortStartupActiveTime, if defined, is used to configure a host ECU to detect a wake-up and wait to receive a Partial Network information. This also applies if a host ECU detects a reset as wake-up reason and listens to the network if Partial Network information is available on that network.⌋

In order to describe the relationships between Partial Networks and Ethernet switch ports an optional reference from CouplingPort to a PncMappingIdent in the role pncMapping is defined.

[constr_3615] Existence of EthernetCluster.couplingPortSwitchoffDelay

Imposition time: CP: IT_SysDesc

⌈The attribute EthernetCluster.couplingPortSwitchoffDelay shall be defined if at least one EcuInstance connected to that EthernetCluster has the attribute ethSwitchPortGroupDerivation set to TRUE.⌋

[constr_3616] Value of EthernetCluster.couplingPortSwitchoffDelay

Imposition time: CP: IT_SysDesc

⌈If defined, the value of EthernetCluster.couplingPortSwitchoffDelay shall be greater than UdpNmCluster.nmNetworkTimeout + UdpNmCluster.nmWaitBusSleepTime of the respective EthernetCluster.⌋

[constr_3617] Existence of EthernetCluster.couplingPortStartupActiveTime

Imposition time: CP: IT_SysDesc

⌈The attribute EthernetCluster.couplingPortStartupActiveTime shall be defined if at least one EcuInstance connected to that EthernetCluster has the attribute ethSwitchPortGroupDerivation set to TRUE.⌋

[constr_3618] Value of EthernetCluster.couplingPortStartupActiveTime

Imposition time: CP: IT_SysDesc

⌈If defined, the value of EthernetCluster.couplingPortStartupActiveTime shall be greater than UdpNmCluster.nmNetworkTimeout + UdpNmCluster.nmWaitBusSleepTime of the respective EthernetCluster.⌋

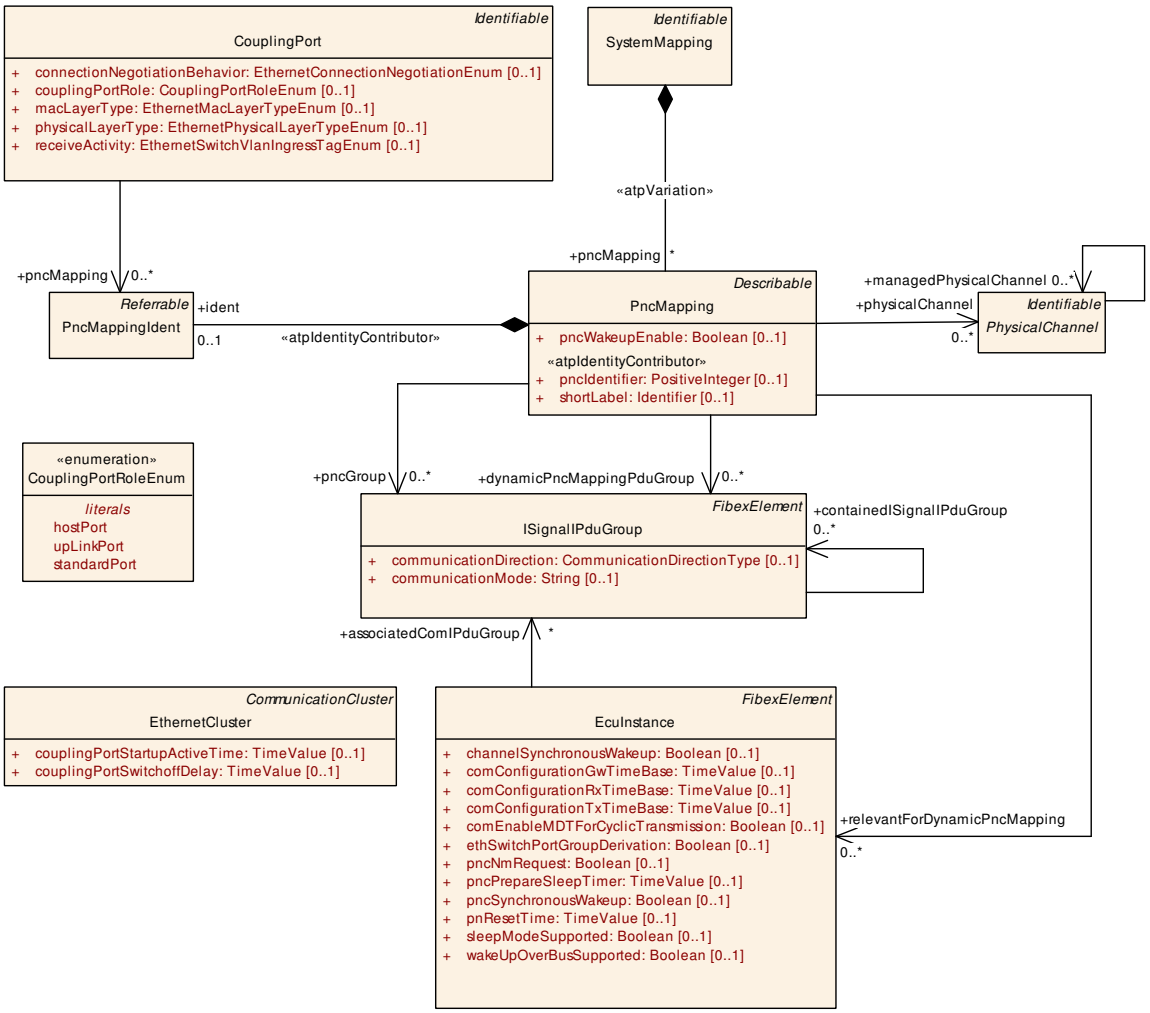

Figure 5.42: Partial Networking and managed Ethernet switch

The example in figure 5.43 illustrates the setup of an host ECU which manages 2 Ethernet switches.

The port `1` of `Switch 1` is a `hostPort`.

The port `5` of `Switch 1` and port `1` of `Switch 2` are `upLinkPort`s.

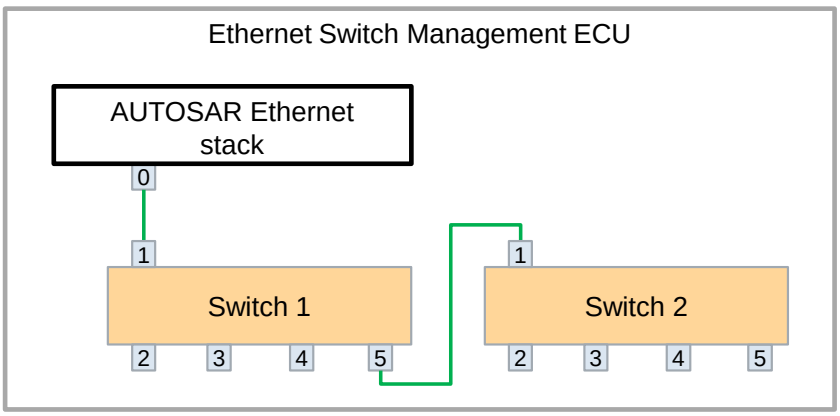

Figure 5.43: Example of managed Ethernet switches

[constr_3523] `CouplingPort` and `PncMapping` in the scope of an `EthernetPhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈If

- a `CouplingPort` referring to an `EthernetPhysicalChannel` – via a `VlanMembership` – references at least one `PncMapping`
- and that `PncMapping` contains PDUs – via the assignment of `PncMapping.pncGroup` – that are transported on this `EthernetPhysicalChannel`

then every `CouplingPort` referring to that `EthernetPhysicalChannel` shall reference at least one `PncMapping` as well.⌋

If a `CouplingPort` referring to an `EthernetPhysicalChannel` – via a `VlanMembership` – references no `PncMapping`, then any other `CouplingPort` referring to that `EthernetPhysicalChannel` shall either

- not reference any `PncMapping` or
- reference only `PncMapping`s which have no relation (according to [TPS_SYST_03080]) to that `EthernetPhysicalChannel`.

The rational for [constr_3523] is that all `CouplingPort`s aggregated by a `CouplingElement` are derived to `EthSwtPorts` of an Ethernet switch in the ECU configuration. `EthSwtPorts` are controlled exclusively either via VLAN membership or via PNC membership.

If `EthSwtPorts` are controlled via VLAN membership, then the `EthSwtPorts` are activated as soon as the first relevant VLAN is activated and deactivated as soon as the last VLAN, where this `EthSwtPort` belongs to, is de-activated.

If `EthSwtPorts` are controlled via PNC membership, then the `EthSwtPorts` are activated as soon as the first relevant PNC is activated and deactivated as soon as the last PNC, where this `EthSwtPort` belongs to, is de-activated.

The PNC membership is modeled upon VLAN membership and could span across multiple VLANs. Thus, the modelling of PNC membership for `CouplingPort`s need to respect the VLAN membership, since the `EthSwtPorts` are controlled exclusively either via the PNC membership or the VLAN membership.

If a PNC is requested and the `EthSwtPorts` are controlled via PNC membership, then all VLANs and `EthSwtPorts`, which are associated with the PNC membership, are activated. As soon as VLAN membership and PNC membership have a relation, VLANs are controlled via a PNC request. The modeling needs to ensure per VLAN, that either all referencing `CouplingPort`s have no reference to a PNC mapping or the `CouplingPort`s referring different PNC mappings which have no relation to the affected VLANs. This ensures the mutual exclusive control of `EthSwtPorts` either via VLAN membership or PNC membership.

[TPS_SYST_03018] Aggregation of PNCs at the `hostPort` ⌈A `CouplingPort` with `couplingPortRole` set to `hostPort` shall reference all `PncMapping`s that are referenced by any `CouplingPort`s of the same `CouplingElement` and all `CouplingElement`s connected to this `CouplingElement`.⌋

[constr_3524] Definition of `couplingPortRole` on `CouplingPort` for managed `CouplingElement`

Imposition time: CP: `IT_SysDesc`

⌈A managed `CouplingElement` shall have either

- at most one `CouplingPort` with `couplingPortRole` set to `hostPort` or
- at least one `CouplingPort` with `couplingPortRole` set to `upLinkPort`.

⌋

[constr_3525] Connection of `CouplingPort` with `couplingPortRole` set to `upLinkPort`

Imposition time: CP: `IT_SysDesc`

⌈A `CouplingPort` with `couplingPortRole` set to `upLinkPort` shall be connected to exactly one other `CouplingPort` with `couplingPortRole` set to `upLinkPort`.⌋

[TPS_SYST_03020] Default value for `CouplingPort.couplingPortRole` if not defined ⌈If no value for the attribute `CouplingPort.couplingPortRole` is defined then `standardPort` shall be assumed.⌋

[TPS_SYST_03019] Modeling of `CouplingPort`s for managed `CouplingElement` ⌈Only `CouplingPort`s that participate in the communication of a managed `CouplingElement` shall be modeled in the System Description.⌋

All other ports of an Ethernet switch are not modeled. The expected behavior of unmodeled Ethernet switch ports on runtime:

1. the Ethernet switch driver switches off this Ethernet switch ports during its initialization
2. unmodeled Ethernet switch ports shall never be switched on.

5.5.1.2 Partial Network Gateway

As Partial Networks are spread over multiple networks their state has to be synchronized by gateway ECUs connected to a Partial Network on more than one network. Therefore a gateway mechanism is available where per network it can be configured if the network takes part in Partial network gateway algorithm or not. When more than one gateway is connected to one network then only one gateway shall be configured as main gateway.

[constr_5093] `pncGatewayType` and `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈When multiple `CommunicationConnector`s with `pncGatewayType` set to a value other than `none` are referenced by the same `PhysicalChannel` then only up to one `CommunicationConnector` shall have the `pncGatewayType` set to `active`.⌋

An ECU having a Partial Network connected to more than one network may not necessarily coordinate this Partial Network. There can be uses-cases where such an ECU is connected to multiple networks where this Partial Network is relevant but all of these networks are handled by the same gateway or by various gateways connected on another network. In this case the ECU does not coordinate the Partial Network.

[constr_5094] `pncGatewayType` and ECU

Imposition time: CP: `IT_SysDesc`

⌈When an ECU is connected to more than one `PhysicalChannel` and has a relation to a Partial Network then all `CommunicationConnector`s of this ECU where this Partial Network is related to shall have the `pncGatewayType` value either set to `none` or to a value different than none (i.e. `active` or `passive`).⌋

Hint: This constraint ensures that an ECU either coordinates a Partial Network on all of its connected channels or on none.

5.5.1.3 Checking the consistency of partial network clusters (PNC)

The gateway ECUs participating in the synchronization of partial network requests are called PNC-Coordinators. Considering the system nature of PNC-Coordinators an example setup might look like sketched in figure 5.44.

It is essential to define to relation of particular PNCs and `PhysicalChannel`s as well as the relation of PNCs and `EcuInstance`s.

[TPS_SYST_03080] `PhysicalChannel` involved in a particular PNC ⌈The involvement of a `PhysicalChannel` in a particular PNC can be defined in several ways:

- direct reference from `PncMapping` to `PhysicalChannel` in the role `physicalChannel`
- `PncMapping` refers to a `PdurIPduGroup` in the role `pncPdurGroup` where that `PdurIPduGroup` refers to at least one `PduTriggering` owned by the `PhysicalChannel`
- `PncMapping` refers to a `ISignalIPduGroup` in the role `pncGroup` where that `ISignalIPduGroup` (or any `ISignalIPduGroup.containedISignalIPduGroup` of that `ISignalIPduGroup`) refers to an `ISignalIPdu` and that `ISignalIPdu` is referenced by a `PduTriggering` owned by the `PhysicalChannel`

- PncMapping refers to a ConsumedProvidedServiceInstanceGroup in the role pncConsumedProvidedServiceInstanceGroup where that ConsumedProvidedServiceInstanceGroup in turn refers to a ProvidedServiceInstance or ConsumedServiceInstance and these refer to a SoConIPduIdentifier that refers to a PduTriggering owned by the PhysicalChannel

⌋

Any of the above listed cases can overlap, thus a PhysicalChannel may be directly referenced and additionally there may exist a reference to a qualified ISignalIPduGroup.

[TPS_SYST_03081] EcuInstance involved in a particular PNC ⌈An EcuInstance is involved in a PNC if at least one PhysicalChannel is involved in that PNC according to [TPS_SYST_03080] and that PhysicalChannel refers to a CommunicationConnector in the role commConnector and that CommunicationConnector is aggregated at the EcuInstance in the role connector.⌋

Some observations and assumptions on the example given in figure 5.44: Each ECU connected to more than one PhysicalChannel may be configured as a PNC-Coordinator. In the sketched example the PNC-Coordinators are *PNC-Coor1*, *PNC-Coor2*, and *PNC-Coor3*. *PNC-Coor1* and *PNC-Coor3* coordinate *PNC1* and *PNC3*. *PNC-Coor2* coordinates only *PNC1*. *PNC-Coor3* coordinates *PNC1* on channels *Ch1*, *Ch2*, and *Ch3*. *PNC-Coor3* additionally coordinates *PNC3* on channels *Ch1* and *Ch2*.

A node which participates in a particular PNC can request this PNC and this node is most likely able to receive a PNC request when this PNC has been requested remotely by another node. In exceptional cases a node may be configured to request a particular PNC, but not receive a PNC request of the same PNC initiated remotely by another node. This is depicted in figure 5.44. *Node4* has the possibility to request *PNC3*. But a request of *PNC3* received by *PNC-Coor3* is not forward to *PNC-Coor3* on *Ch3*. And since *PNC-Coord2* does not handle *PNC3*, *Node4* will never receive a NM message where *PNC3* is requested. This could be used by a node to remotely request a PNC, but do not contribute to this PNC (see also section 5.5.1.3.3).

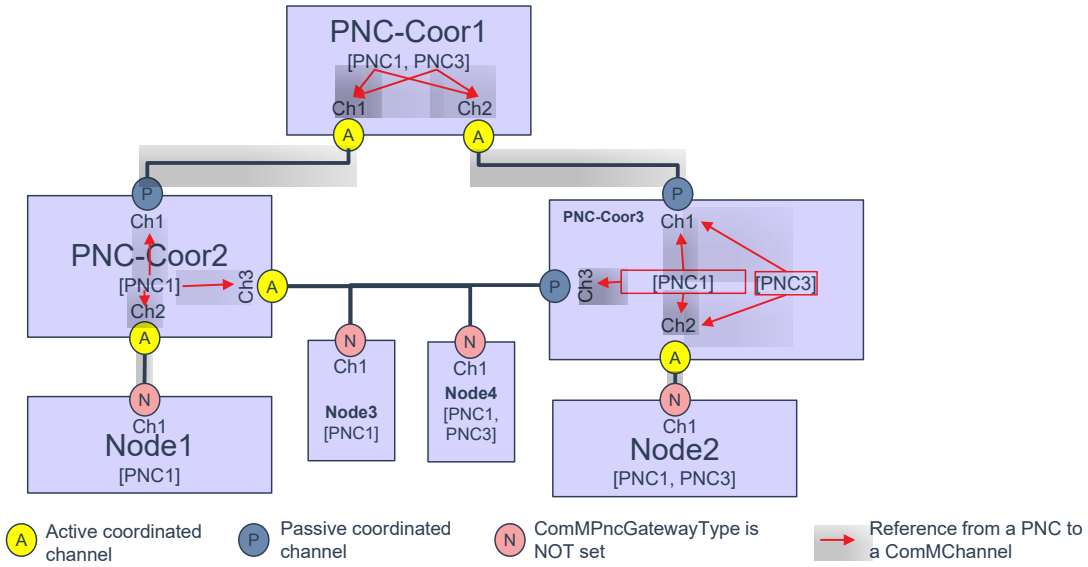

Figure 5.44: PNC topology example

5.5.1.3.1 Check for top level PNC-Coordinator

The definition of a top level PNC-Coordinator is essential for the check on cyclic PNC topologies. The cycle check and the top level PNC-Coordinator are considered per PNC:

For all nodes which have a specific PNC *X* configured, identify the nodes which have PNC coordination configured, with the PNC *X* only referencing channels with `CommunicationConnector.pncGatewayType` set to `active`.

Approach to check:

Create a list of which `EcuInstance`s a PNC *X* is assigned to, called "PNC to ECU mapping" by checking which PNCs are configured for each `EcuInstance` in the `System`.

Example of "PNC to ECU mapping" for figure 5.44:

- PNC1: {PNC-Coor1, PNC-Coor2, Node1, Node3, Node4, PNC-Coor3, Node2}
- PNC3: {PNC-Coor1, Node4, PNC-Coor3, Node2}

Identify the `EcuInstance`s which have PNC coordination enabled and PNC *X* only references channels with `CommunicationConnector.pncGatewayType` set to `active`.

In the example given in figure 5.44 the ECU *PNC-Coor1* is the top level PNC-Coordinator for both, *PNC1* and *PNC3*. But there may exist configurations where different ECUs are top level PNC-Coordinators for different PNCs or even multiple top level PNC-Coordinators for the same PNC. Please note, multiple top level PNC coordinators for the same PNC is only supported, if [constr_3714] is fulfilled.

[constr_3781] Each PNC assigned to multiple PhysicalChannels shall have a top level PNC-Coordinator

Imposition time: CP: IT_SysDesc

⌈In a System, if a PNC is assigned to multiple PhysicalChannels according to [TPS_SYST_03080], then this PNC shall have at least one top level PNC-Coordinator according to [TPS_SYST_03082].⌋

For simple setups of partial networks, where PNCs are not distributed across multiple PhysicalChannels, no PNC-Coordinator is required.

[TPS_SYST_03082] Definition of top level PNC-Coordinator ⌈An EcuInstance is defined as top level PNC-Coordinator for a specific PNC if all involved PhysicalChannels (according to [TPS_SYST_03080]) only refer to CommunicationConnectors aggregated at that EcuInstance, where CommunicationConnector.pncGatewayType is set to active.⌋

[constr_3714] Multiple top level PNC-coordinators shall be allowed

Imposition time: CP: IT_SysDesc

⌈Multiple top level PNC-coordinators shall only be allowed if no network path across all networks exist that connects a CommunicationConnector with pncGatewayType PncGatewayTypeEnum.active to another CommunicationConnector with pncGatewayType PncGatewayTypeEnum.active where both CommunicationConnectors belong to different top level PNC-coordinators.⌋

5.5.1.3.2 Check for cyclic PNC-Coordinator topology

The topology of a partial network cluster (PNC) needs to be cycle-free, that is, the tree spanning the PNC-Coordination needs to be an acyclic directed graph (ADG). Cycles in the graph could make releasing PNC requests impossible (deadlock). Having opposing directions of edges in the graph could cause a split in the PNC network.

The forwarding of PNC requests within an ECU is performed as follows:

- PNC requests received on a CommunicationConnector (channel) with CommunicationConnector.pncGatewayType set to active are forwarded to all CommunicationConnectors where this PNC is assigned to (according to [TPS_SYST_03080]) (this also includes the CommunicationConnector where the PNC request was received)
- PNC requests received on a CommunicationConnector (channel) with CommunicationConnector.pncGatewayType set to passive are forwarded to all CommunicationConnectors where this PNC is assigned to (according to [TPS_SYST_03080]) and the CommunicationConnector.pncGatewayType is set to active
- in all other cases the PNC requests are NOT forwarded

Approach to check for cycles in PNC topology, needs to be checked for each PNC individually:

[TPS_SYST_03083] Creation of a *PNC paths tree* ⌈Beginning with the top level PNC-Coordinator of a specific PNC *X* ([TPS_SYST_03082]), iterate "depth first" over all channels where PNC *X* is assigned to (at each node reached only follow network links which have `CommunicationConnector.pncGatewayType` set to `active`), and build a tree of nodes reachable by the channels (PNC paths tree).⌋

By this, a tree of paths in the system is built. In case a node can reach itself again in a network subtree of its own channels, a cycle is detected and the topology is invalid.

Example PNC paths tree for figure 5.44:

PNC1:
- **PNC-Coor1**
  - Ch1 → *PNC-Coor2*
    - Ch2 → *Node1*
    - Ch3 → *Node3*
      - *Node4*
      - *PNC-Coor3*
        - Ch2 → *Node2*
  - Ch2 → *PNC-Coor3*
    - Ch2 → *Node2*

PNC3:
- ***PNC-Coor1***
  - Ch2 → *PNC-Coor3*
    - Ch2 → *Node2*

Figure 5.45: PNC Coordination tree example

In the example topology of figure 5.44 *PNC-Coor3* can be reached from top level PNC-Coordinator *PNC-Coor1* directly and via *PNC-Coor2* on different paths, but it cannot reach itself, so this is no issue (cycle).

[TPS_SYST_03084] Acyclic PNC graph definition ⌈For each PNC, the communication relations to aggregate and distribute the system's PNC state between all nodes, including PNC-Coordinators, shall form a topology reflecting a directed, acyclic graph (according to [TPS_SYST_03083]).⌋

The PNC topology depicted in figure 5.46 is an example where a cyclic dependency is defined.

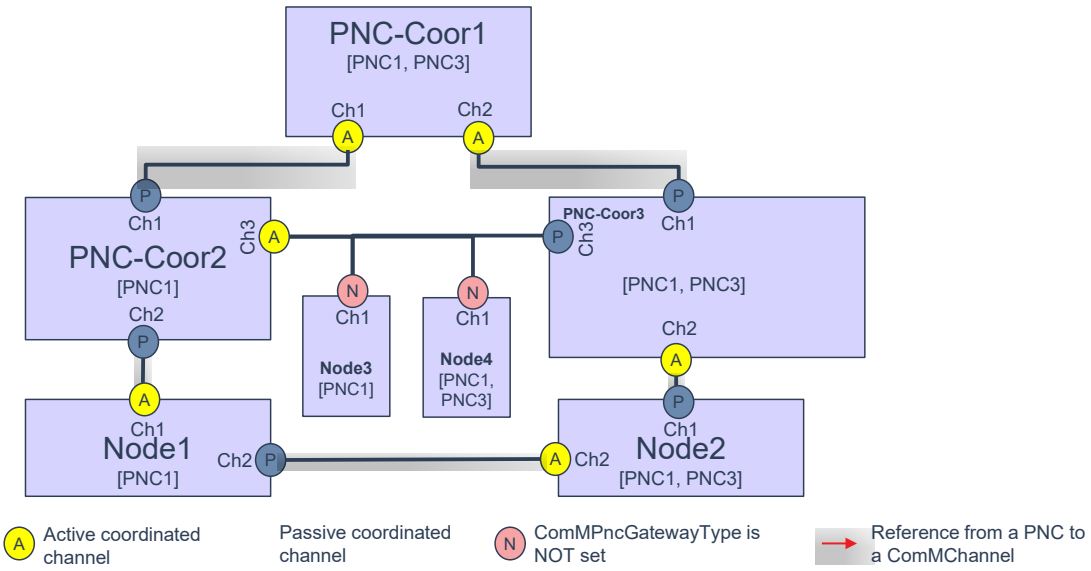

Figure 5.46: PNC topology example with cyclic graph error

Some constraints are fulfilled by this example, e.g. there in only one top level PNC-Coordinator defined (*PNC-Coor1*) according to [constr_3714], also [constr_5093] is fulfilled.

An issue arises with the cyclic definition of *PNC1* coordination between *PNC-Coor2*, *PNC-Coor3*, *Node2*, and *Node1*. This topology is invalid according to [TPS_SYST_03084].

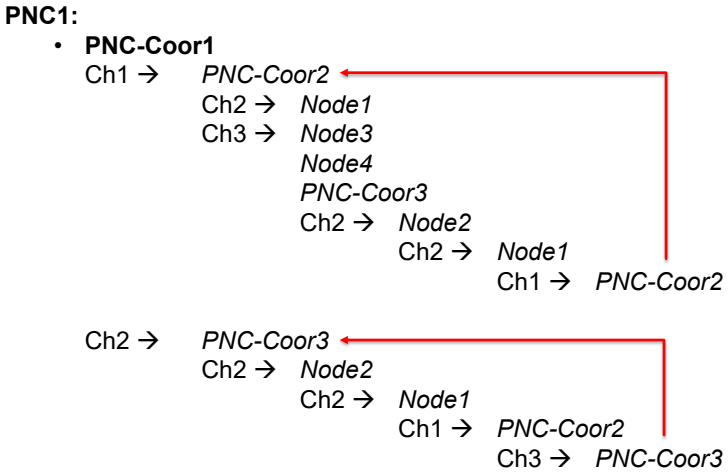

Figure 5.47: PNC Coordination tree example with cycle

5.5.1.3.3 Optional check if all nodes are reachable from PNC-Coordinators

An additional check may compare if all nodes in the "`EcuInstance` interacts with PNC" list ([TPS_SYST_03081]) can be found in the "PNC paths tree" ([TPS_SYST_03083]). For each node which can be found in the "`EcuInstance` interacts with PNC",

but not the "PNC paths tree", a warning shall be issued that this node can request a PNC, but can not be requested by that PNC.

The example in figure 5.44 illustrates that *Node4* is issuing PNC requests on *PNC3*, however in the "PNC paths tree" shown in figure 5.45 *Node4* does not occur in the tree for *PNC3*, thus *Node4* will not be requested by *PNC3*.

[advisory_02004] Check for reachable PNC nodes

Imposition time: CP: `IT_SysDesc`

⌈If an `EcuInstance` is part of the "`EcuInstance` interacts with PNC" list ([TPS_SYST_03081]) for a particular PNC then this `EcuInstance` shall also appear in the "PNC paths tree" ([TPS_SYST_03083]) for that PNC.⌋

5.5.1.4 Dynamic Partial Network Cluster Mapping

In order to reduce the coding/adaptions-variants because of the wakeup-sleep-behavior (i.e. Function that transmits and receives PDUs mapped to PNCs can be deactivated/activated, placed on different ECUs or even run on Adaptive-AUTOSAR partitions) the PNC-to-channel-mapping table of the PNC Gateways can be dynamically reconfigured.

For this functionality, two parts of the PNC-to-channel-mapping need to exist:

- statically configured via SystemDescription and stored in ROM. This is used to define PNC routing for critical system communication (i.e. keep bus communication alive during hazard lights). This is also the default behavior when dynamic PNC-to-channel-mapping is not used or before this dynamic handling was introduced.
- configurable to be changed during runtime, stored in NVRAM. The Entries can be read and set (never unset) by a learning algorithm and via APIs. The table can also be reset by API, but it contains at least the entries out of the statically configured table.

	PN-Cluster 1	PN-Cluster 2	PN-Cluster 3	PN-Cluster 4	PN-Cluster 5	…	PN-Cluster N
Bus X	**S**	**S**					
Bus Y				**S**			
Bus Z							
Bus to other GwECU		**S**			**S**		
…							

	PN-Cluster 1	PN-Cluster 2	PN-Cluster 3	PN-Cluster 4	PN-Cluster 5	…	PN-Cluster N
Bus X	**S**	**S**	**S**				
Bus Y		**L**		**S**			
Bus Z			**L**				
Bus to other GwECU	**L**	**S**			**S**		
…							

Figure 5.48: Example for static and configurable PNC-to-channel-mapping

There are two different kinds of mapping table entries as shown in Figure 5.48.

- **Static PNC Mapping** (S): Entries in PNC-to-channel-mapping are defined in ARXML and stored in nonvolatile memory of the ECU.
- **Configurable PNC Mapping** (L): Entries in PNC-to-channel-mapping can be added dynamically via PNC learning mechanism but can also be reset.

Mappings can only be added to already existing PNCs. New PNCs cannot be introduced dynamically, but it shall be possible to have PNCs available even without existing mapping/actual usage. Note that even if a PNC has at least one static relation it is still learnable to additional channel relations.

“Static” entries of the PNC-to-channel-mapping (black “S” in Figure 5.48) can be defined via the `PncMapping` attributes `pncGroup`, `wakeupFrame` or `physicalChannel`.

If a PNC has no “static” mapping and therefore no entries within the PNC-to-channel-mapping, but entries shall be learnable (blue “L” in Figure 5.48), even if there is no Tx-/Rx-Relation to the Gateway running the PNC Gateway, then the PNC could be defined in System Description via the PncMapping attributes relevantForDynamicPnc-Mapping or dynamicPncMappingPduGroup.

Note: Defining such dynamic PNC-to-channel-mapping entries makes only sense if the PNC is assigned to a CommunicationConnector where dynamicPncToChannel-MappingEnabled is set to TRUE. Otherwise those dynamic mappings will not be handled by the ECU on this channel. Learning of dynamic PNC-to-channel-mapping is started by an explicit trigger. The node that starts this algorithm sends a Partial Network Learning request, which is then responded by all nodes with their current PNC Membership. PNC Gateways will forward this request to other channels. During this learning phase the PNC Gateway observes all responses and updates the PNC-to-channel-mapping accordingly, depending on which channels a PNC member was observed/received.

Note: While the PNC-to-channel-mapping could be dynamically re-configured, all routing paths (including source and target I-PDUs) within a Gateway are still configured statically. Therefore it is expected that the design of the SystemDescription ensure that all routing paths of all potentially expected PNC-to-channel-mappings are provided statically. For this dynamic PNC-to-channel-mapping some configurations restrictions and design limitations are needed to ensure proper functionality. These are provided in the following either as constraints or as text if a constraint is not feasible.

[constr_5167] pncGatewayType and ECU over the whole system

Imposition time: CP: IT_SysDesc

⌈Only one PNC Gateway ECU in the whole System shall exist that sets on all its CommunicationConnectors the pncGatewayType to active.⌋

This constraint ensures that only one PNC coordinator in the partial network exists where all its CommunicationConnectors have pncGatewayType set to active.

[constr_5168] pncGatewayType passive and connected ECUs

Imposition time: CP: IT_SysDesc

⌈For all CommunicationConnectors with pncGatewayType set to passive belonging to one PNC Gateway ECU, all connected counterpart Communication-Connectors, where pncGatewaypncGatewayTypeType is set to active shall belong to one ECU, if dynamicPncToChannelMappingEnabled is set to TRUE for at least one of the affected CommunicationConnectors.⌋

This constraint ensures that one ECU cannot be passively connected to two different coordinators. This avoids coordination problems in normal systems as there is no coordination from passive to passive. For dynamic PNC-to-channel mapping it is essential to avoid cycles in the forwarding of the learning request.

[constr_5170] nmPassiveModeEnabled and dynamicPncToChannelMapping-Enabled

Imposition time: CP: IT_SysDesc

⌈If nmPassiveModeEnabled is set to TRUE on a NmNode then dynamic-PncToChannelMappingEnabled shall be set to FALSE on the according CommunicationConnector referring to the same CommunicationController.⌋

Note: A passive NM node cannot transmit its cluster membership to the Gateway. Therefore, it cannot take part at the dynamic PNC learning and all assigned PNCs have to be statically configured, i.e. no dynamic mapping is allowed.

Additionally this constraint also leads to the fact that defining dynamic mappings for such a NmNode makes no sense as they will be ignored. The CanCommunication-Connector.pncWakeupDataMask of the Gateways have to consider learnable PNCs.

In a system with multiple (cascaded) Gateways, the RepeatMessage time has to be sufficient to fulfill its purpose and provide enough time for PNC Learning mechanism to receive at least one NM PDU of all NmNodes.

5.5.2 Com Management Mapping

The AUTOSAR BSW stack supports the configuration of the ComM module which encapsulates the control of the underlying communication services. The ComM module collects the bus communication access requests from communication requestors and coordinates the bus communication access requests. In order to utilize the communication requests from application software the Software Component Template supports the definition of PortGroups with the category MODE_MANAGEMENT. In this section it is described how PortGroups with the category MODE_MANAGEMENT are mapped to communication channels.

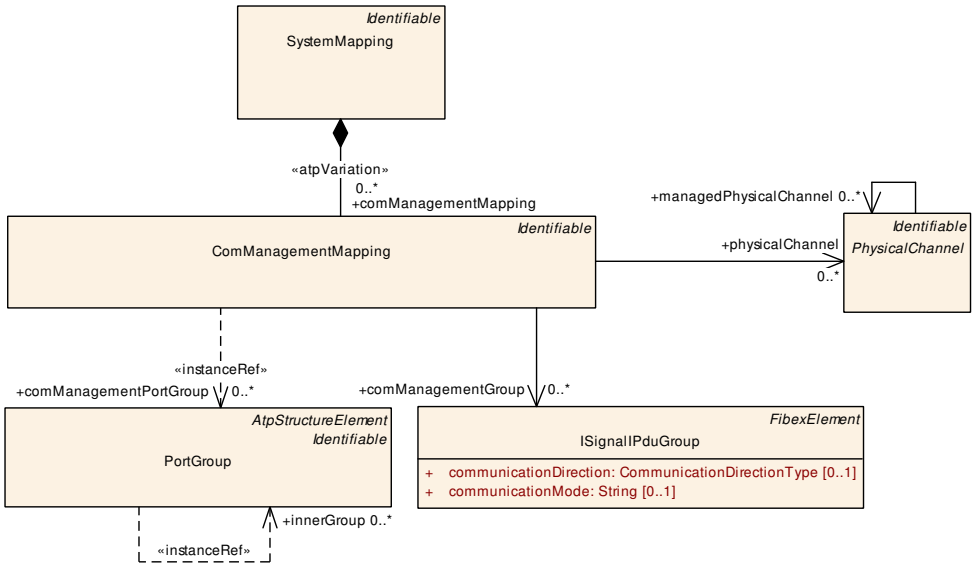

Figure 5.49: Mapping between PortGroups and communication channels

Class	**ComManagementMapping**			
Note	Describes a mapping between one or several Mode Management PortGroups and communication channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.comManagementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
com Management Group	ISignalIPduGroup	*	ref	IPduGroup participating in a Mode Management Port Group. **Tags:** atp.Splitkey=comManagementGroup
com Management PortGroup	PortGroup	*	iref	Mode Management PortGroup to be mapped onto a communication channel. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems. **Tags:** atp.Splitkey=comManagementPortGroup, com ManagementPortGroup.contextComposition, com ManagementPortGroup.contextComponent, com ManagementPortGroup.target **InstanceRef implemented by:** PortGroupInSystem InstanceRef
physical Channel	PhysicalChannel	*	ref	This reference maps the Mode Management PortGroup partial network to communication channels. **Tags:** atp.Splitkey=physicalChannel

Table 5.44: ComManagementMapping

5.6 Software Cluster Mapping

[TPS_SYST_02348] Mapping of `CpSoftwareCluster` to `EcuInstance` ⌈The assignment of a concrete `CpSoftwareCluster` to an `EcuInstance` is done by means of the meta-class `CpSoftwareClusterToEcuInstanceMapping`.⌋

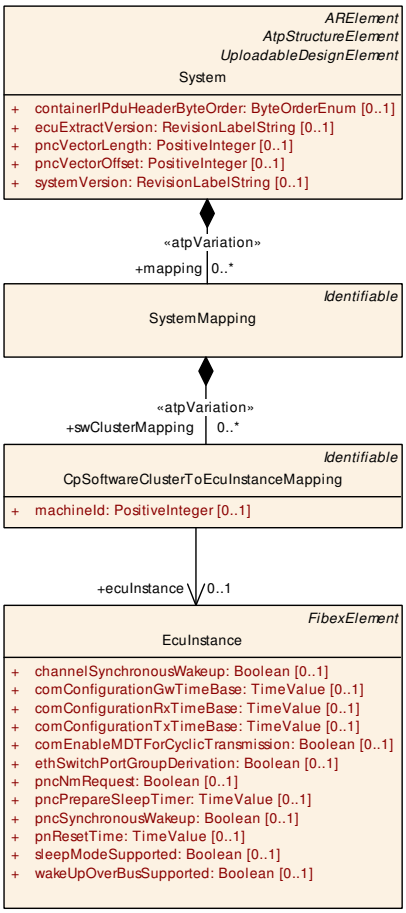

Figure 5.50: Modeling of the `CpSoftwareClusterToEcuInstanceMapping`

Class	**CpSoftwareClusterToEcuInstanceMapping**			
Note	This meta class maps a CpSoftwareCluster to a EcuInstance. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swClusterMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecuInstance	EcuInstance	0..1	ref	Reference to a specific ECU Instance description. **Tags:** atp.Splitkey=ecuInstance
machineId	PositiveInteger	0..1	attr	Unique number of the (virtual or physical) machine to which the Software Cluster is mapped.
swCluster	CpSoftwareCluster	*	ref	The mapped CP Software Cluster **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swCluster, swCluster.cpSoftwareCluster, swCluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table 5.45: CpSoftwareClusterToEcuInstanceMapping

[constr_5337] All `CpSoftwareClusterToEcuInstanceMapping`s that are referencing the same `EcuInstance` shall define the same `machineId`

Imposition time: CP: `IT_SysDesc`

⌈All `CpSoftwareClusterToEcuInstanceMapping`s that define a `machineId` and are referencing the same `EcuInstance` in the role `ecuInstance` shall have the same `CpSoftwareClusterToEcuInstanceMapping.machineId` value set.⌋

[TPS_SYST_02347] Mapping of `CpSoftwareClusterResource` to `ApplicationPartition` ⌈The assignment of a concrete `CpSoftwareClusterResource` to an `ApplicationPartition` is done by means of the meta-class `CpSoftwareClusterResourceToApplicationPartitionMapping`.⌋

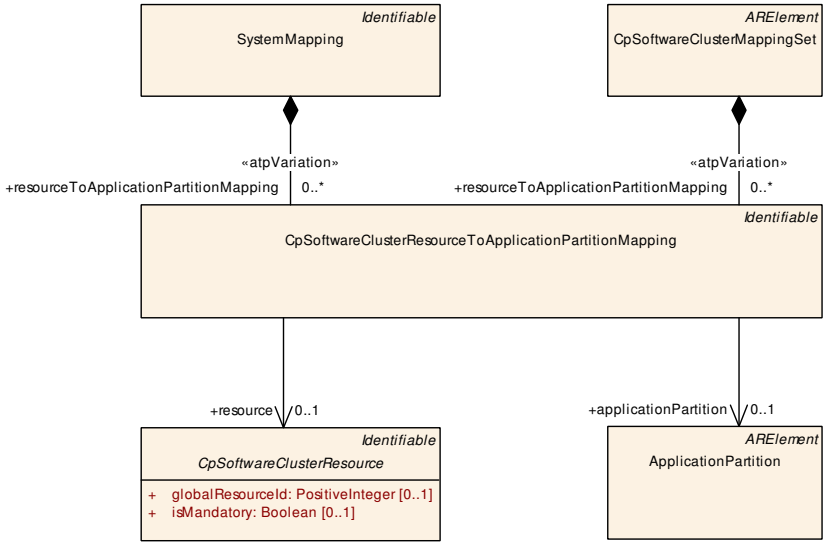

Figure 5.51: Modeling of the `CpSoftwareClusterResourceToApplicationPartitionMapping`

Class	**CpSoftwareClusterResourceToApplicationPartitionMapping**			
Note	This meta class maps a Software Cluster resource to an Application Partition to restrict the usage. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.resourceToApplicationPartitionMapping, SystemMapping.resourceToApplicationPartitionMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Partition	ApplicationPartition	0..1	ref	ApplicationPartition for which the mapping applies. **Tags:** atp.Splitkey=applicationPartition
resource	CpSoftwareCluster Resource	0..1	ref	Software Cluster Resource for which the mapping applies. **Tags:** atp.Splitkey=resource

Table 5.46: CpSoftwareClusterResourceToApplicationPartitionMapping

Class	CpSoftwareClusterMappingSet			
Note	This meta-class represents the ability to aggregate a collection of CP Software Cluster relevant mappings. This is applicable if a CP Software Cluster is described besides a concrete System, e.g. a reusable CP Software Cluster. **Tags:** atp.recommendedPackage=CpSoftwareClusterMappingSets This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
portElementTo ComResource Mapping	PortElementTo Communication ResourceMapping	*	aggr	maps a communication resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portElementToComResourceMapping, port ElementToComResourceMapping.shortName, port ElementToComResourceMapping.variationPoint.short Label vh.latestBindingTime=postBuild
resourceTo Application Partition Mapping	CpSoftwareCluster ResourceToApplication PartitionMapping	*	aggr	Maps a Software Cluster resource to an Application Partition to restrict the usage. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=resourceToApplicationPartitionMapping, resourceToApplicationPartitionMapping.shortName, resourceToApplicationPartitionMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime
softwareCluster ToApplication Partition Mapping	CpSoftwareClusterTo ApplicationPartition Mapping	0..1	aggr	Maps a Software Cluster to an Application Partition to restrict the usage. **Tags:** atp.Splitkey=softwareClusterToApplicationPartition Mapping, softwareClusterToApplicationPartition Mapping.shortName
softwareCluster ToResource Mapping	CpSoftwareClusterTo ResourceMapping	*	aggr	maps a service resource to CP Software Clusters **Stereotypes:** atpVariation **Tags:** atp.Splitkey=softwareClusterToResourceMapping, softwareClusterToResourceMapping.shortName, softwareClusterToResourceMapping.variationPoint.short Label vh.latestBindingTime=preCompileTime
swcTo Application Partition Mapping	SwcToApplication PartitionMapping	*	aggr	maps SwComponentPrototypes in a CP Software Cluster to ApplicationPartitions **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swcToApplicationPartitionMapping, swcTo ApplicationPartitionMapping.shortName, swcTo ApplicationPartitionMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 5.47: CpSoftwareClusterMappingSet

[constr_5219] `CpSoftwareCluster` shall only be mapped to one `EcuInstance`

Imposition time: CP: `IT_SwCluSysDesc`

⌈Within the context of one `CpSoftwareCluster`, for all `CpSoftware-Cluster.swComponentAssignment.swComponent` (and nested instances of `SwComponentPrototype`s) that are referenced by a `SwcToEcuMapping` in the role `component` the following condition shall be be fulfilled: all referencing `SwcToEcu-Mapping`s shall refer to the same `EcuInstance` in the role `ecuInstance` and this

EcuInstance shall also be referenced in the role ecuInstance by all CpSoftwareClusterToEcuInstanceMappings that also refer to said CpSoftwareCluster in the role swCluster.⌋

The statement made by [constr_5219] is visualized by Figure 5.52.

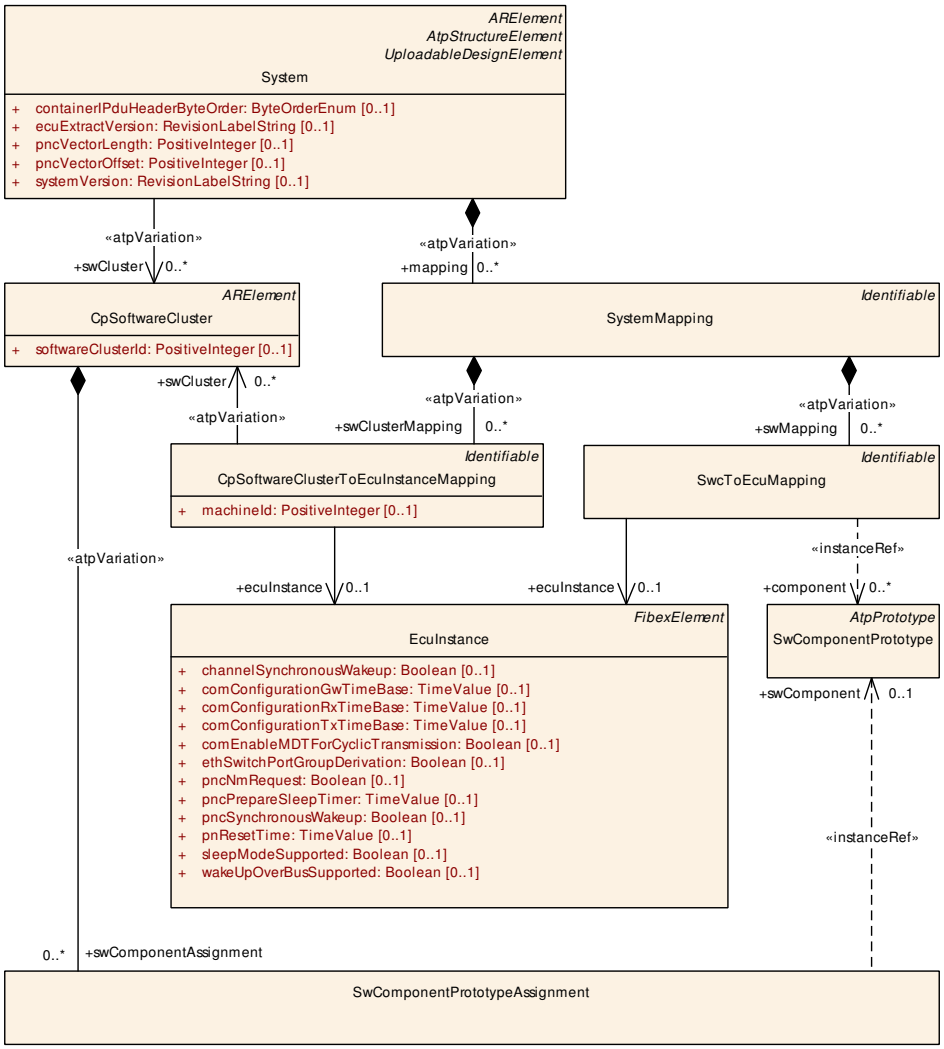

Figure 5.52: Visualization of constr_5219

[TPS_SYST_02393] Semantics of CpSoftwareClusterToApplicationPartitionMapping ⌈The CpSoftwareClusterToApplicationPartitionMapping supports the assignment of ApplicationPartitions to an applicative CpSoftwareCluster to support a top down design of which ApplicationPartitions are available for a CpSoftwareCluster.⌋

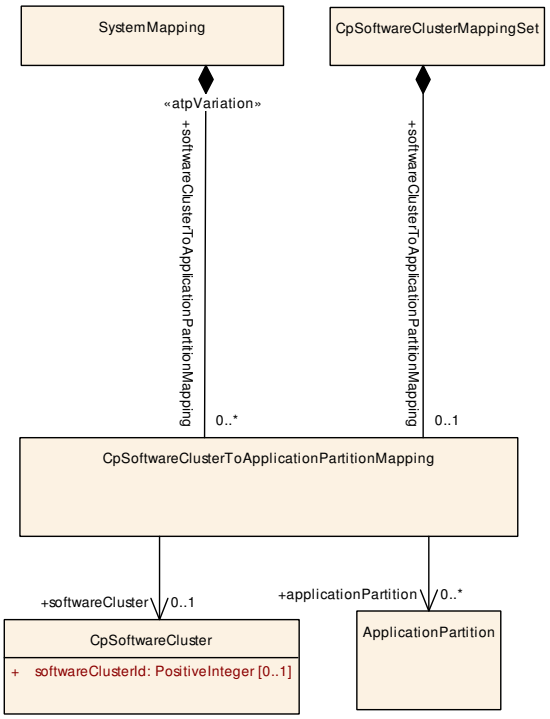

Figure 5.53: Modeling of the `CpSoftwareClusterToApplicationPartitionMapping`

Class	**CpSoftwareClusterToApplicationPartitionMapping**			
Note	This meta class defines ApplicationPartitions that are applicable for the CpSoftwareCluster. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.softwareClusterToApplicationPartitionMapping, SystemMapping.softwareClusterToApplicationPartitionMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Partition	ApplicationPartition	*	ref	Collection of ApplicationPartitions available in the Cp SoftwareCluster **Tags:** atp.Splitkey=applicationPartition
softwareCluster	CpSoftwareCluster	0..1	ref	Software Cluster Resource for which the mapping applies **Tags:** atp.Splitkey=softwareCluster

Table 5.48: CpSoftwareClusterToApplicationPartitionMapping

[TPS_SYST_02394] Aggregation possibilities of `CpSoftwareClusterToApplicationPartitionMapping` ⌈The `CpSoftwareClusterToApplicationPartitionMapping` can be aggregated by `CpSoftwareClusterMappingSet` and by `SystemMapping`.

- `CpSoftwareClusterMappingSet.softwareClusterToApplicationPartitionMapping` can be used in an early stage of the Methodology if the `System` with the `RootSwCompositionPrototype` and all included `SystemMapping`s is not available yet.

- SystemMapping.softwareClusterToApplicationPartitionMapping can be used in a later stage of Methodology if the System with the RootSwCompositionPrototype and all included SystemMappings is available.

⌋

[TPS_SYST_02395] CpSoftwareClusterToApplicationPartitionMapping aggregated by SystemMapping supersedes CpSoftwareClusterToApplicationPartitionMapping aggregated by CpSoftwareClusterMappingSet ⌈If a CpSoftwareClusterToApplicationPartitionMapping that is aggregated by the SystemMapping and a CpSoftwareClusterToApplicationPartitionMapping that is aggregated by the CpSoftwareClusterMappingSet exist at the same time and both are mapping the same ApplicationPartition to the same CpSoftwareCluster then the CpSoftwareClusterToApplicationPartitionMapping that is aggregated by the SystemMapping supersedes the CpSoftwareClusterToApplicationPartitionMapping that is aggregated by the CpSoftwareClusterMappingSet.⌋

[constr_5369] Consistency between SwcToApplicationPartitionMapping and CpSoftwareClusterToApplicationPartitionMapping

Imposition time: CP: IT_SwCluSysDesc

⌈If a CpSoftwareClusterToApplicationPartitionMapping exists between a CpSoftwareCluster and an ApplicationPartition, then all SwComponentPrototypes mapped to that ApplicationPartition by SwcToApplicationPartitionMapping shall be assigned to this CpSoftwareCluster (via CpSoftwareCluster.swComponentAssignment or CpSoftwareCluster.swComposition).⌋

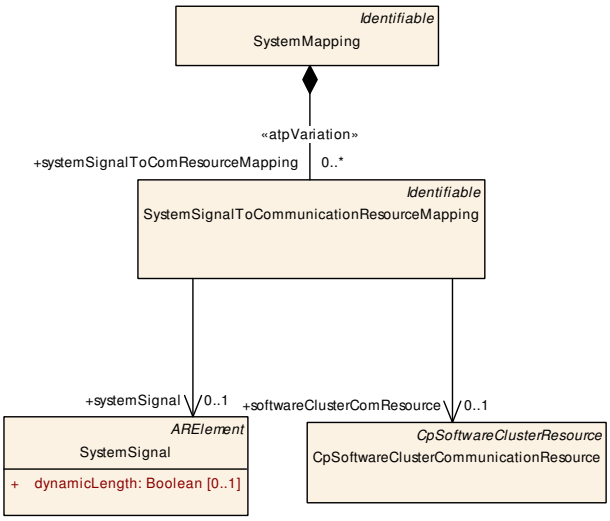

Figure 5.54: Modeling of the SystemSignalToCommunicationResourceMapping

[TPS_SYST_02397] Assignment of CpSoftwareClusterCommunicationResources to a SystemSignal ⌈In an early process stage without given DataMappings that are linking Port elements (e.g. VariableDataPrototypes, ClientServerOperations) to SystemSignals the SystemSignalToCommunicationResourceMapping can be used to map a CpSoftwareClusterCommunicationResource to a SystemSignal.⌋

Class	**SystemSignalToCommunicationResourceMapping**			
Note	This meta class maps a communication resource to a SystemSignal. This mapping can be used in an early process stage in which the DataMapping linking the Ports and mapped CpSoftwareCluster CommunicationResource(s) to the SystemSignal is not yet available. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.systemSignalToComResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
softwareCluster ComResource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=softwareClusterComResource
systemSignal	SystemSignal	0..1	ref	SystemSignal to which the communication resource is assigned **Tags:** atp.Splitkey=systemSignal

Table 5.49: SystemSignalToCommunicationResourceMapping

[constr_5370] Restriction for SystemSignalToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignal

Imposition time: CP: IT_SwCluSysDesc

⌈If a DataMapping to a SystemSignal exists for the port element (e.g. VariableDataPrototype, ClientServerOperation) that is mapped by the PortElementToCommunicationResourceMapping to a CpSoftwareClusterCommunicationResource and a SystemSignalToCommunicationResourceMapping exists for the same SystemSignal then the SystemSignalToCommunicationResourceMapping shall map this SystemSignal to the same CpSoftwareClusterCommunicationResource.⌋

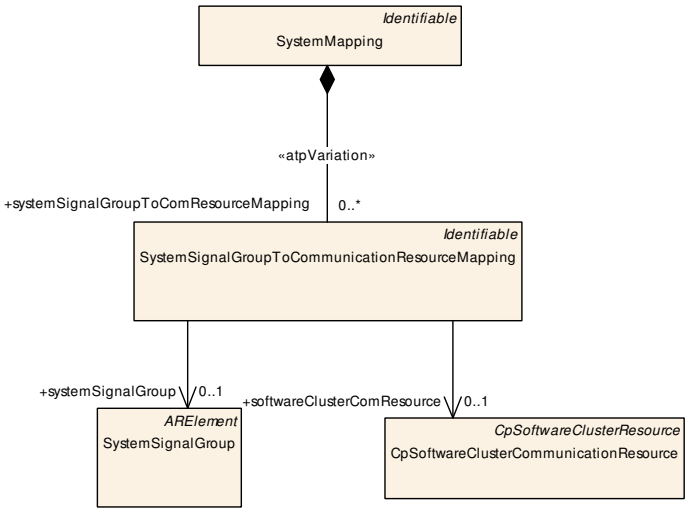

Figure 5.55: Modeling of the SystemSignalGroupToCommunicationResourceMapping

[TPS_SYST_02398] Assignment of CpSoftwareClusterCommunication-Resources to a SystemSignalGroup ⌈In an early process stage without given DataMappings that are linking a VariableDataPrototype to a SystemSignal-Group the SystemSignalToCommunicationResourceMapping can be used to map a CpSoftwareClusterCommunicationResource to a SystemSignal-Group.⌋

Class	**SystemSignalGroupToCommunicationResourceMapping**			
Note	This meta class maps a communication resource to a SystemSignalGroup. This mapping can be used in an early process stage in which the DataMapping linking the Ports and mapped CpSoftwareCluster CommunicationResource(s) to SystemSignals of a SystemSignalGroup is not yet available. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.systemSignalGroupToComResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
softwareCluster ComResource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=softwareClusterComResource
systemSignal Group	SystemSignalGroup	0..1	ref	SystemSignalGroup to which the communication resource is assigned **Tags:** atp.Splitkey=systemSignalGroup

Table 5.50: SystemSignalGroupToCommunicationResourceMapping

[constr_5371] Restriction for SystemSignalGroupToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignalGroup

Imposition time: CP: IT_SwCluSysDesc

⌈If a SenderReceiverToSignalGroupMapping to a SystemSignalGroup exists for the VariableDataPrototype that is mapped by the PortElementToCommunicationResourceMapping to a CpSoftwareClusterCommunicationResource and a SystemSignalGroupToCommunicationResourceMapping exists for the same SystemSignalGroup then the SystemSignalGroupToCommunicationResourceMapping shall map this SystemSignalGroup to the same CpSoftwareClusterCommunicationResource.⌋

5.7 Dds Mapping

In AUTOSAR CP, a DDS Topic can ba associated to an ISignal. To map such an association, the DdsCpISignalToDdsTopicMapping is used, that maps an ISignal to one DdsCpTopic.

It is possible to have multiple scenarios that combine one or more DdsCpTopics mapped to one or more ISignals. In the following paragraphs, an explanation of each scenario is given:

- **(1:1): One ISignal mapped to One DdsCpTopic:**

  In this scenario, a single DdsCpISignalToDdsTopicMapping represents a unique linkage between an AUTOSAR ISignal and a DdsCpTopic in the context of a DdsCpDomain and one or more DdsCpPartitions within it. S/R ports connected to such ISignal will have their I/O operations connected to a DDS DataWriter or DataReader entity suitable for accessing the DDS Topic Type, name, Domain, Partition and QoS policies specified directly or indirectly by the referenced DdsCpTopic.

- **(1:N): One ISignal to Multiple DdsCpTopics (DDS FAN-OUT):**

  In this scenario, multiple DdsCpISignalToDdsTopicMapping elements represent a linkage between one AUTOSAR ISignal and multiple DdsCpTopics within one or more DdsCpDomains and one or more DdsCpPartitions within them. S/R Ports connected to such ISignal will have their I/O operations connected to multiple DDS DataWriter or DataReader entities suitable for accessing the DDS Topic Types, names, Domains, Partitions and QoS policies specified directly or indirectly by the referenced DdsCpTopics. Example use cases might be:

  - Propagating a signal in multiple DDS Domains and/or DDS Partitions
  - Propagating a signal under multiple QoS profiles
  - Propagating a single under multiple DDS Topic names

In the picture below an example of 1:N mapping is shown:

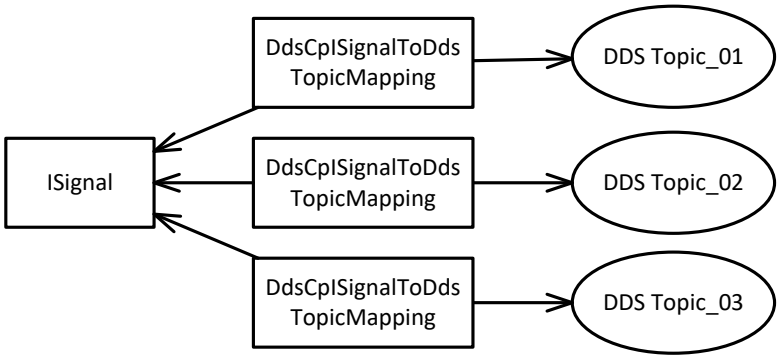

Figure 5.56: One ISignal mapped to more than one DDS Topic

- **(N:1): Multiple `ISignal`s to One `DdsCpTopic`:**

  In this scenario, multiple `DdsCpISignalToDdsTopicMapping` elements represent a linkage between several AUTOSAR `ISignal`s and a single `DdsCpTopic` in the context of a `DdsCpDomain` and one or more `DdsCpPartition`s within it. S/R Ports connected to such `ISignal`s will have their I/O operations connected to a DDS DataWriter or DataReader entity suitable for accessing the DDS Topic Type, name, Domain, Partition and QoS policies specified directly or indirectly by the referenced `DdsCpTopic`. Example use cases might be:

  – Consolidating multiple Elements from different PortInterfaces, realised in multiple PortPrototypes, into a single DDS Topic

  In the picture below an example of N:1 mapping is shown:

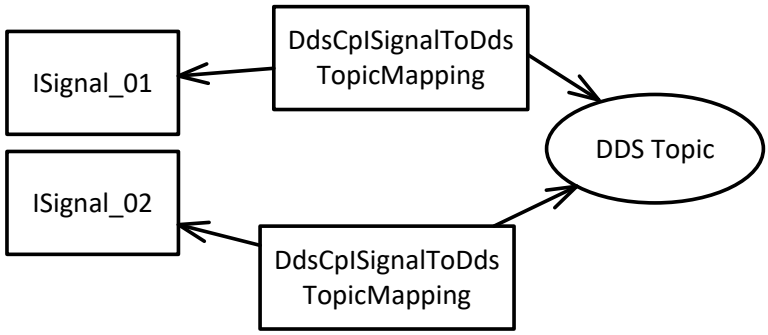

Figure 5.57: One DDS Topic mapped by more than one ISignal

- **(N:M) Multiple `ISignal`s to Multiple `DdsCpTopic`s:**

  In this scenario, multiple `DdsCpISignalToDdsTopicMapping` elements represent a linkage between several AUTOSAR `ISignal`s and several `DdsCpTopic`s within one or more `DdsCpDomain`s and one or more `DdsCpPartition`s within

each of them. S/R Ports connected to such ISignals will have their I/O operations connected to multiple DDS DataWriter or DataReader entities suitable for accessing the DDS Topic Types, names, Domains, Partitions and QoS policies specified directly or indirectly by the referenced DdsCpTopics. Example use cases might be:

- Propagating a signal in multiple DDS Domains and/or DDS Partitions
- Propagating a signal under multiple QoS profiles
- Propagating a single under multiple DDS Topic names
- Consolidating multiple Elements from different PortInterfaces, realised in multiple PortPrototypes into a single DDS Topic

To model each of such scenario, the generic multiplicity N:M is used. It is possible to configure the actual multiplicity according the desired use-case.

Note: Since a Dds Topic is strictly related to the data type it deals with, while configuring the user shall allow different ISignals sharing the same DDS Topic only if the ISignals are related to the same data type.

The picture below shows the architecture of DdsCpISignalToDdsTopicMapping.

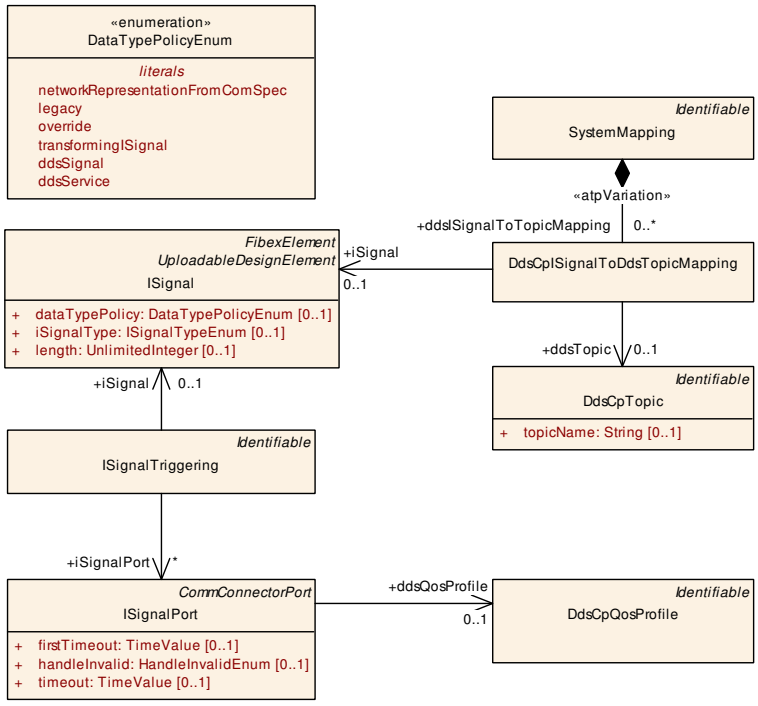

Figure 5.58: ISignal to Dds Topic mapping

Class	**DdsCpISignalToDdsTopicMapping**			
Note	Mapping of an ISignal to a DdsTopic. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	SystemMapping.ddsISignalToTopicMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic. **Tags:** atp.Splitkey=ddsTopic atp.Status=candidate
iSignal	ISignal	0..1	ref	Reference to the ISignal. **Tags:** atp.Splitkey=iSignal atp.Status=candidate

Table 5.51: DdsCpISignalToDdsTopicMapping

5.8 EcuPartition to Core Mapping

[TPS_SYST_02420] Semantic of `EcuPartitionToCoreMapping` ⌈The `EcuPartitionToCoreMapping` maps an `EcuPartition` to a *Core*, which is represented by a `HwElement` with `category` "'ProcessingUnit"'.⌋

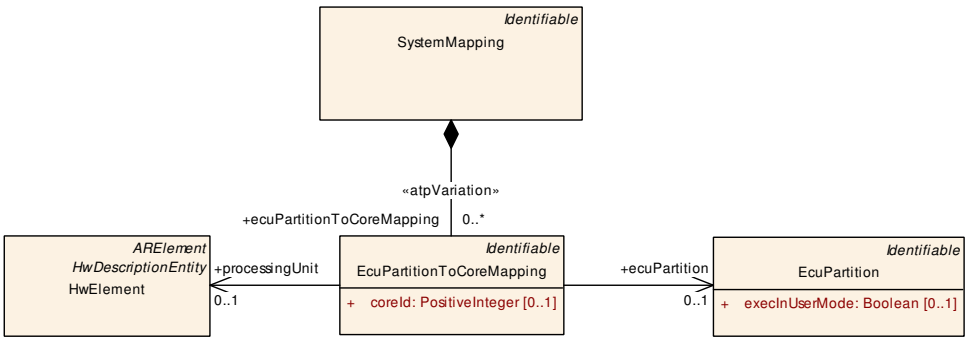

Figure 5.59: EcuPartition to Core Mapping

Class	**EcuPartitionToCoreMapping**			
Note	This element maps an EcuPartion to a Core that is represented by a HwElement with category ProcessingUnit. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.ecuPartitionToCoreMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
coreId	PositiveInteger	0..1	attr	Optional definition of Core Id.
ecuPartition	EcuPartition	0..1	ref	EcuPartition that is mapped to a ProcessingUnit **Tags:** atp.Splitkey=ecuPartition

▽

△

Class	**EcuPartitionToCoreMapping**			
processingUnit	HwElement	0..1	ref	HwElement with category ProcessingUnit to which the EcuPartion is mapped. **Tags:** atp.Splitkey=processingUnit

Table 5.52: EcuPartitionToCoreMapping

[constr_9355] Category of `HwElement` for `EcuPartitionToCoreMapping`

Imposition time: CP: `IT_EcuExt`

⌈The `HwElement`, which is referenced by `EcuPartitionToCoreMapping` in the role `processingUnit`, shall be of `category` "'ProcessingUnit"'.⌋

[constr_9359] Different `EcuPartitionToCoreMapping.coreId` attribute values shall not be assigned to the same `processingUnit`

Imposition time: CP: `IT_EcuExt`

⌈`EcuPartitionToCoreMapping`s that reference the same `HwElement` in the role `processingUnit` shall not define different values for the `EcuPartitionToCoreMapping.coreId` attribute.⌋

If several `EcuPartitionToCoreMapping`s reference the same `HwElement` in the role `processingUnit` it is allowed that only a subset of those `EcuPartitionToCoreMapping`s defines the `coreId` value as long as [constr_9359] is fulfilled and different `coreId` values are not assigned to the same `processingUnit`.

5.9 Cyclic Handling of Communication Data in the RTE Mapping

This chapter defines the possibility to describe a mapping of `VariableDataPrototype`s to `OsTask`s in the context of a `System`. This mapping is used as input to the RTE generator to generate the Cyclic Handling of Communication Data in the RTE behavior (see [SWS_Rte_04654] in SWS RTE [23] for more details).

The meta-class `OsTaskProxy` is used in the System Description to represent an `OsTask` that is defined in the Ecu Configuration of the OS. Please note that if defined the `ReceptionComSpecProps.dataUpdatePeriod` needs to be respected by the `CyclicHandlingComDataToOsTaskProxyMapping`.

[TPS_SYST_02440] Semantic of `CyclicHandlingComDataToOsTaskProxyMapping` ⌈The `CyclicHandlingComDataToOsTaskProxyMapping` element is used to map `VariableDataPrototype`s to an `OsTaskProxy` as input for the generation of cyclic functions in the RTE behavior according to [SWS_Rte_04654] defined in the SWS RTE [23].⌋

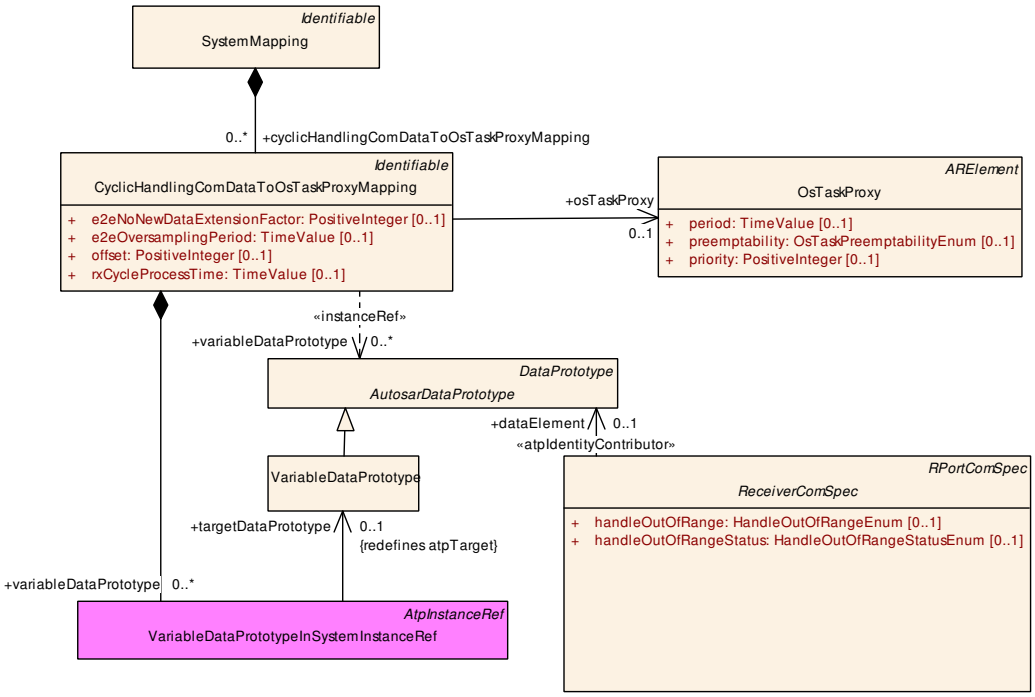

Figure 5.60: `VariableDataPrototype` to `OsTaskProxy` mapping in the context of Cyclic Handling of Communication Data in the RTE

[TPS_SYST_02441] Mapping of `VariableDataPrototype`s to the same `OsTaskProxy` ⌈`VariableDataPrototype`s that are mapped by `CyclicHandlingComDataToOsTaskProxyMapping`s to the same `OsTaskProxy` in the context of the same `System` shall be mapped to the same `OsTask` in the Ecu Configuration.⌋

Class	**CyclicHandlingComDataToOsTaskProxyMapping**			
Note	This meta-class is used to map VariableDataPrototypes to an OsTaskProxy for the Cyclic Handling of Communication Data in the RTE. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cyclicHandlingComDataToOsTaskProxyMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
e2eNoNewData ExtensionFactor	PositiveInteger	0..1	attr	This attribute defines the jitter factor of E2E no new data reception cycle time.
e2e Oversampling Period	TimeValue	0..1	attr	This attribute defines an optional E2E oversampling cycle time for the cyclic processing of the mapped VariableData Prototypes.
offset	PositiveInteger	0..1	attr	This attribute is used to describe the position in the Os Task as a relative value.
osTaskProxy	OsTaskProxy	0..1	ref	Reference to the OsTaskProxy. **Tags:** atp.Splitkey=osTaskProxy
rxCycleProcess Time	TimeValue	0..1	attr	This attribute defines the cyclic processing time within the OsTask.

△

Class	CyclicHandlingComDataToOsTaskProxyMapping			
variableData Prototype	VariableDataPrototype	*	iref	Reference to the VariableDataPrototypes for which the "Cyclic Handling of Communication Data in the RTE" will be implemented by the RTE. **Tags:** atp.Splitkey=variableDataPrototype, variableData Prototype.contextComponent, variableData Prototype.contextComposition, variableData Prototype.contextPort, variableDataPrototype.targetData Prototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef

Table 5.53: CyclicHandlingComDataToOsTaskProxyMapping

[constr_9384] Relation between both `CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime` and `CyclicHandlingComDataToOsTaskProxyMapping.offset` and `OsTaskProxy.period`

Imposition time: CP: `IT_SysDesc`

⌈The `CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime` and `CyclicHandlingComDataToOsTaskProxyMapping.offset` (if defined) shall be a positive integer multitude of `period` defined in the `OsTaskProxy` that is referenced by the `CyclicHandlingComDataToOsTaskProxyMapping`.⌋

[constr_9385] Existence of `CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy`

Imposition time: CP: `IT_SysDesc`

⌈For each `CyclicHandlingComDataToOsTaskProxyMapping` the reference `CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy` shall exist.⌋

[constr_9386] Existence of `CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype`

Imposition time: CP: `IT_SysDesc`

⌈For each `CyclicHandlingComDataToOsTaskProxyMapping` the reference `CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype` shall exist at least once.⌋

[constr_9387] One `CyclicHandlingComDataToOsTaskProxyMapping` for every `VariableDataPrototype` with `ReceptionComSpecProps.comHandlerTaskMappingEnabled` = TRUE

Imposition time: CP: `IT_SysDesc`

⌈Every `VariableDataPrototype` that

- is referenced by a `ReceiverComSpec` with `ReceptionComSpecProps.comHandlerTaskMappingEnabled` = TRUE and
- is mapped to a `SystemSignal` or `SystemSignalGroup` and

- that SystemSignal or SystemSignalGroup has an ISignal or ISignal-Group that is received in the context of the System,

, then there shall be exactly one CyclicHandlingComDataToOsTaskProxy-Mapping in the context of that System referencing that VariableDataPrototype.⌋

6 Communication

This chapter describes all topics that deal with constraints or configurations that describe the information exchange between the ECUs. The description of communication matrices in the System Template is based on the description in ASAM FIBEX [8]. Because of the requirements of AUTOSAR some extensions were made to the original FIBEX model.

The main elements to describe communication in the System Template are SystemSignals, ISignals, Pdus and Frames, as it can be seen on Figure 6.1.

Frames can be defined independently of communication clusters. On the communication channel the Frame is represented by the referencing FrameTriggering.

A Frame has a payload section of a certain length in bytes, which contains an arbitrary number of non-overlapping Pdus. In AUTOSAR only FlexRay supports the packing and unpacking of multiple Pdus into/out of one FlexRay Frame. The AUTOSAR CanIf and LinIf are not capable of packing multiple Pdus into one Frame.

[constr_3036] Pdus in CAN and LIN Frames

Imposition time: CP: IT_SysDesc

⌈CAN Frames and LIN Frames shall only contain one Pdu.⌋

Note that via the ContainerIPdu it is possible to transport several IPdus in one ContainerIPdu in order to support CAN FD.

A Pdu (Protocol Data Unit) is the information delivered through a network layer. For the network to understand which layer is being discussed, a single-letter prefix is added to the PDU.

- IPdu - Interaction Layer Protocol Data Unit (assembled and disassembled in Com). In the case of external communication the Interaction Layer packs one or more signals into assigned IPdus and passes them to the underlying layer for transfer between nodes in a network.
- NPdu - Network Layer Protocol Data Unit (assembled and disassembled in a Transport Protocol module). The TP module's main purpose is the segmentation and reassembly of IPdus that do not fit in one of the assigned NPdus.
- LPdu - Data Link Layer Protocol Data Unit (assembled and disassembled in AUTOSAR Hardware Abstraction layer). The element Frame in the System Template represents the AUTOSAR Layered Architectures LSdu. Sdu is the abbreviation of "Service Data Unit". The Data Link Layers LPdu contains the LSdu and PCI (Protocol Control Information). The LPdu is not described in the System Template.

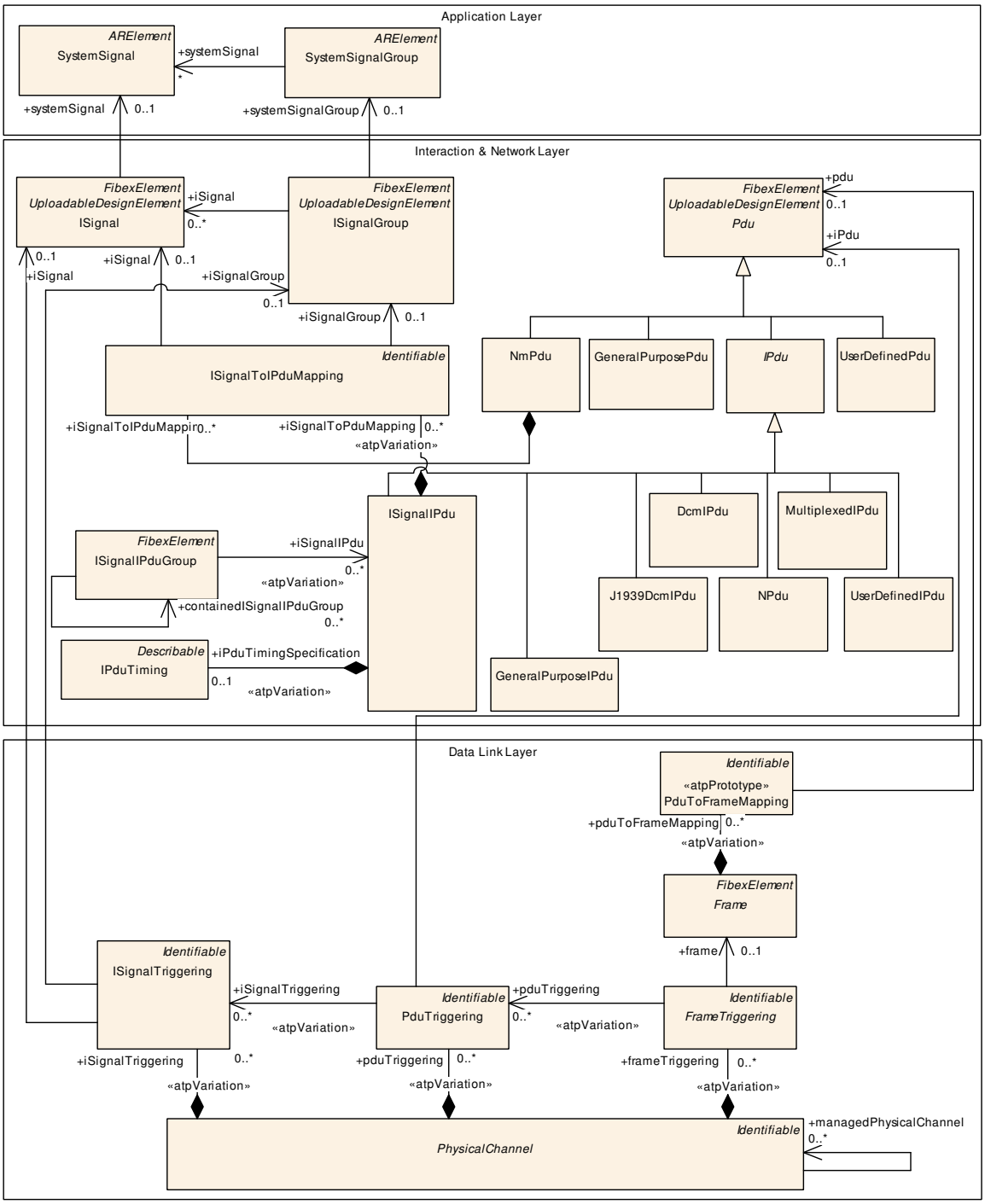

Figure 6.1: Communication Overview (FibexCore: Communication)

In case no multiplexing is performed the IPdus of Com that fit into one frame can be passed directly via the PDU Router to the communication interfaces.

[constr_3037] maximum Frame frameLength for CAN and LIN

Imposition time: CP: IT_SysDesc

⌈For CAN and LIN the maximum frameLength is 8 bytes and 64 bytes in case of CAN FD.⌋

[constr_3038] maximum `Frame frameLength` for FlexRay

Imposition time: CP: `IT_SysDesc`

⌈For FlexRay the maximum `frameLength` is 254 bytes.⌋

[TPS_SYST_01048] Handling of large `IPdu`s ⌈Large `IPdu`s that are too long to fit into one `Frame` of the respective subclass of `CommunicationCluster` shall be routed via a Transport Protocol to the communication interfaces.⌋

For example an `IPdu` with the length of 10 bytes needs to be routed via a Transport Protocol on CAN but on FlexRay this is not required.

The Transport Protocols are described in more detail in chapter 6.8.

If multiplexing is performed an `IPdu` is routed between the IPdu Multiplexer and the Interface Layer or Transport Layer. To distinguish these two different cases two specializations `ISignalIPdu` and `MultiplexedIPdu` are introduced. A `ISignalIPdu` represents an `IPdu` handled by AUTOSAR Com. The AUTOSAR IPduM is responsible to combine Com `ISignalIPdu`s to `MultiplexedIPdu`s. On receiver-side the IPduM is responsible to interpret the content of `MultiplexedIPdu`s and provide Com separated `ISignalIPdu`s by taking into account the value of the selector field. The IPdu Multiplexer is described in more detail in chapter 6.5.

AUTOSAR Com provides the possibility to define Transmission Modes for each Com `ISignalIPdu`. For this reason the `ISignalIPdu` aggregates the `IPduTiming`. The Transmission Modes are described in more detail in chapter 6.4.

6.1 Triggerings and Ports

The elements `FrameTriggering`, `PduTriggering` and `ISignalTriggering` are describing the usage of `Frame`s, `IPdu`s and `ISignal`s on a `PhysicalChannel`.

A `FrameTriggering` need to fulfill requirements for contained `Pdu`s that are defined by the corresponding `PduTriggering`s. And the `PduTriggering` need to fulfill requirements for contained `ISignal`s that are defined by the corresponding `ISignalTriggering`s. The references between the Triggering elements can be used to describe these relationships. More details can be found in class tables of `FrameTriggering`, `PduTriggering` and `ISignalTriggering`.

In AUTOSAR the timing of bus messages can be controlled by send requests of the Application layer in combination with the Com Transmission Modes and Transfer Properties (esp. CAN). On the other hand it can be controlled by the FlexRay or LIN Interface. In this case the Bus Interface only requests `IPdu`s that have to be provided by Com.

In the System Template the Com controlled timing is described with the aggregation between the `ISignalIPdu` and the `IPduTiming`. The LIN and FlexRay Scheduling Tables are described in the `FrameTriggering`.

Timing requirements for FlexRay and LIN Pdus can be specified with the Timing Extension model. More details are described in chapter 1.4.3.

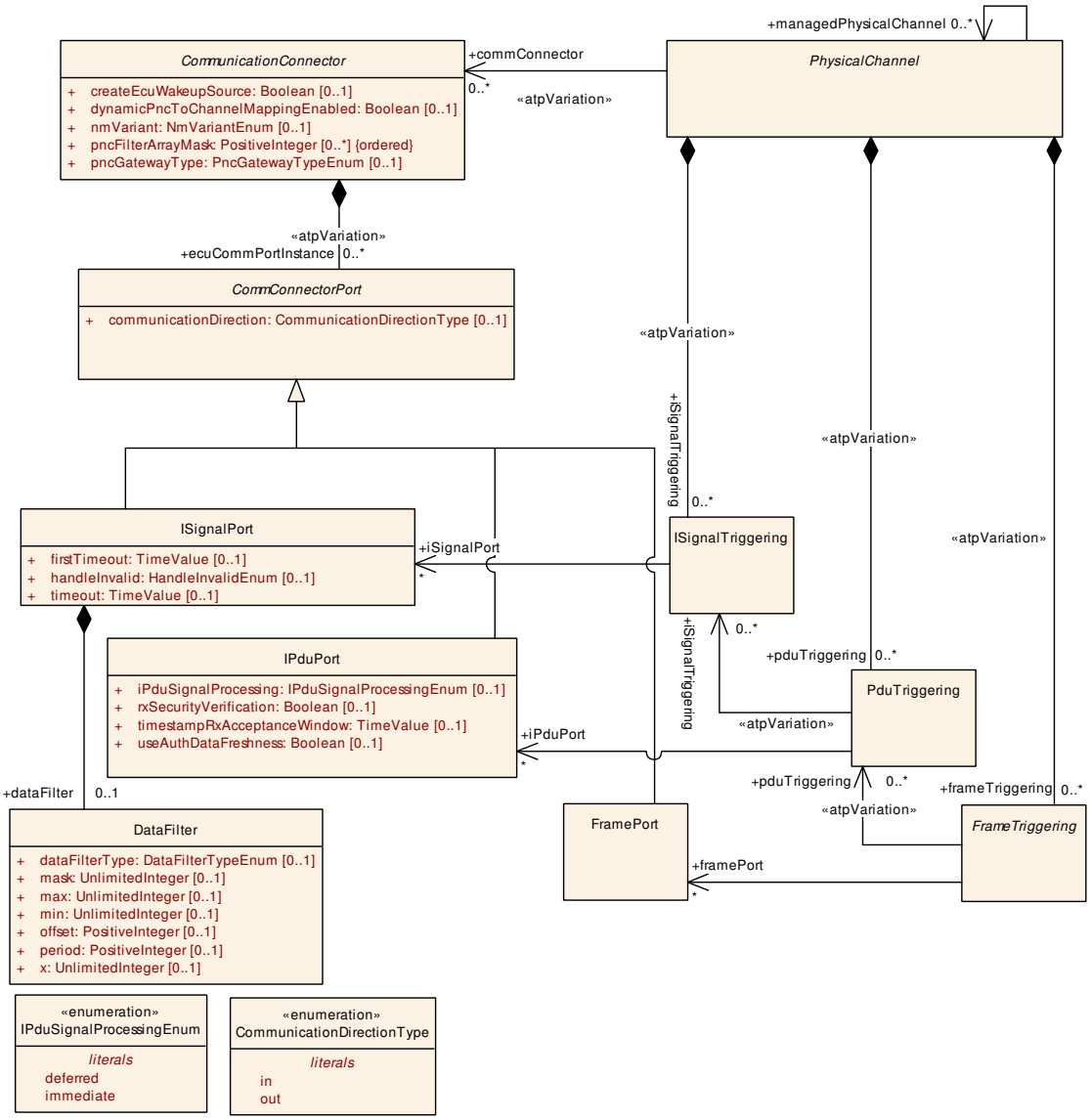

Figure 6.2: Communication Matrix (FibexCore: CommunicationMatrix)

Figure 6.2 shows the relationship between the `CommConnectorPort` and the `FrameTriggering`, `PduTriggering` and `ISignalTriggering`. This relationship allows to specify explicitly which `Frame`s, `Pdu`s, `ISignal`s are received/sent by the connected ECU on the connected channel.

[constr_3243] `FrameTriggering.pduTriggering` condition

Imposition time: CP: `IT_SysDesc`

⌈A `FrameTriggering` shall reference a `PduTriggering` if the `PduTriggering` references a `Pdu` that is referenced by a `PduToFrameMapping` which in turn is aggregated by the `Frame` that is referenced by that `FrameTriggering`.⌋

[constr_3250] `PduTriggering.iSignalTriggering` condition

Imposition time: CP: `IT_SysDesc`

⌈A `PduTriggering` shall reference an `ISignalTriggering` if the `ISignal-Triggering` references an `ISignal` or an `ISignalGroup` that is referenced by an `ISignalToIPduMapping` which in turn is aggregated by the `Pdu` that is referenced by that `PduTriggering`.⌋

[TPS_SYST_02102] `FrameTriggering.pduTriggering` references that shall be ignored ⌈References from `FrameTriggering` to `PduTriggering` which are not covered by [constr_3243] shall be ignored.⌋

As a consequence of [constr_3243] the following implications can be derived:

- The `PduTriggering` of the `ContainerIPdu` is referenced from the `Frame-Triggering` but the `PduTriggering`s of the contained `IPdu`s are not referenced from the `FrameTriggering`.
- The `PduTriggering` of the `MultiplexedIPdu` is referenced from the `Frame-Triggering` but the `PduTriggering`s of the multiplexed Part `Pdu`s are not referenced from the `FrameTriggering`.

[TPS_SYST_02104] Triggerings on `PhysicalChannel` ⌈The following modeling creates a "membership" of `ISignal`s, `ISignalGroup`s, `Pdu`s, and `Frame`s in a given `PhysicalChannel`:

- `PhysicalChannel` aggregates
  - `ISignalTriggering` that in turn references `ISignal` in the role `iSignal`
  - `ISignalTriggering` that in turn references an `ISignalGroup` in the role `iSignalGroup` ([constr_5106] applies).
- `PhysicalChannel` aggregates `PduTriggering` that in turn references a `Pdu` in the role `iPdu`.
- `PhysicalChannel` aggregates `FrameTriggering` that in turn references a `Frame` in the role `frame`.

⌋

[constr_5106] `ISignalGroup` and `ISignal` referenced from `ISignal-Triggering`

Imposition time: CP: `IT_SysDesc`

⌈Either an `ISignalGroup` and all `ISignal`s referenced from the `ISignalGroup` are also referenced from `ISignalTriggering`s aggregated at the same `Physical-Channel` or neither the `ISignalGroup` nor any of the `ISignal`s referenced by the `ISignalGroup` shall be referenced from `ISignalTriggering`s.⌋

[TPS_SYST_01142] Rules for the creation of references to Ports (`ecuCommPort-Instance`) with `communicationDirection out` on sending Ecu ⌈

- Application sends ISignal or ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup members that are sent by the Application.
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- COM Signal Gateway
  - Reference from ISignalTriggering to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup
  - Reference from ISignalTriggering to ISignalPort for a subset of ISignals inside the ISignalGroup shall be created (in case not all members of the ISignalGroup participate in the target Signal Gateway relation).
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- ISignal or ISignalGroup is mapped to ISignalIPdu but NOT sent by Application or Signal Gateway
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for any ISignal that is part of an ISignalGroup
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- Neither ISignal, ISignalGroup, Pdu, nor Frame sent by the ECU
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignalGroup

  - No ISignalTriggering reference to ISignalPort shall be created for any ISignal that is part of an ISignalGroup
  - No PduTriggering reference to IPduPort shall be created
  - No FrameTriggering reference to FramePort shall be created

⌋

Please note that it is possible to configure a signal that is transmitted by an application and also routed by a SignalGateway. At runtime it has to be ensured that only one path is active at a particular point in time (to avoid race conditions in COM Stack).

[TPS_SYST_02106] Rules for the creation of references to Ports (ecuCommPortInstance) with communicationDirection in on receiving Ecu ⌈

- Application receives ISignal or ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup members that are received by the Application.
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- COM Signal Gateway
  - Reference from ISignalTriggering to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - Reference from ISignalTriggering to ISignalPort shall be created for the ISignalGroup
  - Reference from ISignalTriggering to ISignalPort for a subset of ISignals inside the ISignalGroup shall be created (in case not all members of the ISignalGroup participate in the source Signal Gateway relation).
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- ISignal or ISignalGroup is mapped to ISignalIPdu but NOT received by Application or Signal Gateway
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup

  - No ISignalTriggering reference to ISignalPort shall be created for an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for any ISignal that is part of an ISignalGroup
  - PduTriggering reference to IPduPort shall be created
  - FrameTriggering reference to FramePort shall be created
- Neither ISignal, ISignalGroup, Pdu, nor Frame received by the ECU
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignal that is not part of an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for an ISignalGroup
  - No ISignalTriggering reference to ISignalPort shall be created for any ISignal that is part of an ISignalGroup
  - No PduTriggering reference to IPduPort shall be created
  - No FrameTriggering reference to FramePort shall be created

⌋

[constr_3252] ISignalTriggering.iSignalPort reference condition

Imposition time: CP: IT_SysDesc

⌈An ISignalTriggering shall only reference an ISignalPort if the CommunicationConnector aggregating that ISignalPort is referenced by the PhysicalChannel which in turn aggregates that ISignalTriggering.⌋

[constr_3253] PduTriggering.iPduPort reference condition

Imposition time: CP: IT_SysDesc

⌈A PduTriggering shall only reference an IPduPort if the CommunicationConnector aggregating that IPduPort is referenced by the PhysicalChannel which in turn aggregates that PduTriggering.⌋

[constr_3254] FrameTriggering.framePort reference condition

Imposition time: CP: IT_SysDesc

⌈A FrameTriggering shall only reference a FramePort if the CommunicationConnector aggregating that FramePort is referenced by the PhysicalChannel which in turn aggregates that FrameTriggering.⌋

[constr_3255] `FrameTriggering.pduTriggering` reference condition with regard to the `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈A `FrameTriggering` shall only reference a `PduTriggering` in the role `pduTriggering` if both the `FrameTriggering` and `PduTriggering` are aggregated by the same `PhysicalChannel`.⌋

[constr_3256] `PduTriggering.iSignalTriggering` reference condition with regard to the `PhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈A `PduTriggering` shall only reference an `ISignalTriggering` in the role `iSignalTriggering` if both the `PduTriggering` and `ISignalTriggering` are aggregated by the same `PhysicalChannel`.⌋

The following rules apply for the creation of `PduTriggering`s and `IPduPort`s:

- **[TPS_SYST_01052] Routing of `UserDefinedPdu`s, `NmPdu`s, `NPdu`s** ⌈`UserDefinedPdu`s, `NmPdu`s, `NPdu`s which are not going through the Pdu Router get their triggering information via the containing `FrameTriggering` and `FramePort` (no `PduTriggering` is defined for these `Pdu`s).⌋

- **[TPS_SYST_03021] Routing of `GeneralPurposePdu`s with category GLOBAL_TIME** ⌈`GeneralPurposePdu`s with category GLOBAL_TIME shall have `PduTriggering` and `IPduPort`s defined.⌋

- **[TPS_SYST_02091] Routing of `GeneralPurposePdu`s with category SD and `GeneralPurposePdu`s with category DoIP** ⌈`GeneralPurposePdu`s with category SD and `GeneralPurposePdu`s with category DoIP shall have `PduTriggering` and `IPduPort`s defined since no `Frame`s and `FrameTriggering`s are defined for `Pdu`s that are handled by the SoAd.⌋

- **[TPS_SYST_01053] Low-level routing of `NPdu`s** ⌈In case of a low-level routing of `NPdu`s the `Pdu`s are handled like `IPdu`s and the `PduTriggering` and `IPduPort` shall be defined.⌋

- **[TPS_SYST_01138] Low-level routing of XcpPdus** ⌈Low-level routing of `GeneralPurposeIPdu`s with `category` XCP: In case of a low-level routing of `GeneralPurposeIPdu`s with `category` XCP the `Pdu`s are handled like `IPdu`s and the `PduTriggering` and `IPduPort` shall be defined.⌋

- **[TPS_SYST_01054] Routing of `DcmIPdu`s** ⌈`DcmIPdu`s shall have `PduTriggering` and `IPduPort`s since they are handled by the PduR (connection to the Dcm and/or DcmIPdu-routing).⌋

- **[TPS_SYST_01055] Routing of ISignalIPdus that are part of a MultiplexedIPdu** ⌈ISignalIPdus that are part of a MultiplexedIPdu (static or dynamic) and are also handled by the Com module shall have a PduTriggering and IPduPorts since they are handled by the PduR (and Com). Especially it is allowed to ignore certain received parts of a MultiplexedIPdu in a specific ECU.⌋

- **[TPS_SYST_01056] Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus, ContainerIPdus** ⌈ISignalIPdus (not part of MultiplexedIPdus), UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus and ContainerIPdus shall have a PduTriggering and IPduPort if they are handled by the PduR. Especially it is allowed to ignore a certain IPdu out of a Flexray frame if it is not considered in a specific ECU.⌋

- **[TPS_SYST_01057] Routing of NmPdus** ⌈If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com the NmPdu shall also be referenced by a corresponding PduTriggering where attribute iPduPort exists accordingly.⌋

- **[TPS_SYST_02059] Routing of SecuredIPdus** ⌈SecuredIPdus shall have a PduTriggering and IPduPort defined since they are handled by the PduR. Pdus that are part of a SecuredIPdu and are also handled by the Com module shall have a PduTriggering and IPduPorts since they are handled by the PduR (and Com).⌋

- **[TPS_SYST_02061] Routing of IPdus that are part of a ContainerIPdu** ⌈IPdus that are part of a ContainerIPdu shall have a PduTriggering and IPduPorts since they are handled by the PduR.⌋

The following rule applies to the creation of ISignalTriggering and ISignalPort:

[TPS_SYST_01058] Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content ⌈In case of a Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content, the reference between the ISignalTriggerings (that are referred to by the PduTriggering in the role iSignalTriggering) and the respective ISignalPorts shall not be created.⌋

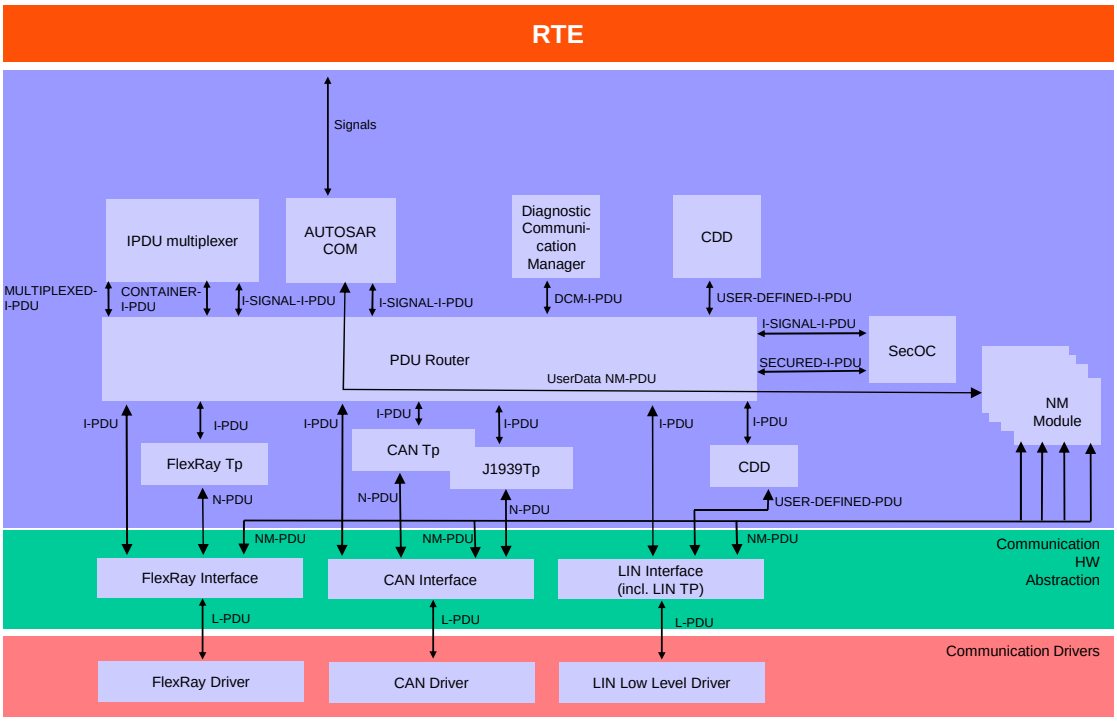

Figure 6.3: AUTOSAR Layered Architecture

Class	***CommConnectorPort*** (abstract)			
Note	The Ecu communication relationship defines which signals, Pdus and frames are actually received and transmitted by this ECU. For each signal, Pdu or Frame that is transmitted or received and used by the Ecu an association between an ISignalPort, IPduPort or FramePort with the corresponding Triggering shall be created. An ISignalPort shall be created only if the corresponding signal is handled by COM (RTE or Signal Gateway). If a Pdu Gateway ECU only routes the Pdu without being interested in the content only a FramePort and an IPduPort needs to be created.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	FramePort, IPduPort, ISignalPort			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..1	attr	Communication Direction of the Connector Port (input or output Port).

Table 6.1: CommConnectorPort

[constr_9103] Existence of `communicationDirection`

Imposition time: CP: `IT_SysDesc`

⌈For each `CommConnectorPort`, the attribute `communicationDirection` shall exist.⌋

Class	**FramePort**			
Note	Connectors reception or send port on the referenced channel referenced by a FrameTriggering.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.2: FramePort

Class	**IPduPort**			
Note	Connectors reception or send port on the referenced channel referenced by a PduTriggering.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduSignal Processing	IPduSignalProcessing Enum	0..1	attr	Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.
rxSecurity Verification	Boolean	0..1	attr	This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.
timestampRx Acceptance Window	TimeValue	0..1	attr	This attribute is used to define the maximum allowed deviation in seconds from the expected timestamp for which a SecuredIPdu is still deemed authentic. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
useAuthData Freshness	Boolean	0..1	attr	This attribute describes whether a part of AuthenticPdu contained in a SecuredIPdu shall be passed on to the SWC that verifies and generates the Freshness. The part of the Authentic-PDU is defined by the authData FreshnessStartPosition and authDataFreshnessLength.

Table 6.3: IPduPort

[constr_3137] `IPduPort.rxSecurityVerification` is configurable on the receiver side

Imposition time: CP: `IT_SysDesc`

⌈The `IPduPort.rxSecurityVerification` attribute shall only be used in `IPduPort`s with the `communicationDirection` = in.⌋

[constr_3138] `IPduPort.rxSecurityVerification` validness

Imposition time: CP: `IT_SysDesc`

⌈The `IPduPort.rxSecurityVerification` information is only valid for `SecuredIPdu`s.⌋

[constr_3337] `IPduPort.useAuthDataFreshness` is configurable on the receiver side

Imposition time: CP: `IT_SysDesc`

⌈The `IPduPort.useAuthDataFreshness` attribute shall only be used in `IPduPort`s with the `communicationDirection` = in.⌋

[constr_3338] IPduPort.useAuthDataFreshness validness

Imposition time: CP: IT_SysDesc

⌈The IPduPort.useAuthDataFreshness information is only valid for Secured-IPdus.⌋

Enumeration	**IPduSignalProcessingEnum**
Note	Definition of signal processing modes.
Aggregated by	IPduPort.iPduSignalProcessing
Literal	***Description***
deferred	The signal indications / confirmations are deferred. **Tags:** atp.EnumerationLiteralIndex=0
immediate	The signal indications / confirmations are performed. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.4: IPduSignalProcessingEnum

Class	**ISignalPort**			
Note	Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or DataFilters for ISignals need to be specified several ISignalPorts may be created.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	0..1	aggr	Optional specification of a signal COM filter at the receiver side in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec. In this case the ReceiverComSpec overrides this optional specification. **Tags:** atp.Splitkey=dataFilter
ddsQosProfile	DdsCpQosProfile	0..1	ref	Reference to the DDS Qos profile used for this ISignal. **Tags:** atp.Splitkey=ddsQosProfile atp.Status=candidate
firstTimeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. • ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.

▽

△

Class	**ISignalPort**			
timeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverComSpec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. • ISignalPort with communicationDirection = out: Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the SenderComSpec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the System Signal and the value is available in the configured Sender ComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. This attribute can be used in the following cases: • legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the DataMapping is missing. • bus monitoring use cases in which the DataMapping is ignored.

Table 6.5: ISignalPort

Enumeration	**HandleInvalidEnum**
Note	Strategies of handling the reception of invalidValue.
Aggregated by	InvalidationPolicy.handleInvalid, ISignalPort.handleInvalid
Literal	***Description***
dontInvalidate	Invalidation is switched off. **Tags:** atp.EnumerationLiteralIndex=0
external Replacement	Replace a received invalidValue. The replacement value is sourced from the aggregation in the role replaceWith. **Tags:** atp.EnumerationLiteralIndex=1
keep	The application software is supposed to handle signal invalidation on RTE API level either by Data ReceiveErrorEvent or check of error code on read access. **Tags:** atp.EnumerationLiteralIndex=2
replace	Replace a received invalidValue. The replacement value is specified by the initValue. **Tags:** atp.EnumerationLiteralIndex=3

Table 6.6: HandleInvalidEnum

[TPS_SYST_01059] Relationship between `FrameTriggering` and `CommConnectorPort` ⌈For the reference between `FrameTriggering` and `FramePort` two approaches are supported:

- One to One relationship between FrameTriggering and FramePort per EcuInstance
- One FramePort per communicationDirection per EcuInstance exists and is referenced by all applicable FrameTriggerings (n to 1).

⌋

[TPS_SYST_01060] Relationship between PduTriggering and CommConnectorPort ⌈For the reference between PduTriggering and IPduPort two approaches are supported:

- One to One relationship between PduTriggering and IPduPort per EcuInstance
- One IPduPort per communicationDirection per EcuInstance exists and is referenced by all applicable PduTriggerings (n to 1).

⌋

[TPS_SYST_01061] Relationship between ISignalTriggering and CommConnectorPort ⌈For the reference between ISignalTriggering and ISignalPort two approaches are supported:

- One to One relationship between ISignalTriggering and ISignalPort per EcuInstance
- One ISignalPort per communicationDirection per timeout per EcuInstance exists and is referenced by all applicable PduTriggerings (n to 1).

⌋

[TPS_SYST_02208] ISignalPort.handleInvalid defines the reception invalidation behavior ⌈The attribute ISignalPort.handleInvalid defines the behavior during signal reception if the respective ISignal's invalidValue is received. The ISignal is assigned to this ISignalPort via the ISignalTriggering.⌋

[TPS_SYST_02209] Not defined ISignalPort.handleInvalid behavior ⌈If the attribute ISignalPort.handleInvalid is not defined then the value dontInvalidate shall be assumed.⌋

[constr_5053] Existence of ISignalPort.handleInvalid

Imposition time: CP: IT_SysDesc

⌈If the ISignalPort has a networkRepresentationProps.invalidValue defined then the ISignalPort.communicationDirection shall equal in.⌋

[constr_5054] `externalReplacement` not applicable for `ISignalPort.handleInvalid`

Imposition time: CP: `IT_SysDesc`

⌈In the context of `ISignalPort.handleInvalid` the value `externalReplacement` shall not be used.⌋

The action `externalReplacement` can only be implemented by the RTE. Thus it is required to have a ComSpec definition.

[TPS_SYST_02210] Data invalidation in case the `dataTypePolicy` is set to `override` or `legacy` ⌈If the `dataTypePolicy` of an `ISignal` is set to `override` or `legacy`, the `ISignalPort.handleInvalid` attribute defines the data invalidation.⌋

6.1.1 Ownership of `PduTriggering`s

`PduTriggering`s are owned by `PhysicalChannel`s. A `PduTriggering` can not exist on its own, it has to be aggregated by exactly one `PhysicalChannel` (see figure 6.2). In figure 6.4 the example of section 6.11.2.1.2 is taken to illustrate the ownership of `PduTriggering`s.

If a `SecuredIPdu` is transmitted on different `PhysicalChannel`s this can be described in a System Description by a single `SecuredIPdu` element that is referenced by `PduTriggering`s of the different `PhysicalChannel`s (*PduTriggeringCan* and *PduTriggeringFr*).

The payload of the `SecuredIPdu` is described by exactly one `PduTriggering` (*PduTriggeringSec*) that is referenced in the role `SecuredIPdu.payload`.

Note on the right side that, although the payload `PduTriggering` (*PduTriggeringSec*) will also be transmitted on the same `PhysicalChannel`s as the `SecuredIPdu`, it is completely sufficient that the payload `PduTriggering` (*PduTriggeringSec*) that is referenced by the `SecuredIPdu` is aggregated by one of the `PhysicalChannel`s on which the `SecuredIPdu` is transmitted. It is irrelevant which `PduTriggering` of which `PhysicalChannel` is chosen to describe the payload of the `SecuredIPdu` since the assignment of the `SecuredIPdu` to `PhysicalChannel`s is done only by the `PduTriggering`s that are pointing to the `SecuredIPdu`. Thus either the *orange* or the *red* aggregation represent the ownership of the `PduTriggering` (*PduTriggeringSec*).

This can be motivated by the left part of figure 6.4: The path from RTE, via COM, PduR, SecOC, and back to the PduR has just one flow. The fan-out actually happens only after the message has entered the PduR from the SecOC, only then there are two paths to the target networks.

The reason for this modeling is the configuration of the PduR. One use case in the PduR is the enabling and disabling of PDUs by PdurIPduGroups. The PdurIPduGroup works on PduTriggerings and there is also the use case to enable/disable the SecuredIPdus in the PduR.

A similar example can also be observed in the context of ContainerIPdus as discussed in section 6.11.2.1.3.

[TPS_SYST_03117] Only one PduTriggering for the payload of a ContainerIPdu or SecuredIPdu in case of fan-out ⌈If a PduTriggering is defined as payload of a ContainerIPdu or SecuredIPdu and this ContainerIPdu or SecuredIPdu is fan-out to several PhysicalChannels, then it is sufficient that the payload PduTriggering is aggregated by one of the target PhysicalChannels.

It is irrelevant by which PhysicalChannel the payload PduTriggering is aggregated, as long as one of the transporting PhysicalChannels aggregates the payload PduTriggering.⌋

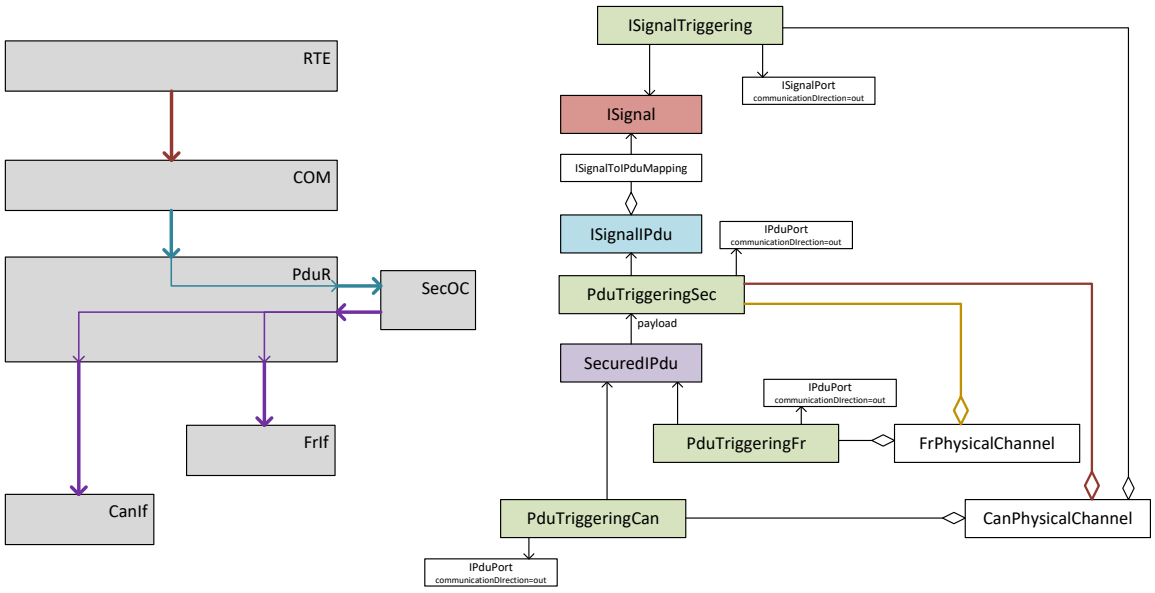

Figure 6.4: Ownership of PduTriggerings in case of fan-out

Although [TPS_SYST_03117] claims that it is irrelevant which PhysicalChannel aggregates the payload PduTriggering, there are scenarios where the surrounding modeling might be influenced by the choice which PhysicalChannel aggregates the payload PduTriggering. The example illustrated in figure 6.128 can be modeled in two ways, this is shown in figure 6.5.

If the *FrPhysicalChannel* aggregates the *PduTriggeringSec*, then it is required to model another PduTriggering which is then aggregated by the *CanPhysicalChannel* (illustrated in orange).

If already the *CanPhysicalChannel* aggregates the *PduTriggeringSec* (illustrated in red), then there is no need to have another `PduTriggering`, because it is already modeled that *PduTriggeringSec* shall be transported on the *CanPhysicalChannel*.

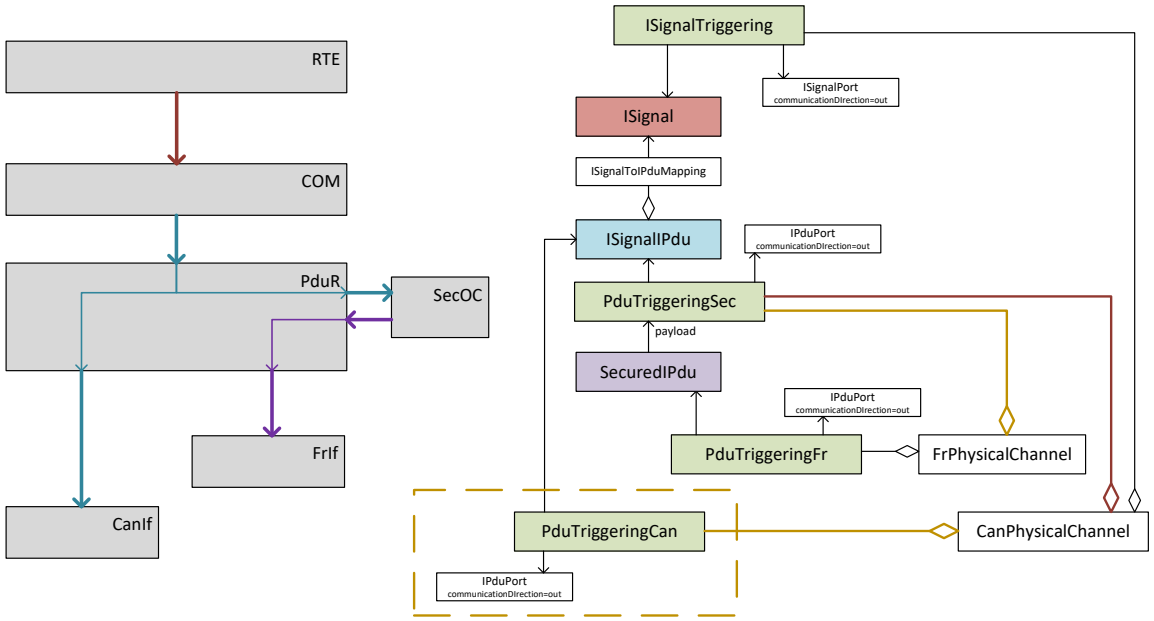

Figure 6.5: Alternative ownership of `PduTriggering`s in case of fan-out

6.1.2 Ownership of `ISignalTriggering`s

`ISignalTriggering`s are owned by `PhysicalChannel`s. An `ISignal-Triggering` can not exist on its own, it has to be aggregated by exactly one `PhysicalChannel` (see figure 6.2). In figure 6.6 the example of section 6.11.2.1.1 is taken to illustrate the ownership of `ISignalTriggering`s.

Note on the right side that, although the `ISignal` will also be transmitted on the same `PhysicalChannel`s as the `ISignalIPdu`, it is completely sufficient that the `ISignalTriggering` that is referenced by the `ISignalIPdu` is aggregated by one of the `PhysicalChannel`s on which the `ISignalIPdu` is transmitted. It is irrelevant which `ISignalTriggering` of which `PhysicalChannel` is chosen to describe the existence of the `ISignalTriggering`. Thus either the *orange* or the *red* aggregation represent the ownership of the `ISignalTriggering`.

This can be motivated by the left part of figure 6.6: The path from RTE, via COM, and to the PduR has just one flow. The fan-out actually happens only after the message has entered the PduR, only then there are two paths to the target networks.

[TPS_SYST_03118] Only one `ISignalTriggering` in case of PDU fan-out ⌈If an `ISignalIPdu` is defined to be fan-out to several `PhysicalChannel`s, then it is

sufficient that the ISignalTriggerings of that ISignalIPdu are aggregated by one of the target PhysicalChannels.

It is irrelevant by which PhysicalChannel the ISignalTriggerings are aggregated, as long as one of the transporting PhysicalChannels aggregates the ISignalTriggerings.

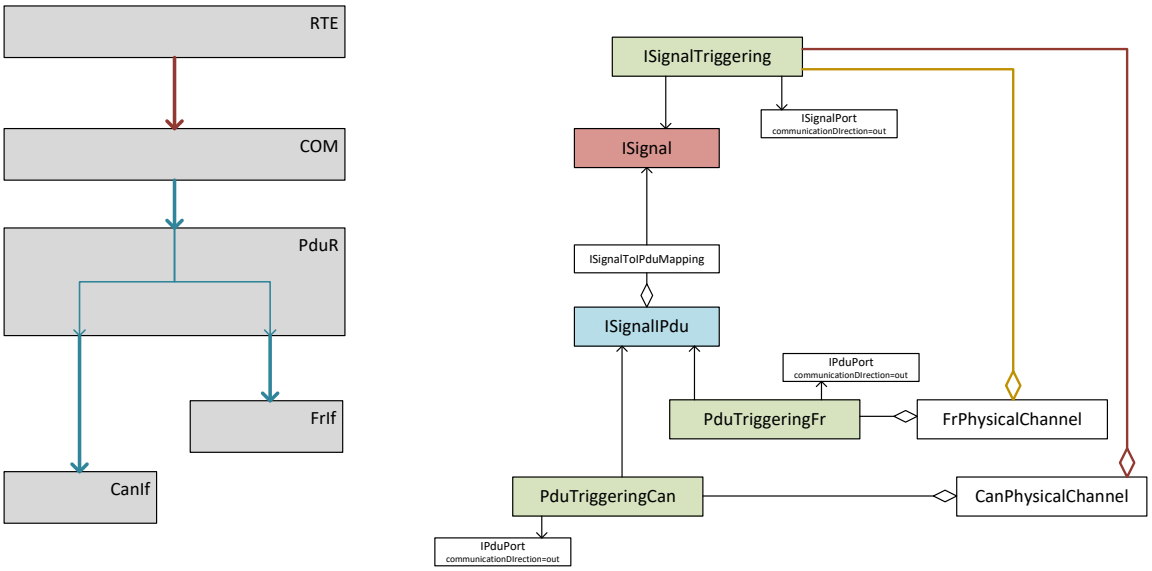

Figure 6.6: Ownership of ISignalTriggerings in case of fan-out

6.2 ISignals

SystemSignals can be defined independently of CommunicationClusters and are representing the VariableDataPrototypes, ArgumentDataPrototypes, Triggers and ModeDeclarationGroupPrototypes in the communication description.

The RTE supports a "signal fan-out" where the same signal (System Signal) is sent in different IPdus to multiple receivers. The Pdu Router supports the "PDU fan-out" where the same IPdu is sent to multiple destinations.

To support the "signal fan-out" ISignals and ISignalGroups are introduced. An ISignal(ISignalGroup) represents the SystemSignal(SystemSignalGroup) of the Interaction Layer.

In case of "signal fan-out", several ISignals in different IPdus refer to the same SystemSignal. The "Signal fan-out" will be executed by the RTE. ISignals describe the Interface between the precompile configured RTE and the potentially postbuild configured Com Stack.

The ISignalToIPduMapping element describes the mapping of ISignals to ISignalIPdus and defines the position of an ISignal within an ISignalIPdu.

[constr_3009] Overlapping of ISignals is prohibited

Imposition time: CP: IT_SysDesc

⌈ISignals mapped to an ISignalIPdu shall not overlap.⌋

[constr_5253] Value range of ISignal.length

Imposition time: CP: IT_SysDesc

⌈The value of ISignal.length shall be in the range of 0..34359738360 Bits.⌋

[constr_3010] ISignalIPdu length shall not be exceeded

Imposition time: CP: IT_SysDesc

⌈The combined length of all ISignals and updateIndicationBitPositions that are mapped into an ISignalIPdu shall not exceed the defined Pdu length.⌋

[constr_3011] Overlapping of updateIndicationBits of ISignals is prohibited

Imposition time: CP: IT_SysDesc

⌈The updateIndicationBitPosition for an ISignal in an ISignalIPdu shall not overlap with other updateIndicationBitPositions or ISignal locations.⌋

[TPS_SYST_01062] Network representation of an ISignal ⌈With the aggregation of SwDataDefProps in the role networkRepresentationProps the actual representation of the ISignal on the network can be specified.⌋

[TPS_SYST_01063] Context of network representation of an ISignal ⌈The dataTypePolicy defines from which context the network representation specification shall be taken.⌋

For an alternative network representation it is important to define an alternative SwDataDefProps especially SwBaseType defining alternative encoding (e.g. from float in PortInterface to integer on bus).

[constr_3060] Allowed Attributes for networkRepresentationProps and physicalProps

Imposition time: CP: IT_SysDesc

⌈

Attributes of SwDataDefProps	SystemSignal.physicalProps	ISignal.networkProps
additionalNativeTypeQualifier	NA	NA
annotation	NA	NA
baseType	NA	D
baseType.category	NA	M
BaseTypeDirectDefinition.baseTypeEncoding	NA	D
BaseTypeDirectDefinition.byteOrder	NA	NA
BaseTypeDirectDefinition.baseTypeSize	NA	0..1
BaseTypeDirectDefinition.memAlignment	NA	NA
BaseTypeDirectDefinition.nativeDeclaration	NA	NA
compuMethod	D	I
dataConstr	D	M
displayFormat	D	M
implementationDataType	NA	NA
invalidValue	NA	D
stepSize	NA	NA
swAddrMethod	NA	NA
swAlignment	NA	NA
swBitRepresentation	NA	NA
swCalibrationAccess	NA	NA
swCalprmAxisSet	NA	NA
swComparisonVariable	NA	NA
swDataDependency	NA	NA
swHostVariable	NA	NA
swImplPolicy	NA	NA
swIntendedResolution	NA	NA
swInterpolationMethod	NA	NA
swIsVirtual	NA	NA
swPointerTargetProps	NA	NA
swRecordLayout	NA	NA
swRefreshTiming	NA	NA
swTextProps	D	NA
swValueBlockSize	NA	NA
unit	D	M

▽

△

valueAxisDataType	NA	NA

⌋

The following settings apply in [constr_3060]:

D Define the attribute independent from settings to the left.

I Inherit the definition from the left for usage in the scope of this element. This means that the information is taken over in the respective context without further ARXML configuration. The attribute of the SwDataDefProps shall not exist on the right side.

NA Attribute is **not applicable** for usage in the scope of this element.

M Attribute is **meaningless** in the scope of this element. As it was allowed in previous versions, declaring it as Not Applicable (NA) would break compatibility. Tools shall ignore such an attribute without a warning.

In case that the System Description doesn't use a complete Software Component Description (VFB View) the `physicalProps` and `networkRepresentationProps` are used to configure the Data Semantics.

The `networkRepresentationProps` contains a reference to the `SwBaseType`. This reference can be used for the derivation of the `ComSignalType` in the AUTOSAR Com Configuration.

Please note that a `DataTransformation` that is based on the network representation is explained in more detail in chapter 7.3.2.2.1. This chapter also contains an explanation that describes a data conversion based on `CompuMethod`s (see section 7.3.2.2.1.1 for more details).

[TPS_SYST_02001] `networkRepresentationProps` are mandatory in case the `dataTypePolicy` is set to `override` or `legacy` ⌈If the `dataTypePolicy` of an `ISignal` is set to `override` or `legacy`, the `networkRepresentationProps` for the respective `ISignal` have to be specified.⌋

[TPS_SYST_02006] Usage of `networkRepresentationFromComSpec` ⌈If the `networkRepresentationFromComSpec` is used either the `SwDataDefProps` in the role `networkRepresentation` aggregated by the `SenderComSpec` or `ReceiverComSpec` shall exist or the `ImplementationDataType` shall exist.⌋

Please note that some `category`s of `CompuMethod` cannot be successfully converted to A2L [24] because A2L does not provide an equivalent semantics that comes close to the respective AUTOSAR semantics.

A prominent example for such a case is a `CompuMethod` of category `SCALE_LINEAR_AND_TEXTTABLE` that actually has more than one linear interval and a texttable part.

[TPS_SYST_02079] Identification of ImplementationDataType for a given ISignal in an Ecu Extract ⌈

1. From the ISignal go to the referenced SystemSignal
2. Find all DataMappings that refer to the SystemSignal
3. For all VariableDataPrototypes referenced by the applicable DataMappings
   (a) If the VariableDataPrototype is typed by an ApplicationDataType and belongs to a CompositionSwComponentType then for all DataTypeMappingSets referenced by the CompositionSwComponentType find the DataTypeMap that refers to this ApplicationDataType. The DataTypeMap also refers to the wanted ImplementationDataType.
   (b) If the VariableDataPrototype is typed by an ApplicationDataType and belongs to an AtomicSwComponentType then for all DataTypeMappingSets referenced by the InternalBehavior of the AtomicSwComponentType find the DataTypeMap that refers to this ApplicationDataType. The DataTypeMap also refers to the wanted ImplementationDataType.
   (c) If the VariableDataPrototype is typed by an ImplementationDataType then the ImplementationDataType is the wanted one.

⌋

[constr_3199] ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation

Imposition time: CP: IT_SysDesc

⌈In a complete model every ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation.⌋

[constr_9398] dataTypePolicy = transformingISignal not applicable for ISignals that are transformed by the ComBasedTransformer

Imposition time: CP: IT_SysDesc

⌈If an ISignal is referenced by an ISignalGroup in the role iSignal and this ISignalGroup references the DataTransformation in the role comBasedSignalGroupTransformation then the ISignal is not allowed to set the dataTypePolicy to transformingISignal.⌋

The reason for [constr_9398] is that the dataTypePolicy = transformingISignal is applicable only for ISignals that are transformed by an algorithmic transformer like SOME/IP but not for the configuration based ComBasedTransformer.

[constr_9358] `ISignal` that references a `DataTransformation` shall have the `dataTypePolicy` set to `transformingISignal`

Imposition time: CP: `IT_SysDesc`

⌈In a complete model every `ISignal` that references the `DataTransformation` in the role `dataTransformation` shall have the `dataTypePolicy` set to `transformingISignal`, `ddsSignal`, or `ddsService`.⌋

[constr_3736] `ISignal` that has `dataTypePolicy` set to `ddsSignal` shall be referenced by a `DdsCpISignalToDdsTopicMapping`

Imposition time: CP: `IT_SysDesc`

⌈Every `ISignal` that has `dataTypePolicy` set to `ddsSignal` shall be referenced by a `DdsCpISignalToDdsTopicMapping`.⌋

[constr_3737] `ISignal` referenced from `DdsCpISignalToDdsTopicMapping`

Imposition time: CP: `IT_SysDesc`

⌈Every `ISignal` that has `dataTypePolicy` set to any value different to `ddsSignal` shall NOT be referenced by a `DdsCpISignalToDdsTopicMapping`.⌋

[constr_9349] An `ISignal` shall not be used for SOME/IP and DDS transport at the same time

Imposition time: CP: `IT_SysDesc`

⌈An `ISignal`

- shall not be referenced by `DdsCpISignalToDdsTopicMapping` in the role `iSignal` and
- shall not aggregate `SOMEIPTransformationISignalProps` in the role `transformationISignalProps`

at the same time.⌋

DDS Data is created in the application layer and passed to DDS directly (as unserialised data): neither modification nor transformation (and vice versa at receiver side) should be performed on data. Serialization is performed inside the Dds BSW and it is completely opaque to the AUTOSAR stack.

[TPS_SYST_03133] Pass-through DDS transformer ⌈If an `ISignal` has `dataTypePolicy` set to `ddsSignal` or `ddsService`, then there will be a raw copy of the data to the `ISignal` and `ISignalIPdu` by the DDS transformer and the data will arrive unmodified in the DDS basic software module.⌋

[TPS_SYST_01065] Mapping onto the `ComSignalType` enumeration ⌈

BaseTypeEncoding	*BaseTypeSize*	*ISignal.iSignalType*	*SystemSignal.dynamicLength*	*ComSignalType*
2C	8 Bits	primitive	not applicable	SINT8, ComBitSize derived from ISignal.length
2C	not available	primitive	not applicable	SINT8 if ISignal.length <= 8. ComBitSize derived from ISignal.length
2C	16 Bits	primitive	not applicable	SINT16, ComBitSize derived from ISignal.length
2C	not available	primitive	not applicable	SINT16 if ISignal.length > 8 and <= 16. ComBitSize derived from ISignal.length
2C	32 Bits	primitive	not applicable	SINT32, ComBitSize derived from ISignal.length
2C	not available	primitive	not applicable	SINT32 if ISignal.length > 16 and <= 32. ComBitSize derived from ISignal.length
2C	64 Bits	primitive	not applicable	SINT64, ComBitSize derived from ISignal.length
2C	not available	primitive	not applicable	SINT64 if ISignal.length > 32 and <= 64. ComBitSize derived from ISignal.length
NONE	8 Bits	primitive	not applicable	UINT8, ComBitSize derived from ISignal.length
NONE	not available	primitive	not applicable	UINT8 if ISignal.length <= 8. ComBitSize derived from ISignal.length
NONE	16 Bits	primitive	not applicable	UINT16, ComBitSize derived from ISignal.length
NONE	not available	primitive	not applicable	UINT16 if ISignal.length > 8 and <= 16. ComBitSize derived from ISignal.length
NONE	32 Bits	primitive	not applicable	UINT32, ComBitSize derived from ISignal.length
NONE	not available	primitive	not applicable	UINT32 if ISignal.length > 16 and <= 32. ComBitSize derived from ISignal.length
NONE	64 Bits	primitive	not applicable	UINT64, ComBitSize derived from ISignal.length

▽

△

BaseTypeEncoding	*BaseTypeSize*	*ISignal.iSignalType*	*SystemSignal.dynamicLength*	*ComSignalType*
NONE	not available	primitive	not applicable	UINT64 if ISignal.length > 32 and <= 64. ComBitSize derived from ISignal.length
IEEE754	32 Bits	primitive	not applicable	FLOAT32, ComBitSize derived from ISignal.length
IEEE754	not available	primitive	not applicable	FLOAT32, ISignal.length = 32. ComBitSize derived from ISignal.length
IEEE754	64 Bits	primitive	not applicable	FLOAT64, ComBitSize derived from ISignal.length
IEEE754	not available	primitive	not applicable	FLOAT64, ISignal.length = 64. ComBitSize derived from ISignal.length
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	8 Bits	array	false	UINT8_N, ComSignalLength derived from ISignal.length
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	not available	array	false	UINT8_N, ComSignalLength derived from ISignal.length
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	8 Bits	array	true	UINT8_DYN, ComSignalLength derived from ISignal.length
NONE, ISO-8859-1, ISO-8859-2, WINDOWS-1252, UTF-8, UTF-16, UCS-2	not available	array	true	UINT8_DYN, ComSignalLength derived from ISignal.length
BOOLEAN	ignored	primitive	not applicable	BOOLEAN

The table describes the mapping of baseTypeSize, baseTypeEncoding, ISignal.iSignalType and SystemSignal.dynamicLength onto the ComSignalType enumeration

⌋

In other words [TPS_SYST_01065] focuses only on the derivation of the ComSignalType. This table shall not be taken as a source to derive requirements on the modeling of SwBaseTypes used on the level of the RTE.

The setting "not applicable" for an Attribute in [TPS_SYST_01065] means that no value shall be set for this Attribute. The setting "ignored" for an Attribute in [TPS_SYST_01065] means than any value is accepted for this Attribute, but the value will be ignored in creation of the ECU configuration value file.

[constr_3258] Restriction on `ISignal.length` in case `iSignalType` is set to `array`

Imposition time: CP: `IT_SysDesc`

⌈If `ISignal.iSignalType` is set to `array` then `ISignal.length` shall be a multiple of 8.⌋

[TPS_SYST_02111] `VariableDataPrototype` in case `ISignal.iSignalType` is set to `array` ⌈If `ISignal.iSignalType` is set to `array` the corresponding `VariableDataPrototype` shall boil down to an Array according to [TPS_SYST_02083], [TPS_SYST_02084], [TPS_SYST_02085] and [TPS_SYST_02089].⌋

The `invalidValue` is aggregated by the `SwDataDefProps` element. The `SwDataDefProps` and the `SwBaseType` classes are described in more detail in the Software Component Template [4].

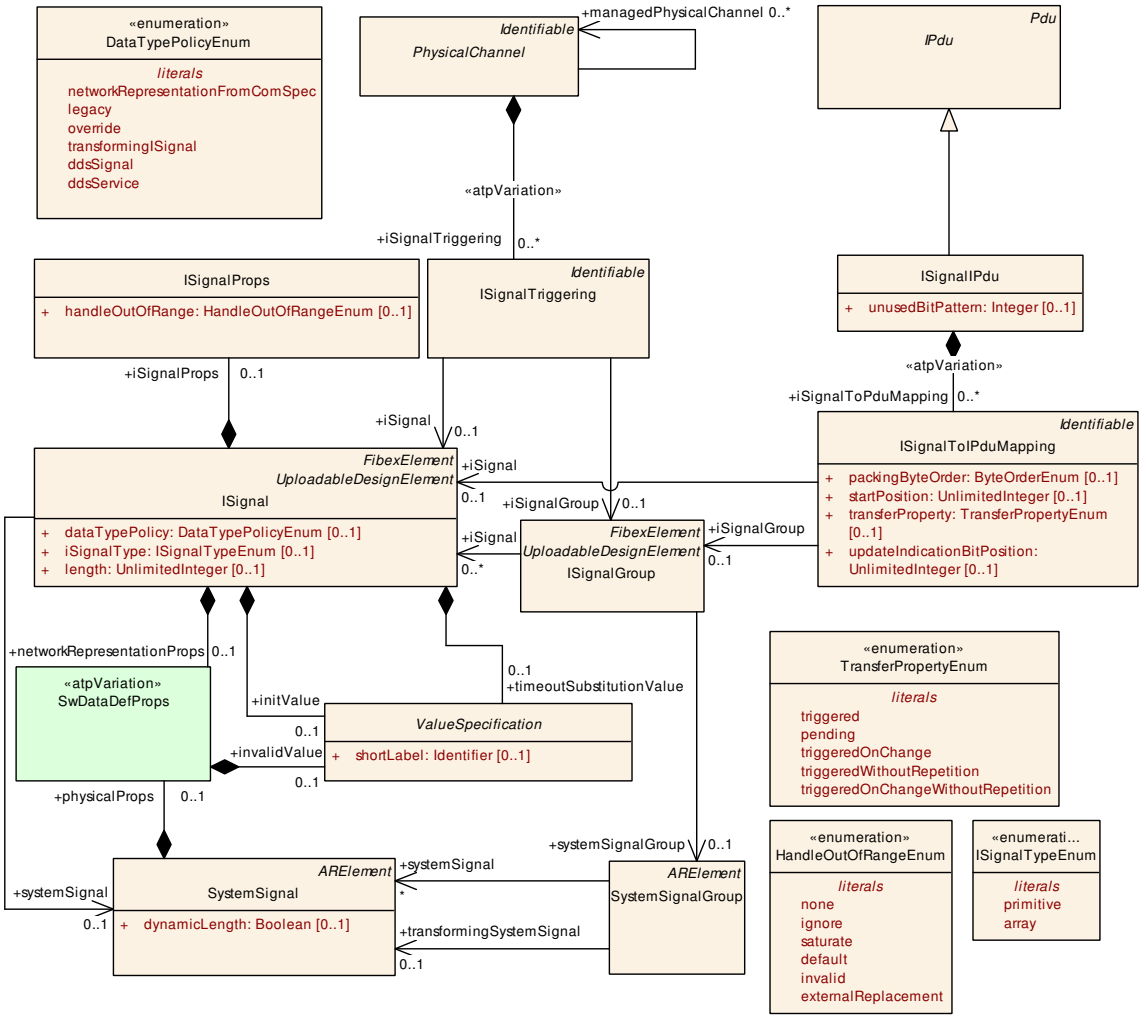

Figure 6.7: ISignals and the mapping into IPdus (FibexCore: SignalOverview)

The configuration of the COM Module for atomic signals can largely be derived from the System Template.

[TPS_SYST_01066] Derivation of Tx COM Signals ⌈A ComSignal shall be defined in the COM module configuration for each ISignalToIPduMapping that is aggregated by ISignalIPdu that in turn is referenced by a PduTriggering that in turn references an IPduPort where the communicationDirection is set to out of the regarded ECU.

Exception: If the ISignal is part of a Signal Gateway relation (ISignalMapping.targetSignal pointing to an ISignalTriggering referencing this ISignal) the creation of a ComSignal is not mandated if

- the ISignal does not point to a SystemSignal that is referenced by a DataMapping (application does not send the gatewayed signal content) or
- the ISignal points to a SystemSignal that is referenced by a DataMapping in which the RPortPrototype is used as the context element and the destination ISignalTriggering.iSignalPort.communicationDirection equals out (application sends the gatewayed signal content) or
- the ISignalToIPduMapping.iSignal.dataTypePolicy is set to legacy.

In these cases the configuration of ComGwMapping can be done by means of ComGwSourceDescription and ComGwDestinationDescription. However it is possible to create ComSignals for the ComGwSignal approach as well (i.e., even if the application does not require access to the respective SystemSignal).

Exception2: If the ISignal is part of an ISignalIPdu that in turn is part of a Pdu Gateway relation (IPduMapping.targetIPdu pointing to a PduTriggering referencing this ISignalIPdu) the creation of the ComSignal is not mandated if

- Application does not send the gatewayed signal content: the ISignal does not point to a SystemSignal that is referenced by a DataMapping in which the PPortPrototype is used as the context element or
- Legacy use case is configured in which ISignal is processed by Com without using of PortPrototypes: the ISignal has the ISignalToIPduMapping.iSignal.dataTypePolicy set to legacy and in addition the ISignal is referenced by an ISignalTriggering that in turn references an ISignalPort with communicationDirection = out.

⌋

Please note that Exception 2 supports runtime variants in which the ISignalIPdu is received by the PduR from a BusIf and routed to a different BusIf in one variant or in a second variant a local Application produces data and the ISignalIPdu is sent via Com to the BusIf. Such an example is shown in the figure below:

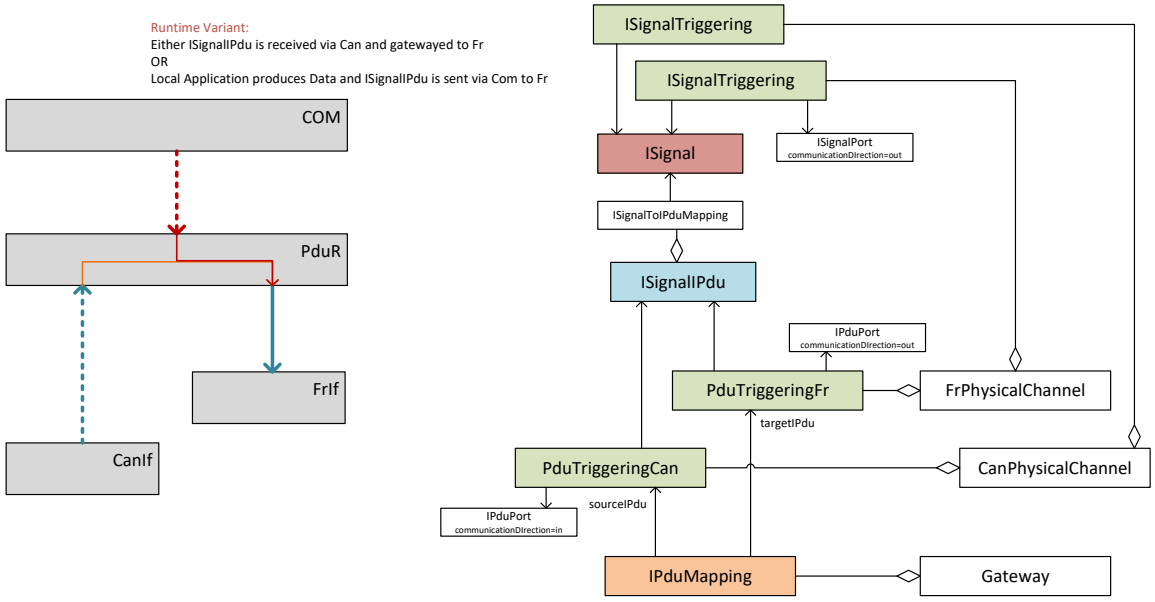

Figure 6.8: Runtime Gw Variant example

[TPS_SYST_01067] Derivation of Rx COM Signals ⌈A ComSignal shall be defined in the COM module configuration for each ISignalToIPduMapping that is aggregated by ISignalIPdu that in turn is referenced by a PduTriggering that in turn references an IPduPort where the communicationDirection is set to in in the regarded ECU.

Exception: If the ISignal is part of a Signal Gateway relation (ISignalMapping.sourceSignal pointing to an ISignalTriggering referencing this ISignal) the creation of a ComSignal is not mandated if

- the ISignal does not point to a SystemSignal that is referenced by a DataMapping (application is not interested in the gatewayed signal content) or
- the ISignal points to a SystemSignal that is referenced by a DataMapping in which the PPortPrototype is used as the context element and source ISignalTriggering.iSignalPort.communicationDirection equals in (application is not interested in the gatewayed signal content) or
- the ISignalToIPduMapping.iSignal.dataTypePolicy is set to legacy.

In these cases the configuration of ComGwMapping can be done by means of ComGwSourceDescription and ComGwDestinationDescription. However it is possible to create ComSignals for the ComGwSignal approach as well (i.e., even if the application does not require access to the respective SystemSignal).

Exception2: If the ISignal is part of an ISignalIPdu that in turn is part of a Pdu Gateway relation (IPduMapping.sourceIPdu pointing to a PduTriggering referencing this ISignalIPdu) the creation of the ComSignal is not mandated if

- Application does not receive the gatewayed signal content: the ISignal does not point to a SystemSignal that is referenced by a DataMapping in which the RPortPrototype is used as the context element or
- Legacy use case is configured in which ISignal is processed by Com without using of PortPrototypes: the ISignal has the ISignalToIPduMapping.iSignal.dataTypePolicy set to legacy and in addition the ISignal is referenced by an ISignalTriggering that in turn references an ISignalPort with communicationDirection = in.

⌋

The following table summarizes the derivation rules for ComSignals that are described by [TPS_SYST_01066] and [TPS_SYST_01067].

Scenario	ComSignal derivation rule
Transmission by Application (DataMapping defined)	ComSignal needed
Reception by Application (DataMapping defined)	ComSignal needed
Signal Gateway	ComSignal needed
Pdu Gateway	no ComSignal needed
Pdu Gateway and Reception by Application	Rx ComSignal needed, no Tx ComSignal needed
Pdu Gateway and Transmission by Application	Tx ComSignal needed, no Rx ComSignal needed
ISignalToIPduMapping.iSignal.dataTypePolicy is set to legacy and ISignalTriggering.iSignalPort.communicationDirection equals out	Tx ComSignal needed
ISignalToIPduMapping.iSignal.dataTypePolicy is set to legacy and ISignalTriggering.iSignalPort.communicationDirection equals in	Rx ComSignal needed

Overview of ComSignal derivation rules

To support the AUTOSAR concept of composite data types the AUTOSAR COM layer provides signal groups. Every record or array element of a composite data type requires a SystemSignal for the transmission. But the RTE has to guarantee the consistent transmission of data.

[TPS_SYST_01153] Atomic transport of SystemSignalGroups ⌈A SystemSignalGroup shall be transmitted and received consistently; therefore it provides data consistency for composite data types.⌋

A SystemSignalGroup refers to a set of SystemSignals that shall always be kept together in a common IPdu. An ISignalGroup represents a SystemSignalGroup of the Interaction Layer. In the case of "signal fan-out", several ISignalGroups refer to the same SystemSignalGroup.

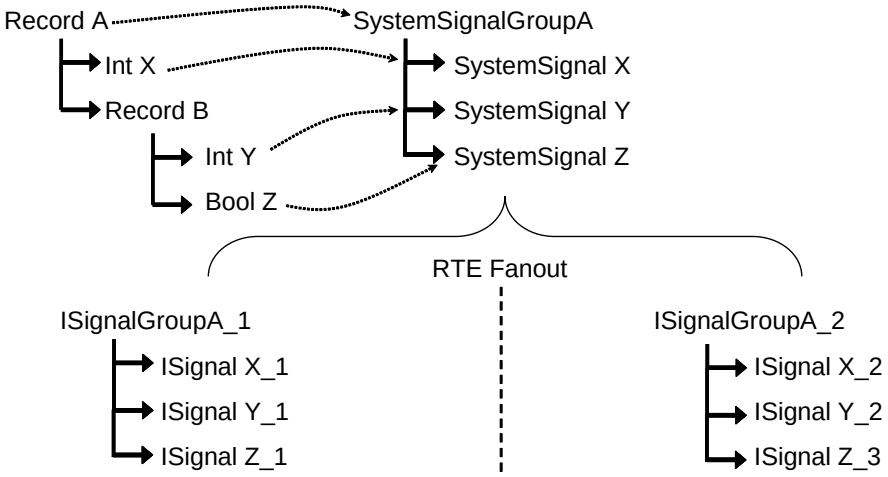

Figure 6.9: ISignal example

The example in Figure 6.9 shows the usage of ISignalGroups and ISignals. In this example a record is mapped to a SystemSignalGroup. All ApplicationRecordElements with ApplicationPrimitiveDataType are mapped to individual SystemSignals. If the same SystemSignalGroup is sent to different receivers (RTE Fanout) then two different ISignalGroups are created. For each SystemSignal within the SystemSignalGroup an ISignal is created. The different ISignals of the same SystemSignal can have different network representations.

[constr_3094] Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals

Imposition time: CP: IT_SysDesc

⌈In case the ISignals contained in an ISignalGroup are referenced by an ISignalTriggering, the communicationDirection of the ISignalPort referenced by the ISignal's ISignalTriggering shall be identical to the communicationDirection of the ISignalPort referenced by the containing ISignalGroup's ISignalTriggering.⌋

Please note that not all ISignals that are part of the ISignalGroup need to have a reference to an ISignalPort via an ISignalTriggering as described by [TPS_SYST_02106].

[TPS_SYST_01157] Allowed usage of attributes for ISignals, ISignalGroups and GroupSignals ⌈

Attributes	ISignal	ISignalGroup	GroupSignal
startPosition	1	NA	1
updateIndicationBitPosition	0..1	0..1	NA
transferProperty	0..1	0..1	0..1
packingByteOrder	1	NA	1

▽

△

dataFilter	0..1	NA	0..1

Attributes that may be used to configure ISignals in different roles (ISignals that are not part of an ISignalGroup and ISignals that are part of an ISignalGroup) and ISignalGroups.

⌋

[constr_3067] initValue defined in the context of ISignal

Imposition time: CP: IT_SysDesc

⌈The definition of an initValue in the context of an ISignal shall only be a NumericalValueSpecification, TextValueSpecification or ArrayValueSpecification that aggregates elements of type NumericalValueSpecification or TextValueSpecification.⌋

Please note that for ISignals that are referencing a SystemSignal with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE the [constr_3067] is further restricted by [advisory_03000] for the definition of the initValue. [advisory_03000] corresponds to the [constr_1225] that is defined in the Software Component Template [4].

[advisory_03000] initValue defined in the context of ISignal that references a SystemSignal with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE

Imposition time: CP: IT_SysDesc

⌈If an ISignal references a SystemSignal that in turn aggregates physicalProps with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE then the definition of an initValue of the ISignal should be a TextValueSpecification. In this case the value provided should match to the applicable text values (vt, shortLabel,symbol, where the preference rule defined in [TPS_SWCT_01696] is applied) defined by the applicable CompuScales.⌋

[constr_3437] invalidValue defined in the context of ISignal

Imposition time: CP: IT_SysDesc

⌈The definition of SwDataDefProps.invalidValue aggregated by an ISignal in the role networkRepresentationProps shall only be a NumericalValueSpecification, TextValueSpecification or ArrayValueSpecification that aggregates elements of type NumericalValueSpecification or TextValueSpecification.⌋

Please note that for ISignals that are referencing a SystemSignal with a CompuMethod of category category TEXTTABLE or BITFIELD_TEXTTABLE the [constr_3067] is further restricted by [advisory_03001] for the definition of the invalidValue. [advisory_03001] corresponds to the [constr_10196] that is defined in the Software Component Template [4].

[advisory_03001] invalidValue defined in the context of ISignal that references a SystemSignal with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE

Imposition time: CP: IT_SysDesc

⌈If an ISignal references a SystemSignal that in turn aggregates physicalProps with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE then the definition of an invalidValue of the ISignal should be a TextValueSpecification if the value fits into the intervals defined by the CompuMethod. In this case the value provided should match to one of the applicable text values (vt, shortLabel,symbol, where the preference rule defined in [TPS_SWCT_01696] is applied) defined by the applicable CompuScales.⌋

[constr_3438] timeoutSubstitutionValue defined in the context of ISignal

Imposition time: CP: IT_SysDesc

⌈The definition of an timeoutSubstitutionValue in the context of an ISignal shall only be a NumericalValueSpecification, TextValueSpecification or ArrayValueSpecification that aggregates elements of type NumericalValueSpecification or TextValueSpecification.⌋

Please note that for ISignals that are referencing a SystemSignal with a CompuMethod of category category TEXTTABLE or BITFIELD_TEXTTABLE the [constr_3438] is further restricted by [advisory_03002] for the definition of the timeoutSubstitutionValue. [advisory_03002] corresponds to the [constr_1225] that is defined in the Software Component Template [4].

[advisory_03002] timeoutSubstitutionValue defined in the context of ISignal that references a SystemSignal with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE

Imposition time: CP: IT_SysDesc

⌈If an ISignal references a SystemSignal that in turn aggregates physicalProps with a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE then the definition of an timeoutSubstitutionValue of the ISignal should be a TextValueSpecification. In this case the value provided should match to one of the applicable text values (vt, shortLabel,symbol, where the preference rule defined in [TPS_SWCT_01696] is applied) defined by the applicable CompuScales.⌋

[TPS_SYST_02012] initValue and invalidValue represent internal values ⌈The initValue and invalidValue aggregated by the networkRepresentationProps shall represent the internal values.⌋

[TPS_SYST_02110] Default behavior for ISignal.iSignalType ⌈In case ISignal.iSignalType is not defined the value "primitive" shall be assumed.⌋

[TPS_SYST_02144] ComTimeoutSubstitution does not apply for signal gateway operation ⌈The specification of ComTimeoutSubstitution by defining the

ISignal.timeoutSubstitutionValue does not apply for signal gateway operation. Only when the ISignal is processed for an upper layer the ComTimeoutSubstitution is actually performed.⌋

Note: Since an ISignal may be candidate for both - local reception and gateway operation - a definition of ISignal.timeoutSubstitutionValue is valid on ISignals which are defined for gateway operation.

Class	**ISignal**			
Note	Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup. **Tags:** atp.recommendedPackage=ISignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataTransformation, data Transformation.dataTransformation, data Transformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	0..1	attr	With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.

▽

△

Class	ISignal			
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "Init Value". If a full DataMapping exist for the SystemSignal this information may be available from a configured Sender ComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification. **Tags:** atp.Splitkey=initValue
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Tags:** atp.Splitkey=iSignalProps
iSignalType	ISignalTypeEnum	0..1	attr	This attribute defines whether this iSignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type.
length	UnlimitedInteger	0..1	attr	Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalid Value" and the Data Semantics. **Tags:** atp.Splitkey=networkRepresentationProps
systemSignal	SystemSignal	0..1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal. **Tags:** atp.Splitkey=systemSignal
timeout Substitution Value	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstituition for this ISignal. **Tags:** atp.Splitkey=timeoutSubstitutionValue

▽

△

Class	ISignal			
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table 6.7: ISignal

[constr_9222] Existence of `ISignal.dataTypePolicy`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the attribute `dataTypePolicy` shall exist.⌋

[constr_9223] Existence of `ISignal.length`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the attribute `length` shall exist.⌋

[constr_9224] Existence of `ISignal.systemSignal`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignal`, the reference to `SystemSignal` in the role `systemSignal` shall exist.⌋

Enumeration	DataTypePolicyEnum
Note	This class lists the supported DataTypePolicies.
Aggregated by	ISignal.dataTypePolicy
Literal	*Description*
ddsService	This literal indicates that this ISignal is used to transport a message as part of a service for Dds. **Tags:** atp.EnumerationLiteralIndex=6 atp.Status=candidate
ddsSignal	This literal indicates that this ISignal is used to transport a signal based signal for Dds. **Tags:** atp.EnumerationLiteralIndex=5 atp.Status=candidate
legacy	In case the System Description doesn't use a complete Software Component Description (VFB View) this value can be chosen. This supports the inclusion of legacy signals. The aggregation of SwDataDefProps shall be used to configure the "ComSignalDataInvalidValue" and the Data Semantics. **Tags:** atp.EnumerationLiteralIndex=0
network Representation FromComSpec	Ignore any networkRepresentationProps of this ISignal and use the networkRepresentation from the ComSpec. Please note that the usage does not imply the existence of the SwDataDefProps in the role network Representation aggregated by the SenderComSpec or ReceiverComSpec if an ImplementationData Type is defined. **Tags:** atp.EnumerationLiteralIndex=1

▽

△

Enumeration	**DataTypePolicyEnum**
override	If this value is chosen the requirements specified in the ComSpec (networkRepresentationFromCom Spec) are not fullfilled by the aggregated SwDataDefProps. In this case the networkRepresentation is specified by the aggregated swDataDefProps. **Tags:** atp.EnumerationLiteralIndex=2
transformingISignal	This literal indicates that a transformer chain shall be used to communicate the ISignal as UINT8_N over the bus. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.8: DataTypePolicyEnum

Enumeration	**ISignalTypeEnum**
Note	This enumeration defines ISignal types that are used for derivation of the ComSignalType in the COM configuration.
Aggregated by	ISignal.iSignalType
Literal	***Description***
array	ISignal shall be interpreted as an array (UINT8_N, UINT8_DYN) **Tags:** atp.EnumerationLiteralIndex=0
primitive	ISignal shall be interpreted as a primitive type (e.g. UINT_8, SINT_32) **Tags:** atp.EnumerationLiteralIndex=1

Table 6.9: ISignalTypeEnum

Class	**ISignalProps**			
Note	Additional ISignal properties that may be stored in different files.			
Base	*ARObject*			
Aggregated by	ISignal.iSignalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
handleOutOf Range	HandleOutOfRange Enum	0..1	attr	This attribute defines the outOfRangeHandling for received and sent signals.

Table 6.10: ISignalProps

Enumeration	**HandleOutOfRangeEnum**
Note	A value of this type is taken for controlling the range checking behavior of the AUTOSAR RTE.
Aggregated by	ISignalProps.handleOutOfRange, *ReceiverComSpec*.handleOutOfRange, *SenderComSpec*.handle OutOfRange
Literal	***Description***
default	The RTE will use the initValue if the actual value is out of the specified bounds. **Tags:** atp.EnumerationLiteralIndex=0
external Replacement	This indicates that the value replacement is sourced from the attribute replaceWith. **Tags:** atp.EnumerationLiteralIndex=1
ignore	The RTE will ignore any attempt to send or receive the corresponding dataElement if the value is out of the specified range. **Tags:** atp.EnumerationLiteralIndex=2
invalid	The RTE will use the invalidValue if the value is out of the specified bounds. **Tags:** atp.EnumerationLiteralIndex=3

▽

△

Enumeration	**HandleOutOfRangeEnum**
none	A range check is not required. **Tags:** atp.EnumerationLiteralIndex=4
saturate	The RTE will saturate the value of the dataElement such that it is limited to the applicable upper bound if it is greater than the upper bound. Consequently, it is limited to the applicable lower bound if the value is less than the lower bound. **Tags:** atp.EnumerationLiteralIndex=5

Table 6.11: HandleOutOfRangeEnum

Class	**ISignalGroup**			
Note	SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage) **Tags:** atp.recommendedPackage=ISignalGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comBased SignalGroup Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation, comBasedSignalGroupTransformation.dataTransformation, comBasedSignalGroupTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together. **Tags:** atp.Splitkey=iSignal
systemSignal Group	SystemSignalGroup	0..1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup. **Tags:** atp.Splitkey=systemSignalGroup
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignal Groups are described in the TransformationTechnology class. **Tags:** atp.Splitkey=transformationISignalProps

Table 6.12: ISignalGroup

[constr_9225] Existence of ISignalGroup.systemSignalGroup

Imposition time: CP: IT_SysDesc

⌈For each ISignalGroup, the reference to SystemSignalGroup in the role systemSignalGroup shall exist.⌋

Class	SystemSignalGroup			
Note	A signal group refers to a set of signals that shall always be kept together. A signal group is used to guarantee the atomic transfer of AUTOSAR composite data types. The SystemSignalGroup defines a signal grouping on VFB level. On cluster level the Signal grouping is described by the ISignalGroup element. **Tags:** atp.recommendedPackage=SystemSignalGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that shall always be kept together. **Tags:** atp.Splitkey=systemSignal
transforming SystemSignal	SystemSignal	0..1	ref	Optional reference to the SystemSignal which shall contain the transformed (linear) data. **Tags:** atp.Splitkey=transformingSystemSignal

Table 6.13: SystemSignalGroup

Class	ISignalToIPduMapping			
Note	An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ISignalIPdu.iSignalToPduMapping, NmPdu.iSignalToIPduMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iSignal	ISignal	0..1	ref	Reference to a ISignal that is mapped into the ISignal IPdu. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive. **Tags:** atp.Splitkey=iSignal
iSignalGroup	ISignalGroup	0..1	ref	Reference to an ISignalGroup that is mapped into the SignalIPdu. If an ISignalToIPduMapping for an ISignal Group is defined, only the UpdateIndicationBitPosition and the transferProperty is relevant. The startPosition and the packingByteOrder shall be ignored. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive. **Tags:** atp.Splitkey=iSignalGroup

▽

△

Class	**ISignalToIPduMapping**			
packingByte Order	ByteOrderEnum	0..1	attr	This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.
startPosition	UnlimitedInteger	0..1	attr	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.
transferProperty	TransferPropertyEnum	0..1	attr	Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.
update IndicationBit Position	UnlimitedInteger	0..1	attr	The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table 6.14: ISignalToIPduMapping

[constr_5322] Value range of `ISignalToIPduMapping.startPosition`

Imposition time: CP: `IT_SysDesc`

⌈The value of `ISignalToIPduMapping.startPosition` shall be in the range of 0..4294967295 Bits.⌋

Please note that the range of ISignalToIPduMapping.startPosition is resctricted by [constr_5322] to the max value of 4294967295 Bits because of the defined range of the ComBitPosition parameter that is defined in the COM Configuration [25].

[constr_5323] Value range of ISignalToIPduMapping.updateIndicationBitPosition

Imposition time: CP: IT_SysDesc

⌈The value of ISignalToIPduMapping.updateIndicationBitPosition shall be in the range of 0..4294967295 Bits.⌋

Please note that the range of ISignalToIPduMapping.updateIndicationBitPosition is resctricted by [constr_5323] to the max value of 4294967295 Bits because of the defined range of the ComUpdateBitPosition parameter that is defined in the COM Configuration [25].

[constr_3514] No two ISignalToIPduMappings shall reference the identical ISignal

Imposition time: CP: IT_SysDesc

⌈No two ISignalToIPduMappings shall reference the identical ISignal in the role iSignal in the scope of one System.⌋

Enumeration	**TransferPropertyEnum**
Note	Transfer Properties of a Signal.
Aggregated by	ISignalToIPduMapping.transferProperty
Literal	***Description***
pending	If the signal has the TransferProperty pending, then the function Com_SendSignal shall not perform a transmission of the IPdu associated with the signal. **Tags:** atp.EnumerationLiteralIndex=0
triggered	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. **Tags:** atp.EnumerationLiteralIndex=1
triggeredOnChange	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. **Tags:** atp.EnumerationLiteralIndex=2
triggeredOnChange WithoutRepetition	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. **Tags:** atp.EnumerationLiteralIndex=3
triggeredWithout Repetition	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.15: TransferPropertyEnum

[TPS_SYST_02198] Applicable transferProperty for ISignal ⌈If the ISignalToIPduMapping refers to an ISignal in the role iSignal then

- the pending transferProperty does not cause transmission of the ISignalIPdu if the ISignal is updated.
- if the ISignalIPdu has an EventControlledTiming aggregated at the TransmissionModeTiming then the transferProperty values
  - triggered and triggeredWithoutRepetition do cause immediate transmission of the ISignalIPdu if the ISignal is updated.
  - triggeredOnChange and triggeredOnChangeWithoutRepetition do cause immediate transmission of the ISignalIPdu if the ISignal is updated and has changed.

⌋

[constr_3460] Full definition of transferProperty for group signal

Imposition time: CP: IT_SysDesc

⌈If at least one of the ISignals belonging to an ISignalGroup has a transferProperty defined (via their respective ISignalToIPduMapping) then
all other ISignals belonging to the same ISignalGroup shall have a transferProperty defined as well.⌋

Note: [constr_3460] ensures that either

- no group signal has a transferProperty defined, then [TPS_SYST_02199] applies, or
- every group signal has a transferProperty defined, then [TPS_SYST_02199] (in case that all group signals have the transferProperty set to pending) or [TPS_SYST_02200] (in all other cases) applies.

[constr_5344] Applicable transferProperty for GroupSignal and ISignalGroup

Imposition time: CP: IT_SysDesc

⌈

transferProperty on ISignalGroup	transferProperty on GroupSignals	Semantic
• not set	Not set or pending for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the ISignalIPdu.
	Subset of Group Signals has transferProperty set to either triggered or triggeredWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group marks the ISignalIPdu for transmission.

▽

△

	Subset of Group Signals has transferProperty set to either triggeredOnChange or triggered-OnChangeWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group and change of a Group Signal that has transferProperty set to triggeredOnChange or triggered-OnChangeWithoutRepetition causes immediate transmission of the ISignalIPdu.
• pending	Not set or pending for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the ISignalIPdu. See [TPS_SYST_02199].
• triggered • triggeredWithoutRepetition	Not set or pending for all Group Signals	Update of the Signal Group marks the ISignalIPdu for transmission. See [TPS_SYST_02199]
	Subset of Group Signals has transferProperty set to either triggered or triggeredWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group marks the ISignalIPdu for transmission. See [TPS_SYST_02200].
• triggeredOnChange • triggeredOnChangeWithout-Repetition	Not set or pending for all Group Signals	Update of Signal Group causes immediate transmission of the ISignalIPdu. See [TPS_SYST_02199]
	Subset of Group Signals has transferProperty set to either triggeredOnChange or triggered-OnChangeWithoutRepetition and the other Group Signals have transferProperty either not set or set to pending.	Update of Signal Group and change of a Group Signal that has transferProperty set to triggeredOnChange or triggered-OnChangeWithoutRepetition causes immediate transmission of the ISignalIPdu. See [TPS_SYST_02200].

If the ISignalToIPduMapping refers to an ISignalGroup in the role iSignalGroup and the ISignalIPdu has an EventControlledTiming aggregated at the Transmission-ModeTiming then combinations of transferProperty attribute settings for the ISignal-Group and the included ISignals are supported as defined in this table.

⌋

[TPS_SYST_02199] Applicable transferProperty for ISignalGroup and all group signals have transferProperty not defined or pending defined ⌈If the ISignalToIPduMapping refers to an ISignalGroup in the role iSignalGroup and
the ISignalIPdu has an EventControlledTiming aggregated at the TransmissionModeTiming and
all ISignals belonging to the ISignalGroup have a transferProperty not defined or pending defined (via their respective ISignalToIPduMapping) then
if the ISignalToIPduMapping of the ISignalGroup has the transferProperty

- pending defined then an update of this ISignalGroup does not cause the transmission of the ISignalIPdu.
- triggered or triggeredWithoutRepetition defined then an update of this ISignalGroup marks the ISignalIPdu for transmission

- triggeredOnChange or triggeredOnChangeWithoutRepetition defined then an update of this ISignalGroup in combination with the change of any of the contained group signals does cause immediate transmission of the ISignalIPdu.

⌋

[TPS_SYST_02200] Applicable transferProperty for ISignalGroup and group signals have transferProperty defined ⌈If the ISignalToIPduMapping refers to an ISignalGroup in the role iSignalGroup and
the ISignalIPdu has an EventControlledTiming aggregated at the TransmissionModeTiming and
at least one of the ISignals belonging to the ISignalGroup has a transferProperty defined (via their respective ISignalToIPduMapping) then
if the ISignalToIPduMapping of the ISignalGroup has the transferProperty

- pending defined then an update of this ISignalGroup does not cause the transmission of the ISignalIPdu.
- triggered or triggeredWithoutRepetition defined then an update of this ISignalGroup marks the ISignalIPdu for transmission
- triggeredOnChange or triggeredOnChangeWithoutRepetition defined then an update of this ISignalGroup in combination with the change of any of the contained group signals which have transferProperty=triggeredOnChange defined does cause immediate transmission of the ISignalIPdu.

⌋

[constr_3461] TransferProperty for group signals if ISignalGroup has transferProperty=pending

Imposition time: CP: IT_SysDesc

⌈If the ISignalToIPduMapping refers to an ISignalGroup in the role iSignalGroup and
the transferProperty is set to pending then
the group signals of this ISignalGroup shall either

- have no transferProperty defined (via their respective ISignalToIPduMapping) or
- every ISignal belonging to the ISignalGroup shall have the transferProperty=pending defined.

⌋

Class	**ISignalTriggering**			
Note	A ISignalTriggering allows an assignment of ISignals to physical channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.iSignalTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iSignal	ISignal	0..1	ref	This reference shall be used if an ISignal is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignalTriggering-ISignalGroup reference. **Tags:** atp.Splitkey=iSignal
iSignalGroup	ISignalGroup	0..1	ref	This reference shall be used if an ISignalGroup is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignal Triggering-ISignal reference. **Tags:** atp.Splitkey=iSignalGroup
iSignalPort	ISignalPort	*	ref	References to the ISignalPort on every ECU of the system which sends and/or receives the ISignal. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=iSignalPort

Table 6.16: ISignalTriggering

[constr_9326] Exclusive existence of `ISignalTriggering.iSignal` and `ISignalTriggering.iSignalGroup`

Imposition time: CP: `IT_SysDesc`

⌈Each `ISignalTriggering` shall either define an `ISignalTriggering.iSignal` or an `ISignalTriggering.iSignalGroup` reference.⌋

6.2.1 Large Data Path in COM

AUTOSAR defines an alternative communication path between the RTE and the Communication Stack called Large Data Path in COM. The System Template does not define specific attributes which would distinguish whether the traditional Com or the Large Data Path in COM shall be used. The idea behind this feature is rather that

- IF the Large Data Path in COM module is enabled in an Ecu
- AND the specific interaction fulfills certain properties
- THEN Large Data Path in COM shall be used.

Thus the usage of Large Data Path in COM inside an ECU is project specific and is not derived from system description properties.

All of the following requirements need to be fulfilled in order to allow the usage of Large Data Path in COM for the specific `ISignal` / `ISignalIPdu` combination.

[TPS_SYST_02015] Large Data Path in COM: only one `ISignal` mapped to the `ISignalIPdu` ⌈Only if exactly one `ISignal` is mapped into an `ISignalIPdu` and

Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02016] Large Data Path in COM: only Transformer output and UINT8_N or UINT8_DYN supported ⌈Only if

- the data type of the ISignal is either UINT8_N or UINT8_DYN
- or the ISignal has a reference to the DataTransformation in the role dataTransformation

and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02017] Large Data Path in COM: Opaque ISignalToIPduMapping.packingByteOrder ⌈Only if packingByteOrder has the value "Opaque" and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02018] Large Data Path in COM: ISignalToIPduMapping.startPosition shall be 0 ⌈Only if ISignalToIPduMapping.startPosition equals 0 (zero) and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02019] Large Data Path in COM: ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition for sent ISignals ⌈Only if ISignalToIPduMapping.transferProperty equals triggered or triggeredWithoutRepetition for a Signal that is sent by the EcuInstance and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02021] Large Data Path in COM: ISignalToIPduMapping.updateIndicationBitPosition shall not be defined ⌈Only if ISignalToIPduMapping.updateIndicationBitPosition is not defined and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02022] Large Data Path in COM: Only the transmissionModeTrueTiming defined ⌈The ISignal is mapped into an ISignalIPdu. Only if this ISignalIPdu has exactly the TransmissionModeDeclaration.transmissionModeTrueTiming defined (via ISignalIPdu.iPduTimingSpecification), the ISignalIPdu is sent by the EcuInstance and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02023] Large Data Path in COM: DataFilter "always" if TransmissionModeCondition defined ⌈The ISignal is mapped into an ISignalIPdu. If this ISignalIPdu has either

- no TransmissionModeDeclaration.transmissionModeCondition defined (via ISignalIPdu.iPduTimingSpecification) or

- DataFilter.dataFilterType is set to "always" for the TransmissionModeCondition of this ISignalIPdu.

and this ISignalIPdu is sent by the EcuInstance and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02024] Large Data Path in COM: No ModeDrivenTransmissionModeCondition defined ⌈The ISignal is mapped into an ISignalIPdu. Only if this ISignalIPdu has no TransmissionModeDeclaration.modeDrivenTrueCondition and modeDrivenFalseCondition, the ISignalIPdu is sent by the EcuInstance and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02025] Large Data Path in COM: Only EventControlledTiming defined ⌈The ISignal is mapped into an ISignalIPdu. Only if this ISignalIPdu has an EventControlledTiming (via TransmissionModeTiming.eventControlledTiming) and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02026] Large Data Path in COM: Only EventControlledTiming with no repetition defined ⌈The ISignal is mapped into an ISignalIPdu. Only if this ISignalIPdu has an EventControlledTiming (via TransmissionModeTiming.eventControlledTiming) with EventControlledTiming.numberOfRepetitions = 0 defined and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

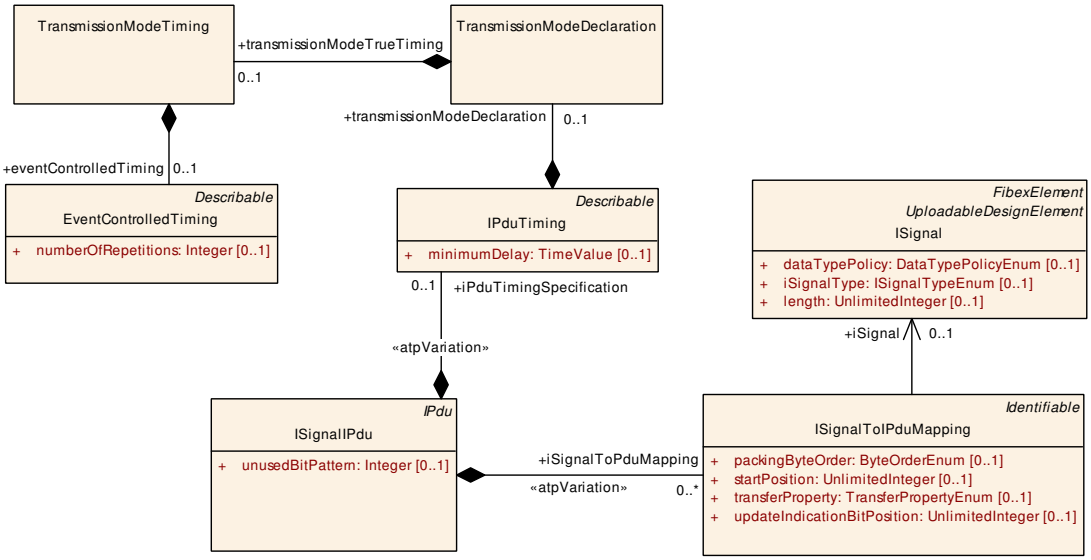

Figure 6.10: Pdu Timing excerpt that may be used to configure Large Data Path in COM

[TPS_SYST_02028] Large Data Path in COM: No ISignalPort.dataFilter defined ⌈Only if the ISignalPort which the ISignalTriggering is referring to has either

- no ISignalPort.dataFilter defined

- or the DataFilter.dataFilterType = always

and Large Data Path in COM is enabled, this ISignal shall be handled by Large Data Path in COM.⌋

[TPS_SYST_02390] Relevance of Large Data Path in COM spec items for the sender and for the receiver side. ⌈

Spec Item Number	*Spec Item Headline*	*Motivation*	*Applies to*	*Model element*
[TPS_SYST_02015]	Only one ISignal mapped to the ISignalIPdu.	Large Data Path in COM only supports one ISignal per PDU.	Rx + Tx	ISignalToIPduMapping
[TPS_SYST_02016]	Only Transformer output and UINT8_N or UINT8_DYN supported.	Large Data Path in COM only supports Byte Array Signals.	Rx + Tx	ISignal
[TPS_SYST_02017]	Opaque ISignalToIPduMapping.packingByteOrder.	Large Data Path in COM only supports Byte Array Signals.	Rx + Tx	ISignalToIPduMapping
[TPS_SYST_02018]	ISignalToIPduMapping.startPosition shall be 0.	Large Data Path in COM only supports Signals starting in the first byte of the PDU.	Rx + Tx	ISignalToIPduMapping
[TPS_SYST_02019]	ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition.	Large Data Path in COM only supports one ISignal per PDU. That signal has to trigger the transmission.	Tx	ISignalToIPduMapping
[TPS_SYST_02021]	ISignalToIPduMapping.updateIndicationBitPosition shall not be defined.	Large Data Path in COM does not support update bits.	Rx + Tx	ISignalToIPduMapping
[TPS_SYST_02022]	Only the transmissionModeTrueTiming defined.	Large Data Path in COM does not support transmission modes.	Tx	ISignalIPdu
[TPS_SYST_02023]	DataFilter "always" if TransmissionModeCondition defined.	Large Data Path in COM does not support transmission modes.	Tx	ISignalIPdu
[TPS_SYST_02024]	No ModeDrivenTransmissionModeCondition defined.	Large Data Path in COM does not support transmission modes.	Tx	ISignalIPdu
[TPS_SYST_02025]	Only EventControlledTiming defined.	Large Data Path in COM does not support cyclic transmission.	Tx	ISignalIPdu
[TPS_SYST_02026]	Only EventControlledTiming with no repetition defined.	Large Data Path in COM does not support cyclic transmission nor repeated transmission.	Tx	ISignalIPdu
[TPS_SYST_02028]	No ISignalPort.dataFilter defined.	Large Data Path in COM does not support timeout handling.	Rx	ISignalPort

⌋

[TPS_SYST_03150] Mixed Large Data Path in COM and Traditional COM for Client Server communication ⌈For one Client Server communication it is possible to define one communication direction (call / return) where ComIPdus in COM have ComIPduLargeData set to TRUE (Large Data Path in COM) and the other communication direction (return / call) where ComIPdus in COM have ComIPduLargeData set to FALSE (Traditional COM).⌋

6.2.2 Big Endian and Little Endian memory layout of Pdus and Frames

The AUTOSAR system description provide means to specify how the memory layout looks like when signals are packed into Pdus and Pdus are packed into Frames. The layout of Pdus and Frames on different communication systems is out of scope of AUTOSAR. The specification of attributes Bit counting (monotone or sawtooth) and Bit order (decreasing or increasing)[1] is not supported by AUTOSAR. In AUTOSAR these attributes are fixed.

[1] More details about Bit counting and Bit order can be found in ASAM FIBEX [8].

[TPS_SYST_01068] Bit Counting in AUTOSAR ⌈The Bit counting shall always be considered as "sawtooth".⌋

[TPS_SYST_01069] Bit Order in AUTOSAR ⌈The bit order shall always be considered as "Decreasing".⌋

When a signal is mapped into a `Pdu` only the `packingByteOrder` affects the memory layout of the signal inside the `Pdu` beginning with it's start bit position.

Little endian stores the least significant byte first and begins with the least significant bit, i.e. loworder bit in the sequence (the least significant bit serves as start bit).

Big endian stores the most significant byte first and begins with the most significant bit, i.e. the bit with the greatest numerical value (the most significant bit serves as start bit).

In both cases the bit positions in the mapped signals increase with the bit positions in the `ISignalIPdu` such that the bit 2^0 is mapped to position n in the `ISignalIPdu` and bit 2^1 is mapped to position n+1 and so on.

Example 6.11 shows the memory layout for Little Endian and Big Endian if an `ISignal` with a length of 10 bits is mappped into a `Pdu`. The start bit position is 5.

Little Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	2^2	2^1	2^0	-	-	-	-	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3

Big Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	-	-	-	-

Figure 6.11: PackingByteOrder Example

The following examples are showing the mapping of Pdus into Frames.

The first example in Figure 6.12 for little endian shows a `Frame` with four bytes that contains a single `Pdu` that is two bytes long. The `PduToFrameMapping.startPosition` is defined with 8 and since the `packingByteOrder` is set to `mostSignificantByteLast` the `startPosition` denotes the least significant bit of the `Pdu` in the `Frame`. The bit position of the mapped `Pdu` increases with the bit positions in the `Frame` such that the bit 2^0 is mapped to position n in the `ISignalIPdu` and bit 2^1 is mapped to position n+1 and so on.

Please note that the Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) for `PduToFrameMapping.startPosition` are allowed.

Figure 6.12 also shows that the Pdu contains three ISignals. The first ISignal has the ISignalToIPduMapping.startPosition defined as 0 and is 5 bits long. The bitposition of the second signal is 5 and the length is 10 bits. And the third signal has the bitposition 15 and is only 1 bit long. Since the ISignalToIPduMapping.packingByteOrder is defined with mostSignificantByteLast as well the startPosition of the ISignals denotes the least significant bit of the ISignal in the Pdu.

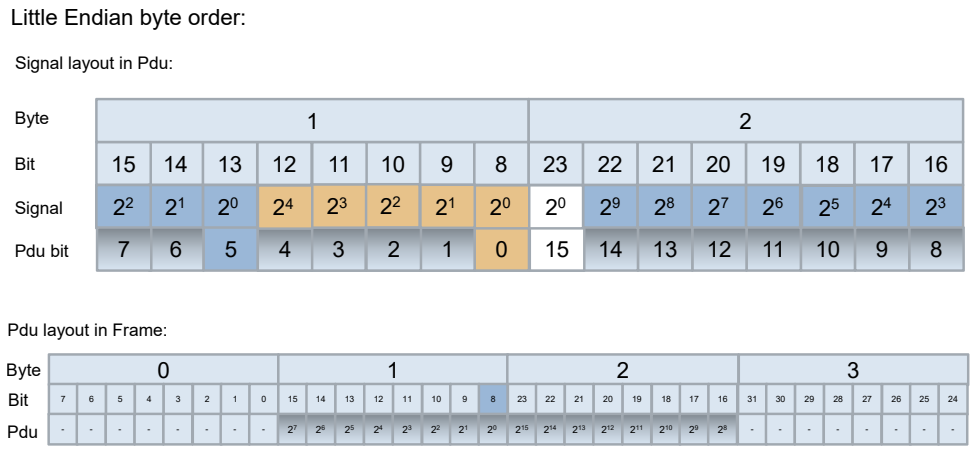

Figure 6.12: PackingByteOrder Example

The second example in Figure 6.13 for big endian shows again a Frame with four bytes that contains a single Pdu that is two bytes long. The PduToFrameMapping.startPosition is defined with 15 and since the packingByteOrder is set to mostSignificantByteFirst the startPosition denotes the most significant bit of the Pdu in the Frame.

Figure 6.13 also shows that the Pdu contains three ISignals. The first ISignal has the ISignalToIPduMapping.startPosition defined as 7 and is 5 bits long. The bitposition of the second signal is 2 and the length is 10 bits. And the third signal has the bitposition 8 and is only 1 bit long. Since the ISignalToIPduMapping.packingByteOrder is defined with mostSignificantByteFirst as well the startPosition of the ISignals denotes the most significant bit of the ISignal in the Pdu.

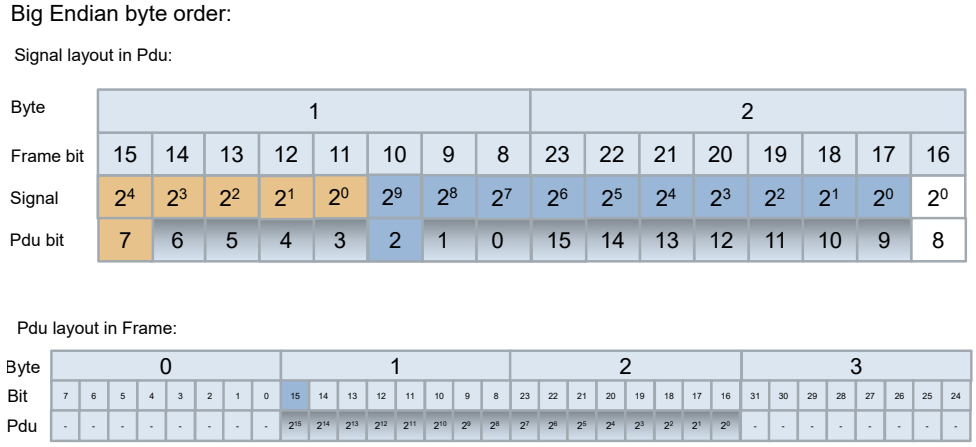

Figure 6.13: PackingByteOrder Example

Please note that the positioning of `SegmentPosition`s in a `MultiplexedIPdu` works in the exact same way. The examples in Figure 6.12 and Figure 6.13 can be taken as well as an example for a `MultiplexedIPdu` where the 1 bit signal defines the selectorField and the other two signals represent segments defined for the `DynamicPart`.

6.3 PDUs

The chapter introduces the different Pdu types that are supported in the AUTOSAR Architecture and by the AUTOSAR Meta-Model.

The PDU Router is responsible only for the routing of IPdus. Other Pdus that are direct specializations of the Pdu meta-class are not routed by the PDU Router.

UserDefinedPdus and UserDefinedIPdus are used to describe PDU-based communication over Complex Drivers. Chapter 6.14 provides a more detailed description of CDDs.

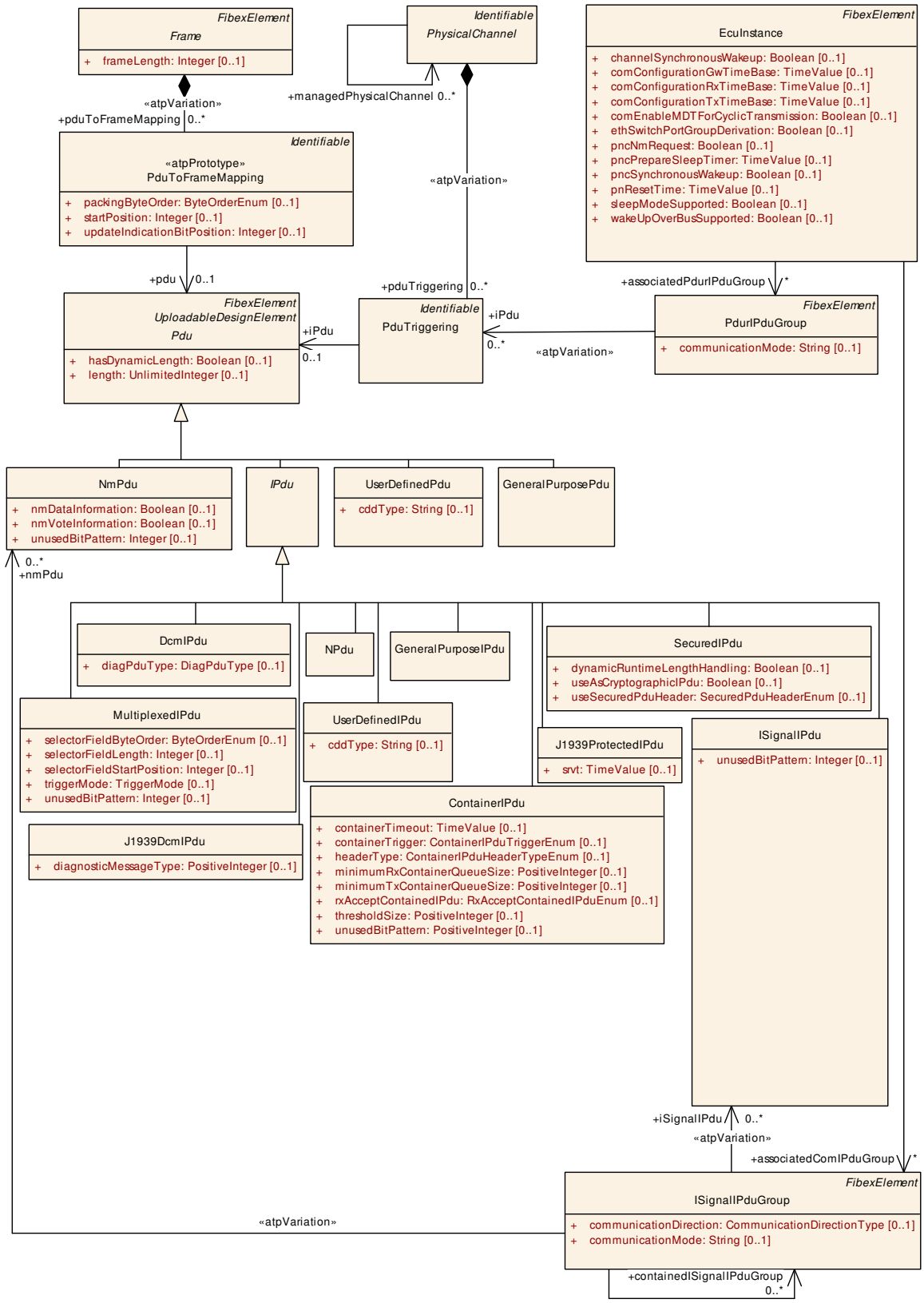

Figure 6.14: Pdus and the mapping into Frames (FibexCore: PDUOverview)

The PduToFrameMapping element describes the mapping of Pdus to Frames and defines the position of a Pdu within a Frame. By using different PduToFrameMappings it is possible to use the same Pdu in different Frames.

[constr_3516] limitation of Frame.frameLength for CAN L-PDUs

Imposition time: CP: IT_SysDesc

⌈The Frame.frameLength of CAN PDUs shall be restricted to

- 0..8 for classic CAN L-PDUs;
- 0..8, 12, 16, 20, 24, 32, 48, 64 for CAN FD L-PDUs and
- 1..2048 for CAN XL L-PDUs.

⌋

Please note that only a single Pdu is allowed to be mapped into a CanFrame. The Pdu.length of the Pdu that is mapped into the CanFrame cannot be smaller than the CanFrame.frameLength, but is allowed to be larger (see [TPS_SYST_02255]). If the CanFrame is a CAN XL frame (SDU Type other than 0x03h, see Ch. 3.3.1.2.2), the Pdu.length cannot be 0.

A timing description IPduTiming can be aggregated directly by the ISignalIPdu. This timing description can be used for the Configuration of COM Transmission Modes. The PduTriggering describes on which channel the Pdu is transmitted. Timing requirements may be specified with the Timing Extension model. More details are described in chapter 1.4.3. Such Pdu timing requirements needs to be fulfilled by the timing specification on the Frame.

Class	***Pdu*** (abstract)			
Note	Collection of all Pdus that can be routed through a bus interface.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	GeneralPurposePdu, *IPdu*, NmPdu, UserDefinedPdu			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hasDynamic Length	Boolean	0..1	attr	This attribute defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].
length	UnlimitedInteger	0..1	attr	Pdu length in bytes. In case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits. The Pdu length of zero bytes is allowed.

Table 6.17: Pdu

[constr_5249] Existence of Pdu.length

Imposition time: CP: IT_SysDesc

⌈For each Pdu, the attribute length shall exist.⌋

Reason: Bsw modules might buffer the payload of the Pdu and for this purpose the Pdu.length needs to be provided. A second reason is that future extensions of the Pdu can be prepared by setting the length to an appropriate value.

[constr_5321] Value range of Pdu.length

Imposition time: CP: IT_SysDesc

⌈The value of Pdu.length shall be in the range of 0..4294967295 Bytes.⌋

[constr_3448] Restriction for usage of Pdu.hasDynamicLength

Imposition time: CP: IT_SysDesc

⌈The Pdu.hasDynamicLength attribute is only relevant for UserDefinedPdus, UserDefinedIPdus, J1939DcmIPdus.⌋

[TPS_SYST_03085] Pdu qualifies as dynamic length ⌈If a Pdu meets one of the following criteria this Pdu is considered dynamic length:

- ISignalIPdu: At least one dynamic signal mapped to the ISignalIPdu.
- SecuredIPdu and MultiplexedIPdu: At least one of the associated upper layer Pdus has dynamic length.
- ContainerIPdu: Is dynamic if:
  - the headerType is shortHeader or longHeader or
  - the headerType is noHeader and the last contained Pdu has dynamic length.
- NPdu: TP layer takes care of length handling, not visible to application.
- DcmIPdu: always dynamic length.
- NmPdu: always static length.
- GeneralPurposePdu and GeneralPurposeIPdu: Depending on upper layer, which could be: SD, TSync, DoIP, XCP, SomeIpTp, Dlt, IDS, VDP.
- Pdu with hasDynamicLength = true

⌋

Class	**IPdu** (abstract)
Note	The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*
Subclasses	ContainerIPdu, DcmIPdu, GeneralPurposeIPdu, ISignalIPdu, J1939DcmIPdu, J1939ProtectedIPdu, MultiplexedIPdu, NPdu, SecuredIPdu, UserDefinedIPdu
Aggregated by	ARPackage.element

▽

△

Class	*IPdu* (abstract)			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.18: IPdu

Class	**ISignalIPdu**			
Note	Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduTiming Specification	IPduTiming	0..1	aggr	Timing specification for Com IPdus (Transmission Modes). This information is mandatory for the sender in a System Extract. This information may be omitted on receivers in a System Extract. atpVariation: The timing of a Pdu can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduTimingSpecification, iPduTiming Specification.variationPoint.shortLabel vh.latestBindingTime=postBuild
iSignalToPdu Mapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the Signal IPdu. atpVariation: The content of a PDU can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalToPduMapping, iSignalToPdu Mapping.shortName, iSignalToPduMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.

Table 6.19: ISignalIPdu

Class	**NmPdu**			
Note	Network Management Pdu **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	NmPdu			
iSignalToIPdu Mapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUser Data that is transmitted in the NmPdu. The counting of the startPosition starts at the beginning of the NmPdu regardless whether Cbv or Nid are used. **Tags:** atp.Splitkey=iSignalToIPduMapping, iSignalToIPdu Mapping.shortName
nmData Information	Boolean	0..1	attr	Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.
nmVote Information	Boolean	0..1	attr	Defines if the Pdu contains NM Vote information.
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM is filling not used areas of an Pdu with this bit-pattern. This attribute can only be used if the nm DataInformation attribute is set to true.

Table 6.20: NmPdu

[constr_5385] Reception of UserData inside of a NmPdu by Applications is not supported

Imposition time: CP: IT_SysDesc

⌈A SystemSignal that is referenced by an ISignal that in turn is mapped via an ISignalToIPduMapping into a NmPdu shall not be mapped by a DataMapping that references a RPortPrototype with the contextPort reference in the VariableDataPrototypeInSystemInstanceRef that the DataMapping aggregates.⌋

The signals in NmPdus are used for debugging purposes and not for exchanging data between Ecus.

Please note that in AUTOSAR only FrNm is able to send out NmPdus with and without voting information:

[constr_3073] nmVoteInformation only valid for FrNm

Imposition time: CP: IT_SysDesc

⌈The nmVoteInformation attribute is only valid for FrNm.⌋

Class	NPdu			
Note	This is a Pdu of the Transport Layer. The main purpose of the TP Layer is to segment and reassemble IPdus. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.21: NPdu

Class	**DcmIPdu**			
Note	Represents the IPdus handled by Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagPduType	DiagPduType	0..1	attr	Attribute is used to distinguish a request from a response.

Table 6.22: DcmIPdu

[constr_9194] Existence of DcmIPdu.diagPduType

Imposition time: CP: IT_SysDesc

⌈For each DcmIPdu, the attribute diagPduType shall exist.⌋

Enumeration	**DiagPduType**
Note	Used to distinguish a diagnostic request from a response.
Aggregated by	DcmIPdu.diagPduType
Literal	***Description***
diagRequest	Diagnostic Request **Tags:** atp.EnumerationLiteralIndex=0
diagResponse	Diagnostic Response **Tags:** atp.EnumerationLiteralIndex=1

Table 6.23: DiagPduType

Class	**J1939DcmIPdu**			
Note	Represents the IPdus handled by J1939Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
diagnostic MessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table 6.24: J1939DcmIPdu

[constr_3096] Allowed values for diagnosticMessageType

Imposition time: CP: IT_SysDesc

⌈The allowed values of diagnosticMessageType range from 1..60.⌋

Class	**GeneralPurposePdu**
Note	This element is used for AUTOSAR Pdus without additional attributes that are routed by a bus interface. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **Tags:** atp.recommendedPackage=Pdus

▽

△

Class	GeneralPurposePdu			
Base	ARElement, ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable, UploadableDesignElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.25: GeneralPurposePdu

[constr_3081] Value of category in GeneralPurposePdu

Imposition time: CP: IT_SysDesc

⌈The attribute category of GeneralPurposePdu can have the following values:

- SD (Service Discovery)
- GLOBAL_TIME
- DoIP

⌋

Class	GeneralPurposeIPdu			
Note	This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **Tags:** atp.recommendedPackage=Pdus			
Base	ARElement, ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable, UploadableDesignElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.26: GeneralPurposeIPdu

[constr_3082] Value of category in GeneralPurposeIPdu

Imposition time: CP: IT_SysDesc

⌈The attribute category of GeneralPurposeIPdu can have the following values:

- XCP
- SOMEIP_SEGMENTED_IPDU
- DLT
- IDS
- VDP

⌋

Class	UserDefinedPdu			
Note	UserDefinedPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the BusIf (e.g. a new Nm module) then this Pdu element shall be used to describe the communication. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedIPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table 6.27: UserDefinedPdu

Class	UserDefinedIPdu			
Note	UserDefinedIPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the PduR (e.g. a Diagnostic Service) then this IPdu element shall be used to describe the communication. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table 6.28: UserDefinedIPdu

Class	«atpPrototype» **PduToFrameMapping**			
Note	A PduToFrameMapping defines the composition of Pdus in each frame.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Frame*.pduToFrameMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
packingByte Order	ByteOrderEnum	0..1	attr	This attribute defines the order of the bytes of the Pdu and the packing into the Frame. Please consider that [constr_3246] and [constr_3222] are restricting the usage of this attribute.
pdu	Pdu	0..1	ref	Reference to a I-Pdu, N-Pdu or NmPdu that is transmitted in the Frame. **Tags:** atp.Splitkey=pdu



△

Class	«atpPrototype» **PduToFrameMapping**			
startPosition	Integer	0..1	attr	This attribute describes the bitposition of a Pdu within a Frame. Please note that the absolute position of the Pdu in the Frame is determined by the definition of the packingByte Order attribute. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the Frame. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the Frame. The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. The Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed.
update IndicationBit Position	Integer	0..1	attr	Indication to the receivers that the corresponding Pdu was updated by the sender. This attribute describes the position of the update bit in the frame that aggregates this PDUToFrameMapping. Length is always one bit. Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing Frame still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table 6.29: PduToFrameMapping

[constr_9195] Existence of PduToFrameMapping.packingByteOrder

Imposition time: CP: IT_SysDesc

⌈For each PduToFrameMapping, the attribute packingByteOrder shall exist.⌋

[constr_9196] Existence of PduToFrameMapping.startPosition

Imposition time: CP: IT_SysDesc

⌈For each PduToFrameMapping, the attribute startPosition shall exist.⌋

[constr_9197] Existence of PduToFrameMapping.pdu

Imposition time: CP: IT_SysDesc

⌈For each PduToFrameMapping, the reference to Pdu in the role pdu shall exist.⌋

[constr_3246] Frame.packingByteOrder mix within a Frame is not allowed

Imposition time: CP: IT_SysDesc

⌈All PduToFrameMappings within a Frame shall have the same packingByte-Order value.⌋

Please note that the absolute position (bit-position) of the Pdu in the Frame is determined by the definition of the packingByteOrder attribute. The Pdus are byte aligned in a Frame and the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed. For reasons of simplicity a mix of the packingByteOrder is not allowed.

[constr_3222] No ByteOrderEnum.opaque allowed for PduToFrameMapping.packingByteOrder

Imposition time: CP: IT_SysDesc

⌈The values of PduToFrameMapping.packingByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

Class	**IPduTiming**			
Note	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES for each IPdu. The Transmission Mode of an IPdu that is valid at a specific point in time is selected using the values of the signals that are mapped to this IPdu. For each IPdu a Transmission Mode Selector is defined. The Transmission Mode Selector is calculated by evaluating the conditions for a subset of signals (class TransmissionModeCondition in the System Template). The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true and is defined to be false, if all Conditions evaluate to false.			
Base	*ARObject*, *Describable*			
Aggregated by	ISignalIPdu.iPduTimingSpecification			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
minimumDelay	TimeValue	0..1	attr	Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.
transmission Mode Declaration	TransmissionMode Declaration	0..1	aggr	AUTOSAR COM allows configuring statically two different transmission modes for each I-PDU (True and False). The Transmission Mode Selector evaluates the conditions for a subset of signals and decides the transmission mode. It is possible to switch between the transmission modes during runtime. **Tags:** atp.Splitkey=transmissionModeDeclaration	

Table 6.30: IPduTiming

Class	**PduTriggering**			
Note	The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu. Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface. If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.pduTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPdu	Pdu	0..1	ref	Reference to the Pdu for which the PduTriggering is defined. One I-Pdu can be triggered on different channels (PduR fan-out). The Pdu routing by the PduR is only allowed for subclasses of IPdu. Nevertheless is the reference to the Pdu element necessary since the PduTriggering element is also used to specify the sending and receiving connections to Ecu Ports. **Tags:** atp.Splitkey=iPdu
iPduPort	IPduPort	*	ref	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=iPduPort

△

Class	**PduTriggering**			
iSignal Triggering	ISignalTriggering	*	ref	This reference provides the relationship to the ISignal Triggerings that are implemented by the PduTriggering. The reference is optional since no ISignalTriggering can be defined for DCM and Multiplexed Pdus. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalTriggering, iSignalTriggering.iSignal Triggering, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
secOcCrypto Mapping	SecOcCryptoService Mapping	0..1	ref	This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced Secured IPdu. Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role i Pdu. **Tags:** atp.Splitkey=secOcCryptoMapping atp.Status=obsolete
secOcCrypto ServiceMapping	SecOcCryptoService Mapping	0..1	ref	This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced Secured IPdu. Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role i Pdu. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secOcCryptoServiceMapping, secOcCrypto ServiceMapping.secOcCryptoServiceMapping, secOc CryptoServiceMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
triggerIPduSend Condition	TriggerIPduSend Condition	*	aggr	Defines the trigger for the Com_TriggerIPDUSend API call. Only if all defined TriggerIPduSendConditions evaluate to true (AND associated) the Com_Trigger IPDUSend API shall be called. **Tags:** atp.Splitkey=triggerIPduSendCondition

Table 6.31: PduTriggering

[constr_9198] Existence of `PduTriggering.iPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `PduTriggering`, the reference to `Pdu` in the role `iPdu` shall exist.⌋

AUTOSAR COM provides a mechanism of starting/stopping COM PDU groups (`ISignalIPduGroup`). Please note that in a System Model an `ISignalIPdu` can belong to several `ISignalIPduGroup`s of different `EcuInstance`s. So it is not possible to deduce the assignment of an `ISignalIPduGroup` to an `EcuInstance` via the `iSignalIPdu`s that are included in the `ISignalIPduGroup`. The assignment of an `ISignalIPduGroup` to an `EcuInstance` is therefore realized with the `EcuInstance.associatedComIPduGroup` reference.

Please note that the `EcuInstance.associatedComIPduGroup` reference is only used to assign the top-level `ISignalIPduGroup`s to an `EcuInstance`. The assignment of the `containedISignalIPduGroup`s to an `EcuInstance` is done via the aggregating `ISignalIPduGroup` that in turn is referenced by an `EcuInstance`.

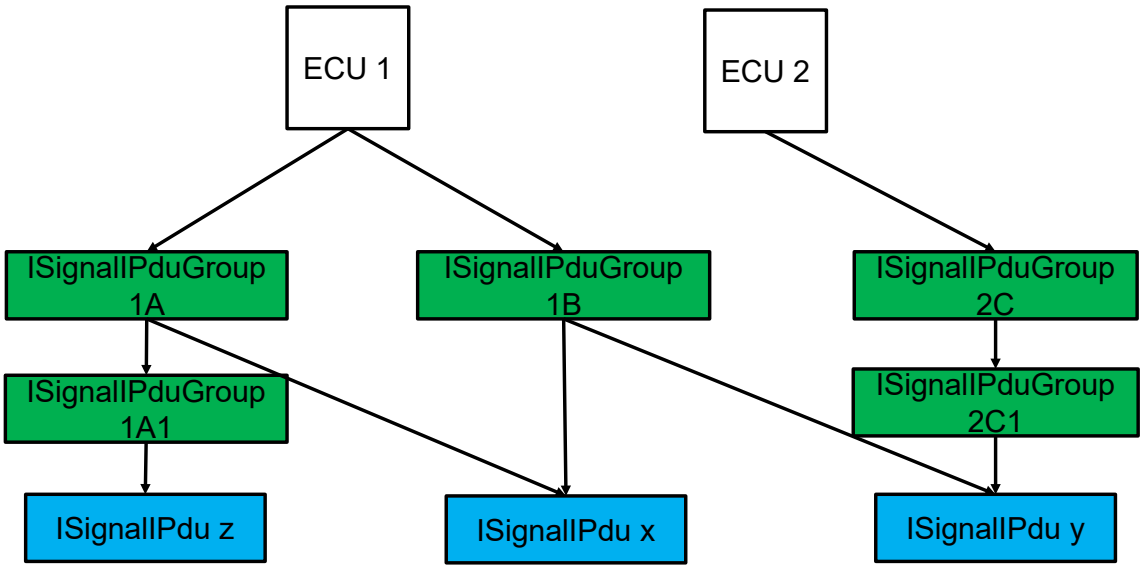

Figure 6.15: Example for ISignalIPduGroups and their assignment to EcuInstances

Class	**ISignalIPduGroup**			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups. **Tags:** atp.recommendedPackage=ISignalIPduGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Direction	Communication DirectionType	0..1	attr	This attribute determines in which direction IPdus that are contained in this IPduGroup will be transmitted (communication direction can be either In or Out).
communication Mode	String	0..1	attr	This attribute defines the use-case for this ISignalIPdu Group (e.g. diagnostic, debugging etc.). For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
contained ISignalIPdu Group	ISignalIPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance. **Tags:** atp.Splitkey=containedISignalIPduGroup
iSignalIPdu	ISignalIPdu	*	ref	Reference to a set of Signal I-Pdus, which are contained in the ISignal I-Pdu Group. atpVariation: The content of a ISignal I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iSignalIPdu, iSignalIPdu.iSignalIPdu, iSignal IPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	ISignalIPduGroup			
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmPdu, nmPdu.nmPdu, nmPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.32: ISignalIPduGroup

[constr_9199] Existence of ISignalIPduGroup.communicationDirection

Imposition time: CP: IT_SysDesc

⌈For each ISignalIPduGroup, the attribute communicationDirection shall exist.⌋

Enumeration	CommunicationDirectionType
Note	Describes the communication direction.
Aggregated by	*CommConnectorPort*.communicationDirection, *IEEE1722TpConnection*.communicationDirection, IPSecRule.direction, ISignalIPduGroup.communicationDirection
Literal	*Description*
in	Reception (Input) **Tags:** atp.EnumerationLiteralIndex=0
out	Transmission (Output) **Tags:** atp.EnumerationLiteralIndex=1

Table 6.33: CommunicationDirectionType

[constr_3020] communicationDirection of containedISignalIPduGroups

Imposition time: CP: IT_SysDesc

⌈The value of the attribute communicationDirection of containedISignal-IPduGroup shall be identical to the value of the attribute communication-Direction of the enclosing ISignalIPduGroup.⌋

The AUTOSAR Pdu Router provides a mechanism of enabling/disabling of routing path groups (PdurIPduGroup).

Please note that in a System Model a PduTriggering can belong to several Pdur-IPduGroups of different EcuInstances. So it is not possible to deduce the assignment of an PdurIPduGroup to an EcuInstance via the PduTriggerings that are included in the ISignalIPduGroup. The assignment of an PdurIPduGroup to an EcuInstance is therefore realized with the EcuInstance.associatedPdurIPdu-Group reference.

Class	**PdurIPduGroup**			
Note	The AUTOSAR PduR will enable and disable the sending of configurable groups of IPdus during runtime according to the AUTOSAR PduR specification. **Tags:** atp.recommendedPackage=PdurIPduGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Mode	String	0..1	attr	This attribute defines the use-case for this PduRIPdu Group. For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
iPdu	PduTriggering	*	ref	Reference to a set of IPdus, which are contained in the PduR I-Pdu Group. If an IPdu is routed by the PduR to different destinations (PduR fan-out) than an Pdu Triggering for each destination is created in the System Template. To enable/disable a specific destination the PdurIPduGroup refers to the PduTriggering. atpVariation: The content of a PduR I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPdu, iPdu.pduTriggering, iPdu.variation Point.shortLabel vh.latestBindingTime=postBuild

Table 6.34: PdurIPduGroup

6.3.1 ContainerIPdu

IPdu collection is used to transport several (smaller) IPdus in one (large) ContainerIPdu. A possible use case for example is the extended payload size for Ethernet and CanFd in combination with the limited payload of Can and Lin, where Pdus from a Can network shall be routed onto an Ethernet network and then back to a Can again.

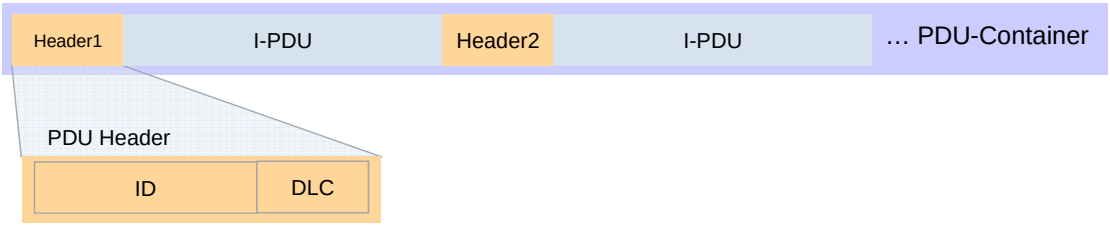

Figure 6.16: Layout of a ContainerIPdu if HeaderMode is used

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader) and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual

contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

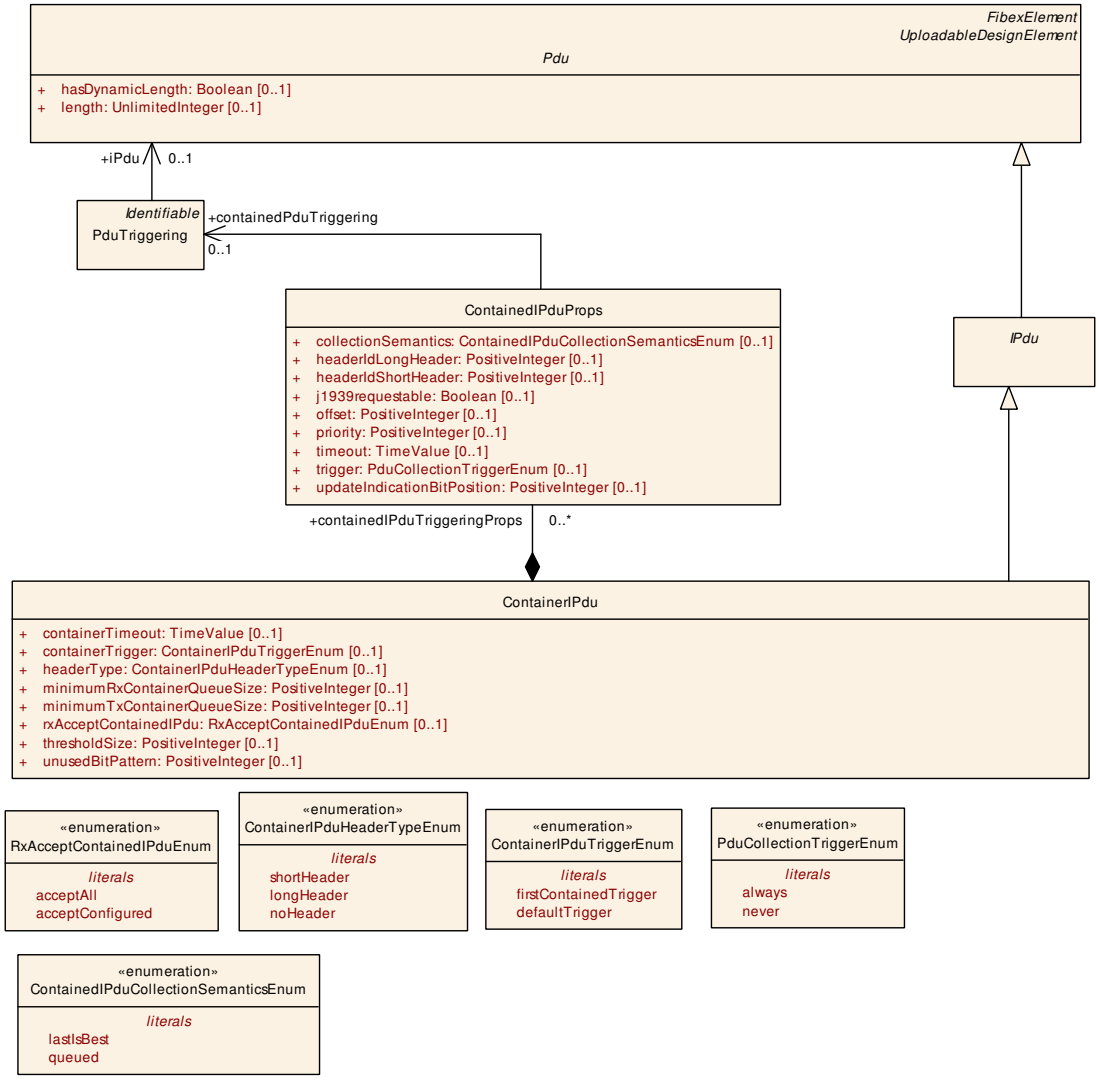

Figure 6.17: ContainerIPdu with ContainedIPduProps

Class	**ContainerIPdu**			
Note	Allows to collect several IPdus in one ContainerIPdu based on the headerType. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
containedIPdu TriggeringProps	ContainedIPduProps	*	aggr	Defines properties for an IPdu that is part of the ContainerIPdu. **Tags:** atp.Splitkey=containedIPduTriggeringProps

△

Class	**ContainerIPdu**			
containerTimeout	TimeValue	0..1	attr	When this timeout expires the ContainerIPdu is sent out. The respective timer is started when the first Ipdu is put into the ContainerIPdu. This attribute is ignored on receiver side.
containerTrigger	ContainerIPduTriggerEnum	0..1	attr	Defines if the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into it. This attribute shall be ignored on receiver side.
headerType	ContainerIPduHeaderTypeEnum	0..1	attr	Defines whether and which header type is used (header id and length).
minimumRxContainerQueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for received containers.
minimumTxContainerQueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for transmitted containers.
rxAcceptContainedIPdu	RxAcceptContainedIPduEnum	0..1	attr	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
thresholdSize	PositiveInteger	0..1	attr	Defines the size threshold which, when exceeded, triggers the sending of the ContainerIPdu although the maximum Pdu size has not been reached yet. Unit: byte.
unusedBitPattern	PositiveInteger	0..1	attr	IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Table 6.35: ContainerIPdu

[constr_9200] Existence of ContainerIPdu.headerType

Imposition time: CP: IT_SysDesc

⌈For each ContainerIPdu the attribute headerType shall exist.⌋

[constr_9201] Existence of ContainerIPdu.rxAcceptContainedIPdu

Imposition time: CP: IT_SysDesc

⌈For each ContainerIPdu the attribute rxAcceptContainedIPdu shall exist.⌋

[constr_3436] Value range of minimumTxContainerQueueSize and minimumRxContainerQueueSize

Imposition time: CP: IT_SysDesc

⌈If defined, the value of minimumTxContainerQueueSize and minimumRxContainerQueueSize shall be in the range of 0..255.⌋

Enumeration	**ContainerIPduTriggerEnum**
Note	Defines when the transmission of the ContainerIPdu shall be requested.
Aggregated by	ContainerIPdu.containerTrigger
Literal	***Description***
defaultTrigger	Defines that the transmission of the ContainerIPdu shall be requested when the default trigger conditions apply (e.g. timeout of threshold). **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**ContainerIPduTriggerEnum**
firstContained Trigger	Defines that the transmission of the ContainerIPdu shall be requested right after the first Contained IPdu was put into the ContainerIPdu. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.36: ContainerIPduTriggerEnum

Enumeration	**ContainerIPduHeaderTypeEnum**
Note	Is used to define the header type and size of ContainerIPdus. The header size includes the header id and the length information.
Aggregated by	ContainerIPdu.headerType
Literal	***Description***
longHeader	Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit **Tags:** atp.EnumerationLiteralIndex=0
noHeader	No Header is used and the location of each containedPdu in the ContainerPdu is statically configured. **Tags:** atp.EnumerationLiteralIndex=2
shortHeader	Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.37: ContainerIPduHeaderTypeEnum

Enumeration	**RxAcceptContainedIPduEnum**
Note	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
Aggregated by	ContainerIPdu.rxAcceptContainedIPdu
Literal	***Description***
acceptAll	No fixed set of containedIPdus is defined for reception, any known containedIPdu (based on header Id) shall be expected within this ContainerIPdu. **Tags:** atp.EnumerationLiteralIndex=0
acceptConfigured	A fixed set of containedIPdus is defined for reception. Only these assigned containedIPdus (based on headerId) are expected in this ContainerIPdu. If a not assigned containedIPdu is received within this ContainerIPdu this containedIPdu is discarded. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.38: RxAcceptContainedIPduEnum

Class	**ContainedIPduProps**			
Note	Defines the aspects of an IPdu which can be collected inside a ContainerIPdu.			
Base	*ARObject*			
Aggregated by	ContainerIPdu.containedIPduTriggeringProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
collection Semantics	ContainedIPdu CollectionSemantics Enum	0..1	attr	Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.

▽

△

Class	**ContainedIPduProps**			
containedPdu Triggering	PduTriggering	0..1	ref	Reference to Pdu for which the ContainedIPduProps are valid. **Tags:** atp.Splitkey=containedPduTriggering
headerIdLong Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.
headerIdShort Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.
j1939requestable	Boolean	0..1	attr	Contained I-PDU can be triggered by the J1939 request message.
offset	PositiveInteger	0..1	attr	Byte offset that describes the location of the Contained Pdu in the ContainerPdu if no header is used.
priority	PositiveInteger	0..1	attr	Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.
timeout	TimeValue	0..1	attr	Defines an IPdu specific sender timeout which adapts the ContainerIPdu transmission timer when this contained IPdu is put inside the ContainerIPdu: • f the ContainerIPdu transmission timer is already running, and the transmission timer value is greater than ContainedIPduProps.timeout, then the transmission timer value is reduced to ContainedIPdu Props.timeout. • If the ContainerIPdu transmission timer is already running, and the transmission timer value is less than or equal to ContainedIPduProps.timeout, then the transmission timer value is not changed. • If the ContainerIPdu transmission timer is not running, then the transmission timer value is set to Contained IPduProps.timeout. This attribute is ignored on receiver side.
trigger	PduCollectionTrigger Enum	0..1	attr	Defines whether this IPdu does trigger the sending of the ContainerIPdu. This attribute is ignored on receiver side.
update IndicationBit Position	PositiveInteger	0..1	attr	The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Table 6.39: ContainedIPduProps

[constr_9202] Existence of `ContainedIPduProps.collectionSemantics`

Imposition time: CP: `IT_SysDesc`

⌈For each `ContainedIPduProps` the attribute `collectionSemantics` shall exist.⌋

[constr_5268] Existence of `ContainedIPduProps.containedPduTriggering` reference

Imposition time: CP: `IT_SysDesc`

⌈If a `ContainedIPduProps` is aggregated at the `ContainerIPdu` in the role `ContainerIPdu.containedIPduTriggeringProps` then the reference `ContainedIPduProps.containedPduTriggering` shall exist.⌋

Enumeration	**ContainedIPduCollectionSemanticsEnum**
Note	Defines the collection semantics for ContainedIPdus.
Aggregated by	ContainedIPduProps.collectionSemantics
Literal	***Description***
lastIsBest	The ContainedIPdu data will be fetched via TriggerTransmit just before the transmission executes. **Tags:** atp.EnumerationLiteralIndex=0
queued	The ContainedIPdu data will instantly be stored to the ContainerIPdu in the context of the Transmit API. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.40: ContainedIPduCollectionSemanticsEnum

Enumeration	**PduCollectionTriggerEnum**
Note	Defines whether a Pdu contributes to the triggering of the data transmission if Pdu collection is enabled.
Aggregated by	ContainedIPduProps.trigger, *IEEE1722TpAcfBusPart*.collectionTrigger, SoConIPduIdentifier.pduCollectionTrigger
Literal	***Description***
always	Pdu will trigger the transmission of the data. **Tags:** atp.EnumerationLiteralIndex=0
never	Pdu will be buffered and will not trigger the transmission of the data. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.41: PduCollectionTriggerEnum

[TPS_SYST_02062] Allowed `ContainedIPduProps.headerIdLongHeader` and `ContainedIPduProps.headerIdShortHeader` values ⌈`ContainedIPduProps.headerIdLongHeader` and `ContainedIPduProps.headerIdShortHeader` shall be restricted to values different from 0 (all bits of the value set to 0).⌋

Since the header information is larger than 8 bit the byte ordering of the header inside the `ContainerIPdu` needs to be defined. This is done at System level. Thus all `ContainerIPdu`s have the header information in the same byte order within one System.

[TPS_SYST_02063] Byte order of `ContainerIPdu` header information ⌈The `System.containerIPduHeaderByteOrder` defines in which byte order the header information shall be put into the `ContainerIPdu`.⌋

[constr_3140] No `ByteOrderEnum.opaque` allowed for `System.containerIPduHeaderByteOrder`

Imposition time: CP: `IT_SysDesc`

⌈The values of `System.containerIPduHeaderByteOrder` are restricted to `ByteOrderEnum.mostSignificantByteFirst` and `ByteOrderEnum.mostSignificantByteLast`. I.e. the value `ByteOrderEnum.opaque` is not allowed.⌋

The following assumptions lead to the modeling of the `ContainerIPdu` structure:

- **[TPS_SYST_02097] Basic definition of contained IPdus** ⌈If an IPdu is referenced by a PduTriggering and that PduTriggering is referenced by a ContainedIPduProps in the role containedPduTriggering, then this IPdu can be collected inside a ContainerIPdu.⌋

- **[TPS_SYST_02098] Header id and header type of a contained IPdu** ⌈A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained IPdu does not need to have a headerId.⌋

- **[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side** ⌈On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.⌋

[constr_5362] Relation between ContainerIPdu and contained PduTriggerings on sender side

Imposition time: CP: IT_SysDesc

⌈In the scope of one EcuInstance, if a PduTriggering has a reference to an IPduPort where attribute communicationDirection is set to the value out, then that PduTriggering shall only be referenced at most once by ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering.⌋

The ContainerIPdu defines which IPdus may be collected inside that ContainerIPdu (ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering). Dynamic assignment of a contained IPdu to different ContainerIPdus during run-time is not supported by the IPdu multiplexer. Nevertheless it is allowed to collect an IPdu in several ContainerIPdus since each of those ContainerIPdus can be transmitted individually (on the same or on a different PhysicalChannel).

If a ContainerIPdu is transmitted on different PhysicalChannels this can be described in a System Description by a single Pdu element that is referenced by PduTriggerings of the different PhysicalChannels. Details on the ownership of PduTriggerings can be found in section 6.1.1.

[constr_3141] Only IPdus shall be part of a ContainerIPdu

Imposition time: CP: IT_SysDesc

⌈The PduTriggering which is referenced in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering shall refer to a subclass of an IPdu in the role PduTriggering.iPdu.⌋

Only subclasses of IPdus are handled by the PduR and therefore are available for the ContainerIPdu.

For the sender side this assignment defines which IPdus may be collected inside this ContainerIPdu. For the receiver side this assignment may be omitted if ContainerIPdu.rxAcceptContainedIPdu is set to RxAcceptContainedIPduEnum.acceptAll.

[TPS_SYST_02064] Reception acceptance of contained IPdus ⌈ContainerIPdu.rxAcceptContainedIPdu defines for the receiver side whether the list of ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering is a closed set.

If ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured, only those IPdus (based on headerId) which are referenced by this ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering are extracted from this ContainerIPdu.

If ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll, the IPdus (based on headerId) which are referenced by this ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering are expected inside this ContainerIPdu but also any other IPdu is extracted which is referenced by any other ContainerIPdu.rxAcceptContainedIPdu with RxAcceptContainedIPduEnum.acceptAll set.⌋

Thus all referenced IPdus which are referenced from ContainerIPdu.rxAcceptContainedIPdu with RxAcceptContainedIPduEnum.acceptAll form the set of IPdus which are considered for the reception of ANY ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll.

For the receiver side ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering may be omitted if ContainerIPdu.rxAcceptContainedIPdu is set to RxAcceptContainedIPduEnum.acceptAll. Such a ContainerIPdu will accept any of the contained IPdus from the acceptAll IPdu set if they have the headerId (headerIdShortHeader or headerIdLongHeader) defined by the ContainerIPdu.headerType configured.

There are use-cases where several IPdus with identical content are transported on the vehicle networks. One motivation to design the communication structure like this is that the communication timing attributes may be different and thus need to be represented by different ISignalIPdus (i.e. different ISignalIPdu.iPduTimingSpecification, e.g. to support the transmission on Can and Fr networks with different transmission modes). When the content of the IPdus is identical they can be transported using the same header Id. Thus it is transparent for the receiver via which channel a specific IPdu arrived in case of ContainerIPdu with acceptAll setting.

In such a setup it may occur that several PduTriggerings for IPdus with identical header IDs are defined as candidates for the set of IPdus which are considered for

the reception of acceptAll ContainerIPdus. This is considered a valid setup and shall be resolved during derivation to ECU Configuration.

The Ecu Configuration for ContainerIPdu.rxAcceptContainedIPdu = acceptAll uses a set of Contained Rx Pdus without a relation to Container Pdus. This set is defined during derivation of the Ecu Configuration from the System Description. If duplicates occur they are only included once in the set of Contained Rx Pdus.

In such a case there is additional information required to select which of these PduTriggerings shall be taken for the configuration and which shall be omitted.

[TPS_SYST_02196] PduTriggering is referenced by several ContainerIPdus ⌈In case a PduTriggering is referenced in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering by several ContainerIPdus with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptAll and has a reference to an IPduPort with direction IN then this PduTriggering shall only be considered once for the set of IPdus which are considered for the reception of acceptAll ContainerIPdus.⌋

[constr_3403] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used

Imposition time: CP: IT_SysDesc

⌈If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.⌋

[TPS_SYST_03014] Transmission triggering by the first contained IPdu put into a ContainerIPdu ⌈The attribute ContainerIPdu.containerTrigger determines whether the transmission of a ContainerIPdu shall be requested when the first contained IPdu was put into the ContainerIPdu.

In case containerTrigger equals firstContainedTrigger the transmission of the ContainerIPdu shall be requested when the first contained IPdu is put into the ContainerIPdu.

In case containerTrigger equals defaultTrigger the transmission of the ContainerIPdu shall be requested when the other trigger conditions defined by the ContainerIPdu are fulfilled (e.g. containerTimeout, thresholdSize).⌋

Note: This trigger condition is independent from PduCollectionTriggerEnum.always which is defined for specific IPdus. With the attribute ContainerIPdu.containerTrigger = firstContainedTrigger on the other hand, any contained IPdu will trigger the ContainerIPdu transmission.

Rationale for this trigger condition is the efficient usage (allow the ContainerIPdus to reach a certain fill level) of triggered transmission on time- (containerTrigger typically set to firstContainedTrigger) and event-driven (containerTrigger typically set to defaultTrigger) buses.

The ContainedIPduProps defines a header Id per ContainerIPdu.headerType which shall be used in the Pdu header of the ContainerIPdu in case that the headerType is set to shortHeader or longHeader. In case that the headerType is set to noHeader the layout of IPdus in the ContainerIPdu is statically configured and no header Id is required.

[constr_3454] Unique headerIdLongHeader for acceptConfigured

Imposition time: CP: IT_SysDesc

⌈For a ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured and ContainerIPdu.headerType = longHeader the following shall apply: All referenced IPdus (via ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering) shall have a unique ContainedIPduProps.headerIdLongHeader within the scope of this ContainerIPdu.⌋

[constr_3455] Unique headerIdShortHeader for acceptConfigured

Imposition time: CP: IT_SysDesc

⌈For a ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu = RxAcceptContainedIPduEnum.acceptConfigured and ContainerIPdu.headerType = shortHeader the following shall apply: All referenced IPdus (via ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering) shall have a unique ContainedIPduProps.headerIdShortHeader within the scope of this ContainerIPdu.⌋

Note: With [constr_3454] and [constr_3455] it is possible to have the same header Id value received in different ContainerIPdus. It just has to be guaranteed that in the scope of one ContainerIPdu the reception of header Id is unambiguous.

[constr_3142] Mandatory headerIdLongHeader for longHeader

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = longHeader the ContainedIPduProps.headerIdLongHeader shall be defined.⌋

[constr_3143] Mandatory headerIdShortHeader for shortHeader

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = shortHeader the ContainedIPduProps.headerIdShortHeader shall be defined.⌋

[constr_3402] Mandatory `offset` if `noHeader` is used

Imposition time: CP: IT_SysDesc

⌈For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering with ContainerIPdu.headerType = noHeader the ContainedIPduProps.offset shall be defined.⌋

[constr_3404] Usage of `ContainedIPduProps.updateIndicationBitPosition`

Imposition time: CP: IT_SysDesc

⌈ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering is set to noHeader.⌋

[constr_3405] Dynamic Length `IPdu` inside of a static configured `ContainerIPdu`

Imposition time: CP: IT_SysDesc

⌈Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with headerType set to noHeader) is allowed to be a dynamic length IPdu (i.e, a contained IPdu that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained IPdus of a ContainerIPdu with static container layout have to be static length IPdus.⌋

[TPS_SYST_02065] Contained `IPdu` specific transmission timeout ⌈The IPdu specific transmission timeout can be specified at ContainedIPduProps.timeout. If no ContainedIPduProps.timeout is provided the timeout from the ContainerIPdu shall be used (ContainerIPdu.containerTimeout).⌋

The case where neither the ContainerIPdu.containerTimeout nor the ContainedIPduProps.timeout is provided, will result in no time-based triggering of ContainerIPdus which might lead to long delays or no transmission at all if no other sending condition for this ContainerIPdu does occur (e.g. no further IPdu is collected inside this ContainerIPdu).

[TPS_SYST_02066] `ContainerIPdu.thresholdSize` ⌈The attribute ContainerIPdu.thresholdSize defines the threshold when a ContainerIPdu shall be triggered for transmission. If the payload size of the ContainerIPdu exceeds the value of thresholdSize this ContainerIPdu shall be transmitted.⌋

Note: The ContainerIPdu.thresholdSize supports the definition of a transmission threshold which takes the data transmission model of the communication into account. Especially when operating with variable length IPdus, only the maximum length of these IPdus is defined in the System Description. Only having the maximumLength information it is not possible to derive a sensible threshold for the ContainerIPdu

this variable length IPdu is collected in. Thus a ContainerIPdu would wait for further contained IPdus. Using a transmission model it can be calculated that the average size contained IPdu will not fit into that ContainerIPdu anymore and provide this as a requirement in ContainerIPdu.thresholdSize.

Another use case for the ContainedIPduProps is to support the usage of optimized trigger transmit collection of IPdus in ContainerIPdus. Therefore it is necessary to distinguish between contained IPdus with lastIsBest (will be fetched via trigger transmit just before the transmission executes) and those with queued semantics (will instantly be stored in the context of the transmit API). This distinction is possible on the level of single contained IPdus with the attribute collectionSemantics.

For all intents and purposes, the different handling of contained IPdus depending on the semantics is supported by the attribute ContainedIPduProps.collectionSemantics that allows the individual setting of the intended semantics per contained IPdu.

[constr_3517] Consistent setting of ContainedIPduProps.collectionSemantics in the context of one ContainerIPdu

Imposition time: CP: IT_SysDesc

⌈The value of the attribute ContainedIPduProps.collectionSemantics shall be identical for all contained IPdus within the context of a given ContainerIPdu.⌋

[constr_3488] Value range of ContainedIPduProps.priority

Imposition time: CP: IT_SysDesc

⌈If defined, the value of ContainedIPduProps.priority shall be in the range of 0..255.⌋

[constr_3489] ContainedIPduProps.priority is only applicable if a ContainerIPdu header is used

Imposition time: CP: IT_SysDesc

⌈ContainedIPduProps.priority is only applicable if the headerType of the ContainerIPdu is set to shortHeader or longHeader.⌋

[constr_3490] ContainedIPduProps.priority is only applicable if collectionSemantics is set to lastIsBest

Imposition time: CP: IT_SysDesc

⌈ContainedIPduProps.priority is only applicable if ContainedIPduProps.collectionSemantics is set to lastIsBest.⌋

6.3.2 SecuredIPdu

AUTOSAR supports an authentication mechanism for critical data on the level of Pdus. Optionally, confidentiality can be ensured using authenticated encryption.

[TPS_SYST_02060] SecuredIPdus ⌈SecuredIPdu shall be used to describe an IPdu that is protected against unauthorized manipulation and replay attacks.⌋

Please note that several SecuredIPdus may exist with the same dataId value since the same data may be transported on different channels.

[TPS_SYST_02148] Meaning of useAsCryptographicIPdu that is not set or set to false ⌈If useAsCryptographicIPdu is not set or set to false the SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator).⌋

[TPS_SYST_02149] Meaning of useAsCryptographicIPdu that is set to true ⌈If useAsCryptographicIPdu is set to true the SecuredIPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the PduTriggering that is referenced with the payload reference.⌋

The attribute useAsCryptographicIPdu decides whether one single Pdu or two Pdus are transferred on the communication bus. In either case always two IPdus shall be modeled:

- SecuredIPdu with a PduTriggering
- payload IPdu with a PduTriggering

[TPS_SYST_02172] Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to false ⌈If the useAsCryptographicIPdu is set to false only the SecuredIPdu shall be either

- mapped into a Frame by the PduToFrameMapping or
- mapped into a StaticSocketConnection or
- assigned to an AbstractServiceInstance via PduActivationRoutingGroup that references the SoConIPduIdentifier that represents the SecuredIPdu or
- assigned to a ContainerIPdu.

⌋

[TPS_SYST_02445] SecuredIPdus with encryption ⌈SecuredIPdu that have a SecuredIPdu.secureCommunicationProps with SecureCommunicationProps.useAuthenticatedEncryption set to true shall be used to describe an IPdu that is encrypted using Authenticated Encryption with Associated Data (AEAD).⌋

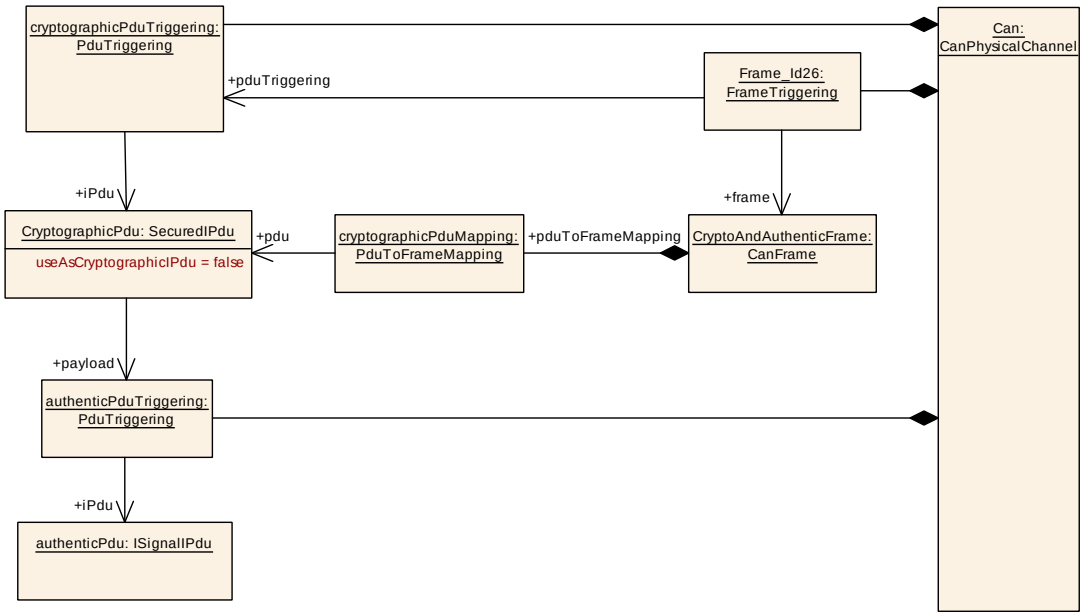

Figure 6.18: Example for the modeling of `SecuredIPdu` with `useAsCryptographicIPdu` set to false

If a `SecuredIPdu` is transmitted on different `PhysicalChannel`s this can be described in a System Description by a single `SecuredIPdu` element that is referenced by `PduTriggering`s of the different `PhysicalChannel`s. Details on the ownership of `PduTriggering`s can be found in section 6.1.1.

[TPS_SYST_02173] Modeling of `SecuredIPdu` in case `useAsCryptographicIPdu` is set to true ⌈If the `useAsCryptographicIPdu` is set to true then the `SecuredIPdu` and the `payload IPdu` shall be either

- mapped into `Frame`s by the `PduToFrameMapping` or
- assigned to `StaticSocketConnection` or
- assigned to `AbstractServiceInstance`s via `PduActivationRoutingGroup` that references the `SoConIPduIdentifier`s that represent the `SecuredIPdu` and the `payload IPdu` or
- assigned to `ContainerIPdu`s.

⌋

[constr_5259] `PduTriggering`s and `FrameTriggering`s of `SecuredIPdu` with `useAsCryptographicIPdu` = true

Imposition time: CP: `IT_SysDesc`

⌈In case that a `SecuredIPdu` is defined with `useAsCryptographicIPdu` = true as described by [TPS_SYST_02173] then:

- the `PduTriggering` of the `AuthenticPdu`
- the `PduTriggering` of the `CryptographicPdu`

- the FrameTriggering that references the Frame to which the AuthenticPdu is mapped
- the FrameTriggering that references the Frame to which the CryptographicPdu is mapped

shall be aggregated by the same PhysicalChannel.⌋

[TPS_SYST_02361] PduR Fan-out of SecuredIPdu with useAsCryptographicIPdu = true ⌈If a PduR fan-out of a SecuredIPdu with useAsCryptographicIPdu = true is defined then [constr_5259] shall be considered on all PhysicalChannels on which the SecuredIPdu is transmitted.⌋

In other words the PduTriggerings of the AuthenticPdu and the CryptographicPdu and the FrameTriggerings that reference the Frames to which the AuthenticPdu and CryptographicPdu are mapped need to be defined on all PhysicalChannels on which the SecuredIPdu is transmitted.

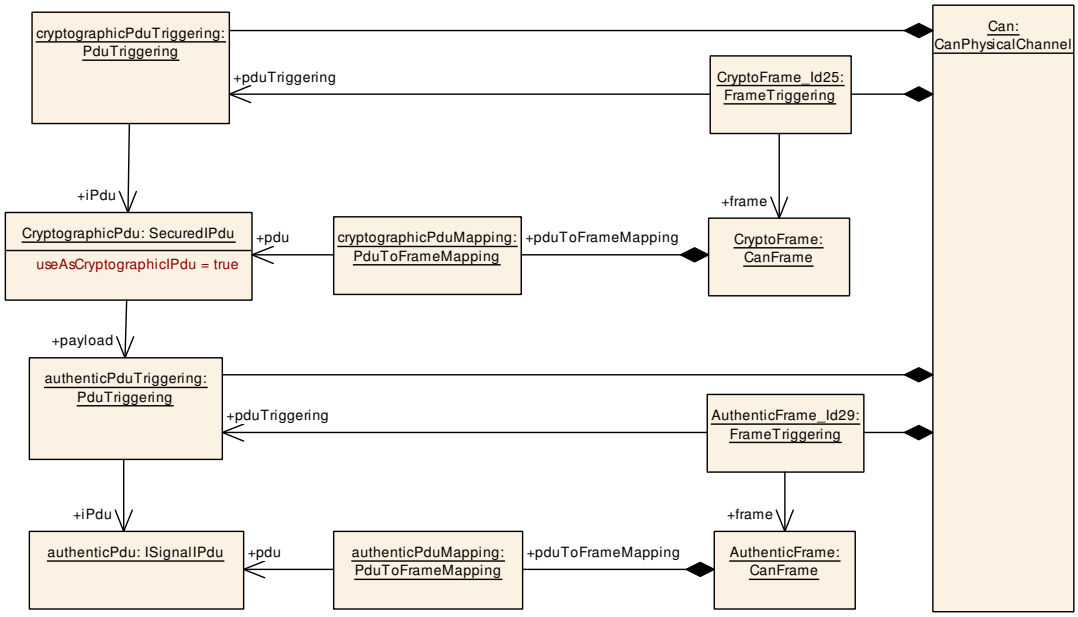

Figure 6.19: Example for the modeling of SecuredIPdu with useAsCryptographicIPdu set to true

Please note that [TPS_SYST_02059] defines that the PduTriggerings of the SecuredIPdu and PduTriggerings of the payload IPdu shall both reference IPduPorts.

A SecuredIPdu defines freshness properties by referencing the reusable SecureCommunicationFreshnessProps in the role freshnessProps. The authentication properties are defined by reusable SecureCommunicationAuthenticationProps that are referenced in the role authenticationProps. Configuration settings that are specific to the SecuredIPdu are defined in SecureCommunicationProps.

Please note that if the SecuredIPdu does not reference the SecureCommunicationFreshnessProps in the role freshnessProps the freshness value will not be included in the SecuredIPdu.

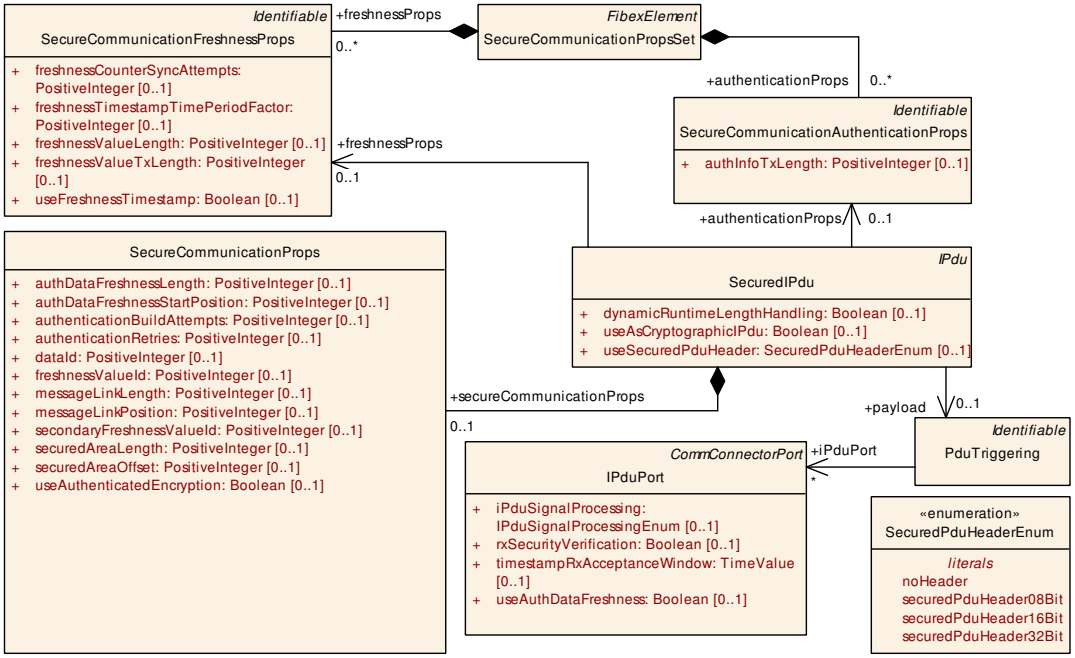

Figure 6.20: SecuredIPdu with SecureCommunicationProps

Class	**SecuredIPdu**			
Note	If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator). If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Props	SecureCommunication AuthenticationProps	0..1	ref	Reference to authentication properties that are valid for this SecuredIPdu. **Tags:** atp.Splitkey=authenticationProps
dynamic RuntimeLength Handling	Boolean	0..1	attr	Defines whether the length information for handling this SecuredIPdu with SecuredIPdu.useSecuredPdu Header=noHeader is taken from the configuration or from the actually provided length information during runtime. true: SecuredIPdu length information is taken from the actually provided length information during runtime. false: SecuredIPdu length information is taken from the configuration.
freshnessProps	SecureCommunication FreshnessProps	0..1	ref	Reference to freshness properties that are valid for this SecuredIPdu. **Tags:** atp.Splitkey=freshnessProps

△

Class	**SecuredIPdu**			
payload	PduTriggering	0..1	ref	Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks. **Tags:** atp.Splitkey=payload
secure Communication Props	SecureCommunication Props	0..1	aggr	Specific configuration properties for this SecuredIPdu. **Tags:** atp.Splitkey=secureCommunicationProps
useAs Cryptographic IPdu	Boolean	0..1	attr	If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.
useSecuredPdu Header	SecuredPduHeader Enum	0..1	attr	This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

Table 6.42: SecuredIPdu

[constr_9203] Existence of `SecuredIPdu.payload`

Imposition time: CP: `IT_SysDesc`

⌈For each `SecuredIPdu`, the reference to `PduTriggering` in the role `payload` shall exist.⌋

[constr_9204] Existence of `SecuredIPdu.secureCommunicationProps`

Imposition time: CP: `IT_SysDesc`

⌈For each `SecuredIPdu` the aggregation of `SecureCommunicationProps` in the role `secureCommunicationProps`, shall exist.⌋

Enumeration	**SecuredPduHeaderEnum**
Note	Defines the header which will be inserted into the SecuredIPdu.
Aggregated by	SecuredIPdu.useSecuredPduHeader
Literal	***Description***
noHeader	No header included in the SecuredPdu. **Tags:** atp.EnumerationLiteralIndex=0
securedPdu Header08Bit	8 Bit Secured I-PDU Header included in the Secured I-PDU. **Tags:** atp.EnumerationLiteralIndex=1
securedPdu Header16Bit	16 Bit Secured I-PDU Header included in the Secured I-PDU. **Tags:** atp.EnumerationLiteralIndex=2
securedPdu Header32Bit	32 Bit Secured I-PDU Header included in the Secured I-PDU. **Tags:** atp.EnumerationLiteralIndex=3

Table 6.43: SecuredPduHeaderEnum

Class	**SecureCommunicationProps**			
Note	This meta-class contains configuration settings that are specific for an individual SecuredIPdu.			
Base	*ARObject*			
Aggregated by	SecuredIPdu.secureCommunicationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authData Freshness Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentic PDU data that is passed to the SWC that verifies and generates the Freshness.
authData FreshnessStart Position	PositiveInteger	0..1	attr	This value determines the start position in bits of the Authentic PDU that shall be passed on to the SWC that verifies and generates the Freshness. The bit counting is done according to TPS_SYST_01068.
authentication BuildAttempts	PositiveInteger	0..1	attr	This attribute specifies the number of authentication build attempts.
authentication Retries	PositiveInteger	0..1	attr	This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given SecuredIPdu. If zero is set than only one authentication attempt is done.
dataId	PositiveInteger	0..1	attr	This attribute defines a numerical identifier for the Secured I-PDU.
freshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.
messageLink Length	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.
messageLink Position	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.
secondary FreshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Secondary Freshness Value. The Secondary Freshness Value might be a normal counter or a time value. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
securedArea Length	PositiveInteger	0..1	attr	This attribute defines the length in bytes of the area within the payload Pdu which will be secured.
securedArea Offset	PositiveInteger	0..1	attr	This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.
use Authenticated Encryption	Boolean	0..1	attr	A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If it set to TRUE, Authenticated Encryption with Associated Data (AEAD) shall be applied.

Table 6.44: SecureCommunicationProps

`SecureCommunicationProps.freshnessValueId` does not need to be defined in case of a time-based Freshness Value. In case of a counter-based Freshness Value the `freshnessValueId` may be defined locally in the Ecuc or may be provided by the OEM in a System Description (this may be useful if several ECUs need to sync the freshness for a certain `freshnessValueId`).

[constr_9205] Existence of SecureCommunicationProps.dataId

Imposition time: CP: IT_SysDesc

⌈For each SecureCommunicationProps, the attribute dataId shall exist.⌋

Class	**SecureCommunicationPropsSet**			
Note	Collection of properties used to configure SecuredIPdus. **Tags:** atp.recommendedPackage=SecureCommunicationPropsSets			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication Props	SecureCommunication AuthenticationProps	*	aggr	Authentication properties used to configure Secured IPdus. **Tags:** atp.Splitkey=authenticationProps, authentication Props.shortName
freshnessProps	SecureCommunication FreshnessProps	*	aggr	Freshness properties used to configure SecuredIPdus. **Tags:** atp.Splitkey=freshnessProps, freshnessProps.short Name

Table 6.45: SecureCommunicationPropsSet

Class	**SecureCommunicationFreshnessProps**			
Note	Freshness properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.freshnessProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
freshness CounterSync Attempts	PositiveInteger	0..1	attr	This attribute defines the number of Freshness Counter re-synchronization attempts when a verification failed for a Secured I-PDU. If the value is zero, there will be no additional verification attempt to synchronize with a potentially better fitting Freshness Counter value. This attribute is only applicable if useFreshnessTimestamp is FALSE.
freshness TimestampTime PeriodFactor	PositiveInteger	0..1	attr	This attribute defines a factor that specifies the time period for the Freshness Timestamp. It holds a multiplication factor that specifies the concrete meaning of a Freshness Timestamp increment by one on basis of microseconds.
freshnessValue Length	PositiveInteger	0..1	attr	This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.
freshnessValue TxLength	PositiveInteger	0..1	attr	This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.
useFreshness Timestamp	Boolean	0..1	attr	This attribute specifies whether the Freshness Value is generated through individual Freshness Counters or by a Timestamps. The value is set to TRUE when Timestamps are used.

Table 6.46: SecureCommunicationFreshnessProps

Class	SecureCommunicationAuthenticationProps			
Note	Authentication properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.authenticationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
authInfoTx Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.

Table 6.47: SecureCommunicationAuthenticationProps

[TPS_SYST_02281] Definition of SecuredIPdu.authDataFreshnessStartPosition ⌈SecuredIPdu.authDataFreshnessStartPosition defines the position of the most significant bit of the Freshness Value in the Pdu.⌋

[TPS_SYST_02282] Definition of SecuredIPdu.messageLinkPosition ⌈SecuredIPdu.messageLinkPosition defines the position of the most significant bit of the Message Link in the Pdu.⌋

Please note that the bit counting and bit order is defined by [TPS_SYST_01068] and [TPS_SYST_01069] and is also valid to determine the position of the authDataFreshness and messageLink.

[constr_3136] Allowed payload of SecuredIPdus

Imposition time: CP: IT_SysDesc

⌈SecuredIPdus are allowed to reference PduTriggerings of ISignalIPdus, ContainerIPdus, DcmIPdus, MultiplexedIPdus, GeneralPurposeIPdus with category SOMEIP_SEGMENTED_IPDU or VDP, and UserDefinedIPdus.⌋

Please note that it is currently not possible to refer to a SecuredIPdu in the roles MultiplexedIPdu.staticPart and MultiplexedIPdu.dynamicPart and therefore it is not possible to include SecuredIPdus in either the static part or dynamic part of a MultiplexedIPdu. In other words, a MultiplexedIPdu can be the payload of a SecuredIPdu, but conversely a SecuredIPdu cannot become a part of a MultiplexedIPdu.

[TPS_SYST_02171] Secured Area in payload Pdu ⌈The area within the payload Pdu that is secured is specified by the securedAreaOffset and securedAreaLength. In case that these two attributes are not configured the complete payload Pdu is secured.⌋

[constr_3399] Existence of securedAreaOffset and securedAreaLength

Imposition time: CP: IT_SysDesc

⌈If the securedAreaOffset is defined then the securedAreaLength shall be defined as well and vice versa.⌋

[constr_3716] SecuredIPdu.dynamicRuntimeLengthHandling for dynamic length Pdus

Imposition time: CP: IT_SysDesc

⌈If a PduTriggering is referenced from a SecuredIPdu in the role payload and the Pdu referenced by the PduTriggering in the role iPdu qualifies according to [TPS_SYST_03085] to be of dynamic length, then the SecuredIPdu shall have the attribute SecuredIPdu.dynamicRuntimeLengthHandling set to true.⌋

[constr_3717] SecuredIPdu.dynamicRuntimeLengthHandling for gateway operation with IPduMapping.pduMaxLength defined

Imposition time: CP: IT_SysDesc

⌈If a PduTriggering refers to a SecuredIPdu in the role iPdu and that PduTriggering is used in an IPduMapping where a pduMaxLength value is defined (either in the role IPduMapping.sourceIPdu or TargetIPduRef.targetIPdu), then the SecuredIPdu shall have the attribute SecuredIPdu.dynamicRuntimeLengthHandling set to true.⌋

[constr_3718] Minimum length of SecuredIPdus

Imposition time: CP: IT_SysDesc

⌈If a SecuredIPdu has the attribute useAsCryptographicIPdu set to false, then the length attribute of that SecuredIPdu shall be at least the sum of the payload Pdu.length and SecuredIPdu.authenticationProps.authInfoTxLength.⌋

[TPS_SYST_02152] Security profile ⌈The Security profile is defined by SecureCommunicationFreshnessProps.category and by SecureCommunicationAuthenticationProps.category.⌋

[constr_3324] Category of SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps

Imposition time: CP: IT_SysDesc

⌈SecureCommunicationFreshnessProps that is referenced by a SecuredIPdu in the role freshnessProps shall have the same category value as the SecureCommunicationAuthenticationProps that is referenced by the same SecuredIPdu in the role authenticationProps.⌋

[TPS_SYST_02153] Standardized values for the attribute category of meta-class SecureCommunicationFreshnessProps ⌈The following values of the attribute category of meta-class SecureCommunicationFreshnessProps are reserved by the AUTOSAR standard: PROFILE_01, PROFILE_02, PROFILE_03, PROFILE_04.⌋

[TPS_SYST_02154] Standardized values for the attribute category of meta-class SecureCommunicationAuthenticationProps ⌈The following values of the attribute category of meta-class SecureCommunicationAuthenticationProps

are reserved by the AUTOSAR standard: PROFILE_01, PROFILE_02, PROFILE_03, PROFILE_04.⌋

[constr_3325] SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute value settings for standardized AUTOSAR security profiles

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈

Attributes	PROFILE_01	PROFILE_02	PROFILE_03	PROFILE_04
algorithmFamily	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES	CRYPTO_ ALGOFAM_ AES
algorithmMode	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ CMAC	CRYPTO_ ALGOMODE_ GCM_SIV
length	128 bits	128 bits	128 bits	128 bits
authInfoTxLength	24 bits	24 bits	28 bits	128 bits
freshnessValueLength	Not specified	0 bits	64 bits	96 bits
freshnessValueTx-Length	8 bits	0 bits	4 bits	96 bits

⌋

In other words if you want to define a SecuredIPdu in a particular Profile you have to reference the SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps and the PduTriggering of this SecuredIPdu that shall reference a SecOcCryptoServiceMapping that points to CryptoServicePrimitive that contributes the authentication Algorithm and the CryptoServiceKey that contributes the length.

[constr_3339] Relation between authDataFreshnessStartPosition, authDataFreshnessLength and useAuthDataFreshness

Imposition time: CP: IT_SysDesc

⌈If authDataFreshnessStartPosition and authDataFreshnessLength are set to a value for a SecuredIPdu then the useAuthDataFreshness shall be set as well to a value on all IPduPorts with communicationDirection = in that are referenced by a PduTriggering of the SecuredIPdu.⌋

[TPS_SYST_02189] Setting of useSecuredPduHeader attribute ⌈The useSecuredPduHeader shall be set to a value other than noHeader if the length of the payload Pdu is dynamic and is transmitted directly over a network which may insert padding bytes depending on the length (e.g. CANFD, Flexray).⌋

In case the SecuredIPdu is contained in a ContainerIPdu or is a TP-N-SDU, its length is correctly passed by the lower layer. In these cases the *SecuredIPduHeader* is not needed.

Please note that the dynamic-length Pdu can be an ISignalIPdu that contains a SystemSignal with dynamicLength set to true. In general it is not possible to run

diagnostics on fixed-length Pdus. Therefore, there is a probability that at least a subset of DcmIPdus and UserDefinedIPdus can have dynamic length.

[constr_3406] All signals before authDataFreshnessStartPosition shall have a static length

Imposition time: CP: IT_SysDesc

⌈In case that

- an ISignalIPdu is referenced by the SecuredIPdu with the payload reference via the PduTriggering and
- the authDataFreshnessStartPosition and authDataFreshnessLength define the area in the ISignalIPdu that is taken to verify and generate the Freshness then

all ISignals that are mapped into the ISignalIPdu in front of the configured authDataFreshnessStartPosition shall have a static length.⌋

Please note that parts of the Authentic IPdu can be used as freshness when authDataFreshnessStartPosition and authDataFreshnessLength are defined. But therefore the part of the Authentic IPdu to be used as the freshness has to be always available at same position in the Authentic IPdu.

[constr_3407] Freshness Value in Authentic IPdu is not allowed to be used in case of ContainerIPdu with a dynamic layout

Imposition time: CP: IT_SysDesc

⌈If a ContainerIPdu that is referenced by the SecuredIPdu with the payload reference via the PduTriggering contains a dynamic layout (i.e. ContainerIPdu.headerType is set to longHeader or shortHeader) and multiple contained IPdus then each IPduPort that is referenced by the PduTriggering of the SecuredIPdu shall have the attribute useAuthDataFreshness set to false.⌋

Please note that for ContainerIPdus with a dynamic layout it cannot be ensured which contained IPdu will be put in which position (depends on various timing and trigger conditions). Therefore [constr_3407] applies.

[constr_5060] Mapping of a SecuredIPdu into a LinFrame is not allowed

Imposition time: CP: IT_SysDesc

⌈The mapping of a SecuredIPdu into a LinFrame with a PduToFrameMapping is not allowed.⌋

In other words the usage of Pdus that are secured by SecOC is not allowed on a LIN Network.

6.3.2.1 Crypto Infrastructure for SecuredIPdu

From the cryptographic point of view, the usage of SecuredIPdu is connected to the application of two cryptographic operations, MacGenerate (for sending the SecuredIPdu) and MacVerify (for receiving the SecuredIPdu).

There are use cases for OEMs to already provide a pre-configuration of the crypto stack specifically for the handling of SecuredIPdu.

In order to support these use cases model elements in the meta-model are defined that allow for modeling a "crypto infrastructure" that facilitates the derivation of the configuration of operations MacGenerate and MacVerify in the crypto stack.

The formalization of a Pdu that is sent or received is based on the PduTriggering and its reference to an IPduPort, specifically by interpreting the attribute IPduPort.communicationDirection. The details of the approach are summarized in Figure 6.21.

[TPS_SYST_05020] Semantics of CryptoServiceMapping ⌈Meta-class CryptoServiceMapping represents an abstract base class for the creation of mappings in the context of cryptographic operations. Concrete sub-classes define the mapping with respect to specific cryptographic use cases, e.g. SecOC, TLS.⌋

Class	***CryptoServiceMapping*** (abstract)			
Note	This meta-class represents an abstract base class for specializations of crypto service mappings.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	SecOcCryptoServiceMapping, TlsCryptoServiceMapping			
Aggregated by	SystemMapping.cryptoServiceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.48: CryptoServiceMapping

[TPS_SYST_05021] Semantics of SecOcCryptoServiceMapping ⌈Meta-class SecOcCryptoServiceMapping represents the mapping functionality required for the configuration of PDU-based secure communication by means of the SecOc.

In particular, the SecOcCryptoServiceMapping associates a CryptoServicePrimitive with the applicable CryptoServiceKey.⌋

Class	**SecOcCryptoServiceMapping**			
Note	This meta-class has the ability to represent a crypto service mapping for the Pdu-based communication via SecOC.			
Base	*ARObject*, *CryptoServiceMapping*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cryptoServiceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	SecOcCryptoServiceMapping			
authentication	CryptoServicePrimitive	0..1	ref	This reference identifies the applicable crypto primitive for the authentication. **Tags:** atp.Splitkey=authentication
cryptoService Key	CryptoServiceKey	0..1	ref	This reference identifies the applicable crypto key. **Tags:** atp.Splitkey=cryptoServiceKey
cryptoService Queue	CryptoServiceQueue	0..1	ref	This reference identifies the CryptoServiceQueue the processing of this SecOcCryptoServiceMapping shall be performed in. **Tags:** atp.Splitkey=cryptoServiceQueue

Table 6.49: SecOcCryptoServiceMapping

[TPS_SYST_05022] Semantics of PduTriggering.secOcCryptoService-Mapping ⌈The reference PduTriggering.secOcCryptoServiceMapping allows for modeling the relation of the usages (send, receive) of a SecuredIPdu to a given CryptoServiceMapping and thereby distinguish between the configuration of cryptographic operations MacGenerate and MacVerify.⌋

In other words, the cryptographic use case is connected to the value of attribute communicationDirection of the IPduPort that is referenced by an Pdu-Triggering that also references a SecOcCryptoServiceMapping:

- If the value of communicationDirection is set to in then the cryptographic use case MacVerify applies.
- If the value of communicationDirection is set to out then the cryptographic use case MacGenerate applies.

[constr_1669] Existence of PduTriggering.secOcCryptoServiceMapping

Imposition time: CP: IT_SysDesc

⌈The reference PduTriggering.secOcCryptoServiceMapping shall only exist if the PduTriggering also references a SecuredIPdu in the role iPdu.⌋

As the SecOcCryptoServiceMapping is referenced by a PduTriggering the SecOcCryptoServiceMapping needs to work for both the sender and the receivers of the corresponding Pdu in the context of a System of category SYSTEM_DESCRIPTION/SYSTEM_EXTRACT as well as in the context of a System of category ECU_EXTRACT.

[TPS_SYST_05023] Semantics of CryptoServicePrimitive ⌈Meta-class CryptoServicePrimitive allows for the description of the applicable cryptographic algorithm.⌋

Class	**CryptoServicePrimitive**			
Note	This meta-class has the ability to represent a crypto primitive. **Tags:** atp.recommendedPackage=CryptoPrimitives			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
algorithmFamily	String	0..1	attr	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.
algorithmMode	String	0..1	attr	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.
algorithm Secondary Family	String	0..1	attr	This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive. The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.

Table 6.50: CryptoServicePrimitive

[TPS_SYST_05024] Semantics of `CryptoServiceKey` ⌈Meta-class `CryptoServiceKey` allows for the description of the applicable cryptographic key. The ability to aggregate a `ValueSpecification` in the role `developmentValue` shall be used to distribute development keys to suppliers such that crypto functionality can be adequately verified during development.⌋

[constr_5334] Supported values for `CryptoServiceKey.length`

Imposition time: CP: `IT_SysDesc`

⌈The values defined for `CryptoServiceKey.length` shall be multiple of 8.⌋

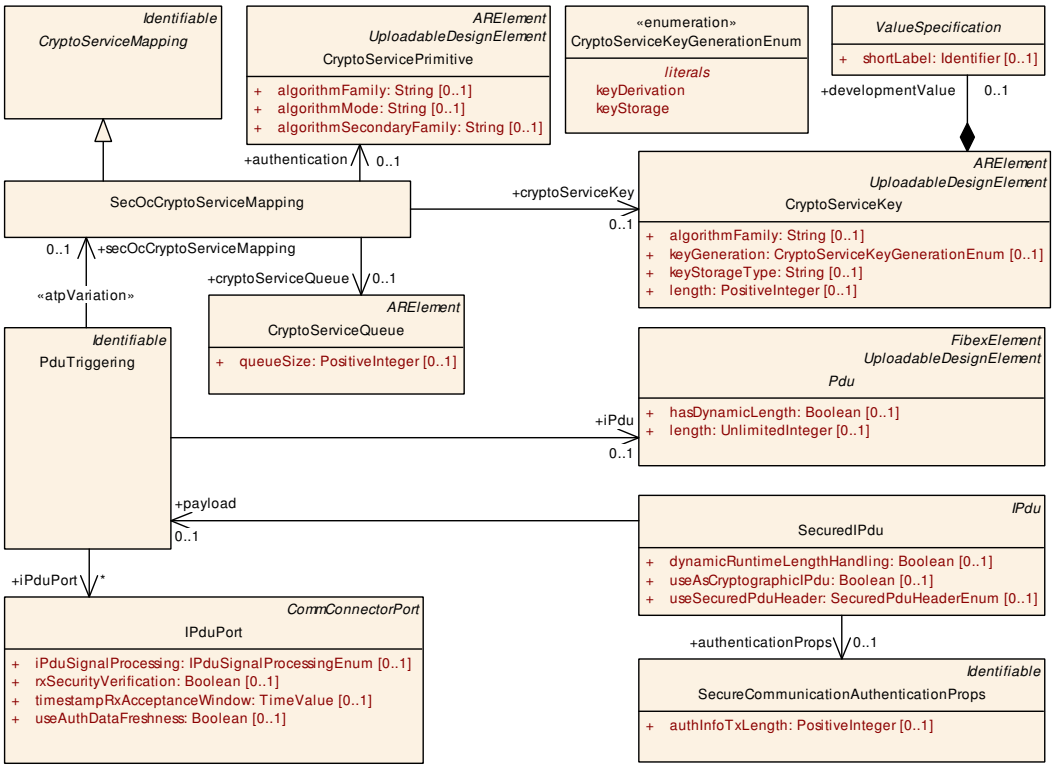

Figure 6.21: Modeling of crypto infrastructure for `SecuredIPdu`

Please note that the `developmentValue` most likely will be used in the form of a `TextValueSpecification`. However, the aggregation has still be modeled by means of using the abstract base class `ValueSpecification` to gain some headroom for future extension.

Class	**CryptoServiceKey**			
Note	This meta-class has the ability to represent a crypto key. **Tags:** atp.recommendedPackage=CryptoDevelopmentKeys			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
algorithmFamily	String	0..1	attr	This attribute represent the description of the family of the applicable crypto algorithm.
development Value	ValueSpecification	0..1	aggr	This aggregation represents the ability to assign a specific value to the crypto key as part of the system description. This value can then be taken for the development of the respective ECU. **Tags:** atp.Splitkey=developmentValue
keyGeneration	CryptoServiceKey GenerationEnum	0..1	attr	This attribute describes how a the specific cryptographic key is created.
keyStorageType	String	0..1	attr	This attribute describes where the enclosing cryptographic key shall be stored. AUTOSAR reserves specific values for this attributes but it is possible to insert custom values as well.
length	PositiveInteger	0..1	attr	This attribute describes the length of the cryptographic key in bits.

Table 6.51: CryptoServiceKey

[constr_9206] Existence of `CryptoServiceKey.length`

Imposition time: CP: `IT_SysDesc`

⌈For each `CryptoServiceKey`, the attribute `length` shall exist.⌋

Enumeration	**CryptoServiceKeyGenerationEnum**
Note	This enumeration shall be taken to express the handling of a crypto key in terms of whether it is obtained from e.g. a diagnostic tester or whether it is created by derivation from a master key.
Aggregated by	CryptoServiceKey.keyGeneration
Literal	***Description***
keyDerivation	This means that the crypto key is created by derivation from a master key. **Tags:** atp.EnumerationLiteralIndex=0
keyStorage	This means that the crypto key is obtained from an external entity, e.g. a diagnostic tester. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.52: CryptoServiceKeyGenerationEnum

[TPS_SYST_05025] Standardized values of `CryptoServicePrimitive.algorithmFamily` and `CryptoServiceKey.algorithmFamily`

⌈The following values of attributes `CryptoServicePrimitive.algorithmFamily` and `CryptoServiceKey.algorithmFamily` are standardized by AUTOSAR:

- **`CRYPTO_ALGOFAM_SHA1`**: SHA1 hash
- **`CRYPTO_ALGOFAM_SHA2_224`**: SHA2-224 hash
- **`CRYPTO_ALGOFAM_SHA2_256`**: SHA2-256 hash
- **`CRYPTO_ALGOFAM_SHA2_384`**: SHA2-384 hash
- **`CRYPTO_ALGOFAM_SHA2_512`**: SHA2-512 hash
- **`CRYPTO_ALGOFAM_SHA2_512_224`**: SHA2-512/224 hash
- **`CRYPTO_ALGOFAM_SHA2_512_256`**: SHA2-512/256 hash
- **`CRYPTO_ALGOFAM_SHA3_224`**: SHA3-224 hash
- **`CRYPTO_ALGOFAM_SHA3_256`**: SHA3-256 hash
- **`CRYPTO_ALGOFAM_SHA3_384`**: SHA3-384 hash
- **`CRYPTO_ALGOFAM_SHA3_512`**: SHA3-512 hash
- **`CRYPTO_ALGOFAM_SHAKE128`**: SHAKE128 hash
- **`CRYPTO_ALGOFAM_SHAKE256`**: SHAKE256 hash
- **`CRYPTO_ALGOFAM_RIPEMD160`**: RIPEMD hash
- **`CRYPTO_ALGOFAM_BLAKE_1_256`**: BLAKE-1-256 hash
- **`CRYPTO_ALGOFAM_BLAKE_1_512`**: BLAKE-1-512 hash
- **`CRYPTO_ALGOFAM_BLAKE_2s_256`**: BLAKE-2s-256 hash

- **CRYPTO_ALGOFAM_BLAKE_2s_512**: BLAKE-2s-512 hash
- **CRYPTO_ALGOFAM_3DES**: 3DES cipher
- **CRYPTO_ALGOFAM_AES**: AES cipher
- **CRYPTO_ALGOFAM_CHACHA**: ChaCha cipher
- **CRYPTO_ALGOFAM_RSA**: RSA cipher
- **CRYPTO_ALGOFAM_ED25519**: ED22518 elliptic curve
- **CRYPTO_ALGOFAM_BRAINPOOL**: Brainpool elliptic curve
- **CRYPTO_ALGOFAM_ECCNIST**: NIST ECC elliptic curves
- **CRYPTO_ALGOFAM_RNG**: Random Number Generator
- **CRYPTO_ALGOFAM_SIPHASH**: SipHash
- **CRYPTO_ALGOFAM_ECIES**: ECIES Cipher
- **CRYPTO_ALGOFAM_SM2**: SM2 elliptic curve algorithm
- **CRYPTO_ALGOFAM_EEA3**: Stream cipher based on ZUC
- **CRYPTO_ALGOFAM_SM3**: Hash algorithm based on ISO/IEC 10118-3:2018 Part 3: Dedicated hash-functions (SM3)
- **CRYPTO_ALGOFAM_EIA3**: Authentication based on ZUC

⌋

[TPS_SYST_05026] Relation of CryptoServicePrimitive.algorithmFamily to CryptoServiceKey.algorithmFamily ⌈The attribute CryptoServiceKey.algorithmFamily shall be taken to check with the value of CryptoServicePrimitive.algorithmFamily in order to make sure that the crypto key fits to the intended usage.⌋

[TPS_SYST_05027] Standardized values of CryptoServicePrimitive.algorithmMode ⌈The following values of attributes CryptoServicePrimitive.algorithmMode are standardized by AUTOSAR:

- **CRYPTO_ALGOMODE_ECB**: Blockmode - Electronic Code Book
- **CRYPTO_ALGOMODE_CBC**: Blockmode - Cipher Block Chaining
- **CRYPTO_ALGOMODE_CFB**: Blockmode - Cipher Feedback Mode
- **CRYPTO_ALGOMODE_OFB**: Blockmode - Output Feedback Mode
- **CRYPTO_ALGOMODE_CTR**: Blockmode - Counter Modex
- **CRYPTO_ALGOMODE_GCM**: Blockmode - Galois/Counter Mode

- **CRYPTO_ALGOMODE_XTS**: XOR-encryption-based tweaked-codebook mode with ciphertext stealing
- **CRYPTO_ALGOMODE_RSAES_OAEP**: RSA Optimal Asymmetric Encryption Padding
- **CRYPTO_ALGOMODE_RSAES_PKCS1_v1_5**: RSA encryption/decryption with PKCS#1 v1.5 padding
- **CRYPTO_ALGOMODE_RSASSA_PSS**: RSA Probabilistic Signature
- **CRYPTO_ALGOMODE_RSASSA_PKCS1_v1_5**: RSA signature with PKCS#1 v1.5
- **CRYPTO_ALGOMODE_8ROUNDS**: 8 rounds (e.g. ChaCha8)
- **CRYPTO_ALGOMODE_12ROUNDS**: 12 rounds (e.g. ChaCha12)
- **CRYPTO_ALGOMODE_20ROUNDS**: 20 rounds (e.g. ChaCha20)
- **CRYPTO_ALGOMODE_HMAC**: Hashed-based MAC
- **CRYPTO_ALGOMODE_CMAC**: Cipher-based MAC
- **CRYPTO_ALGOMODE_GMAC**: Galois MAC
- **CRYPTO_ALGOMODE_CTRDRBG**: Counter-based Deterministic Random Bit Generator
- **CRYPTO_ALGOMODE_SIPHASH_2_4**: Siphash-2-4
- **CRYPTO_ALGOMODE_SIPHASH_4_8**: Siphash-4-8
- **CRYPTO_ALGOMODE_AESKEYWRAP**: AES Key Wrap (RFC 3394)
- **CRYPTO_ALGOMODE_GCM_SIV**: : Blockmode - Galois/Counter Mode with Synthetic Initialization Vector

⌋

Please note that it is positively supported to define custom values for attributes `CryptoServicePrimitive.algorithmFamily` and `CryptoServicePrimitive.algorithmMode` provided that the custom values are guaranteed to not clash with future extension of the AUTOSAR standard. For example, this could be achieved by using a prefix or suffix that is specific to the organization that defines the custom value.

[TPS_SYST_05028] Semantics of `CryptoServiceKey.keyStorageType`
⌈Attribute `CryptoServiceKey.keyStorageType` describes where the actual key shall be stored on the ECU. This attribute has been deliberately modeled as a `String` to allow future (and custom) extensions of the range of possible values.

AUTOSAR reserves the following values for this attribute:

- **SHE**

- **RAM**
- **HSM**
- **NVM**

⌋

Please note that custom values for attribute CryptoServiceKey.keyStorageType are supported as long as the actual values are defined in a way such that a possible clash with a later standardization by AUTOSAR becomes impossible.

The best way to achieve this is to use the company name (e.g. as a prefix) within the custom value of CryptoServiceKey.keyStorageType.

Class	**CryptoServiceQueue**			
Note	This meta-class has the ability to represent a crypto queue. **Tags:** atp.recommendedPackage=CryptoServiceQueues			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
queueSize	PositiveInteger	0..1	attr	Defines the queue size of the CryptoServiceQueue.

Table 6.53: CryptoServiceQueue

[constr_5058] Value range for CryptoServiceQueue.queueSize

Imposition time: CP: IT_SysDesc

⌈If the CryptoServiceQueue.queueSize is defined it shall have a value which is equal or greater than 1.⌋

6.3.3 J1939ProtectedIPdu

[TPS_SYST_02416] Semantics of J1939ProtectedIPdu

Status: DRAFT

⌈The meta-class J1939ProtectedIPdu is used to describe the Safety Data Message (SDM) of SAE J1939-76 that is paired with the Safety Header Message containing the E2E protection data from the Pdu referenced as payload via PduTriggering.

In other words the J1939ProtectedIPdu describes the Safety Data Message (SDM) and references a PduTriggering that in turn references an ISignalIPdu that contains all necessary data for SHM (Safety Header Message) and SDM (Safety Data Message) but only contains the direct representation of the SDM (Safety Data Message) while the layout of the SHM is handled by J1939Fscp (SAE J1939 Functional Safety Communication Protocol). The SHM (Safety Header Message) combines the E2E information (see [TPS_SYST_02412]) with CanId parts of the SDM (Safety Data Message) (not explictly modeled as ISignals of the ISignalGroup).⌋

Note: The Safety Header Message (SHM) has a fixed layout and CAN identifier and is not modeled as a Pdu, but is derived during Ecu configuration.

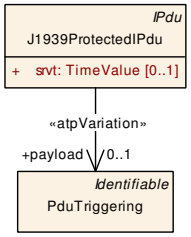

Figure 6.22: J1939ProtectedIPdu

[TPS_SYST_02417] Content of ISignalGroup located in J1939ProtectedIPdu.payload

Status: DRAFT

⌈The ISignalGroup mapped by ISignalToIPduMapping into an ISignalIPdu referenced by J1939ProtectedIPdu.payload via PduTriggering shall define ISignals that represent the E2E header as defined in [TPS_SYST_02068].⌋

[constr_9343] Allowed J1939ProtectedIPdu.payload reference target

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈A J1939ProtectedIPdu is only allowed to reference a PduTriggering with the payload reference that in turn references an ISignalIPdu in the role iPdu to which an ISignalGroup is mapped that aggregates EndToEndTransformationISignalProps in the role transformationISignalProps which references an EndToEndTransformationDescription with profileName PROFILE_76.⌋

Class	**J1939ProtectedIPdu**			
Note	Represents the SDM (Safety Data Message) that contains the actual payload (signals) of the E2E protected J1939 message. **Tags:** atp.Status=draft atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
payload	PduTriggering	0..1	ref	References the ISignalIPdu that represents the SDG (Safety Data Group) that contains both the payload (signals) and the E2E protection data of the protected J1939 message. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=payload, payload.pduTriggering, payload.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild

▽

△

Class	**J1939ProtectedIPdu**			
srvt	TimeValue	0..1	attr	Maximum time between SHM (Safety Header Message) and SDM (Safety Data Message) of one SDG (Safety Data Group) **Tags:** atp.Status=draft

Table 6.54: J1939ProtectedIPdu

The following Figure 6.23 shows in an example how a `J1939ProtectedIPdu` is modeled. The `J1939ProtectedIPdu` is transmitted on the network and is therefore mapped into a `CanFrame`. Both, the `J1939ProtectedIPdu` and the `J1939ProtectedIPdu.payload ISignalIPdu` have a `PduTriggering` on the same CAN `PhysicalChannel`. The `payload ISignalIPdu` contains an `ISignalGroup` with aggregated `EndToEndTransformationISignalProps`.

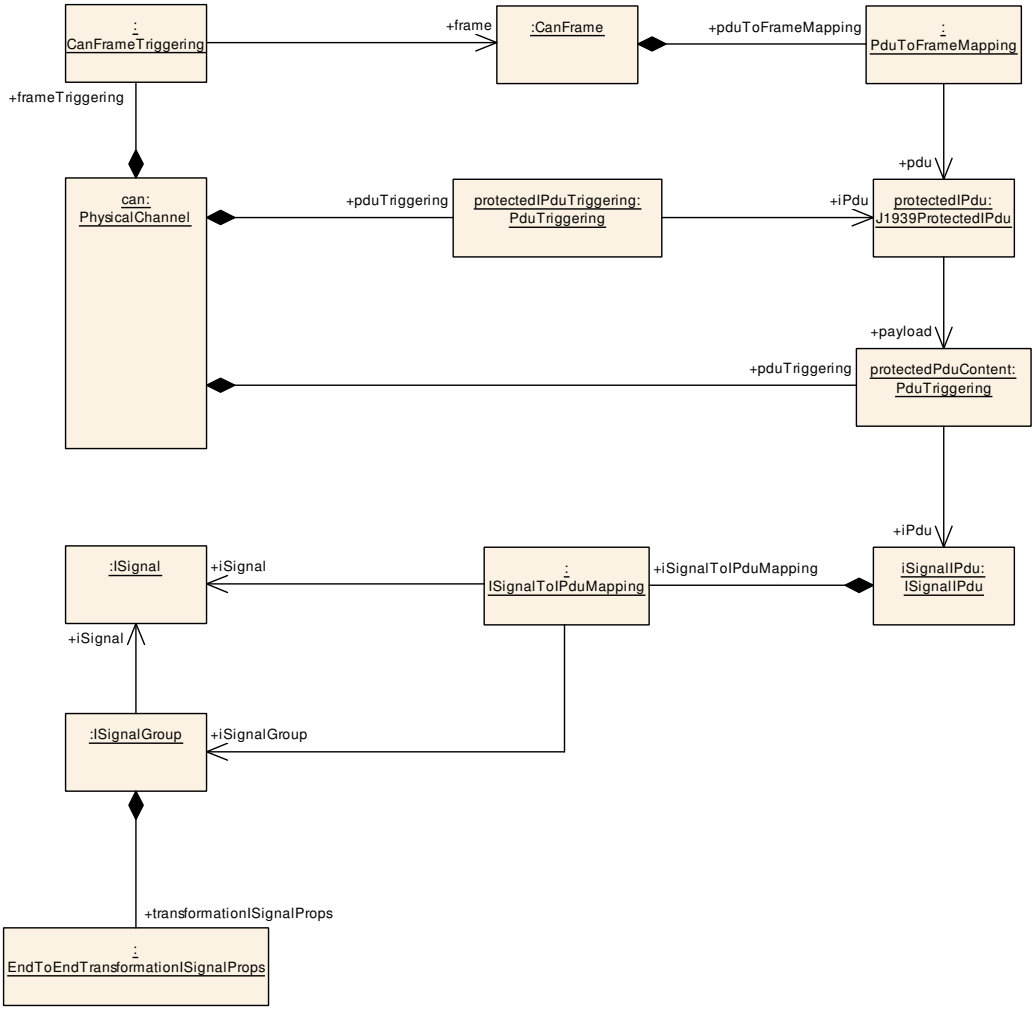

Figure 6.23: Example for modeling of J1939ProtectedIPdu

6.3.4 GeneralPurposeConnection

In some cases it is important to describe a relation between different PduTriggerings that are defined on the same PhysicalChannel, e.g. to create a link between a Rx-Pdu and a Tx-Pdu. The GeneralPurposeConnection meta-class is able to reference a number of PduTriggerings and thereby to set the referenced PduTriggerings into a relationship that is defined by the GeneralPurposeConnection.

[TPS_SYST_02170] category of the GeneralPurposeConnection ⌈The category of the GeneralPurposeConnection is used to define the purpose of the relationship between the referenced PduTriggerings.⌋

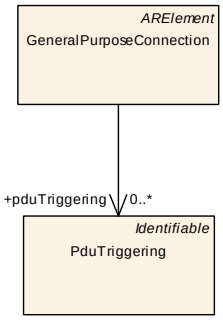

Figure 6.24: GeneralPurposeConnection

Class	**GeneralPurposeConnection**			
Note	This meta-class allows to describe the relationship between several PduTriggerings that are defined on the same PhysicalChannel, e.g. to create a link between Rx and Tx Pdu that are used for request/ response. **Tags:** atp.recommendedPackage=GeneralPurposeConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pduTriggering	PduTriggering	*	ref	Reference to PduTriggerings that are connected to each other by a GeneralPurposeConnection. **Tags:** atp.Splitkey=pduTriggering

Table 6.55: GeneralPurposeConnection

[constr_3384] PduTriggerings referenced by GeneralPurposeConnection shall be defined on the same PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈The PduTriggerings that are referenced by the GeneralPurposeConnection in the role pduTriggering shall be defined on the same PhysicalChannel.⌋

[constr_3383] Standardized values for the attribute `category` of meta-class `GeneralPurposeConnection`

Imposition time: CP: `IT_SysDesc`

⌈The following values of the attribute `category` of meta-class `GeneralPurposeConnection` are reserved by the AUTOSAR standard:

- XcpChannel

⌋

The XcpChannel creates a link between one Tx-Pdu and one Rx-Pdu that are used for request/response from one master.

[constr_3385] XcpChannel is allowed to reference exactly two `PduTriggering`s

Imposition time: CP: `IT_SysDesc`

⌈In case that the `category` of meta-class `GeneralPurposeConnection` is set to the value *XcpChannel* the `GeneralPurposeConnection` is allowed to reference exactly two `PduTriggering`s in the role `pduTriggering`.⌋

[constr_3386] XcpChannel is only allowed to reference `PduTriggering`s of `GeneralPurposeIPdu`s with category XCP

Imposition time: CP: `IT_SysDesc`

⌈In case that the `category` of meta-class `GeneralPurposeConnection` is set to the value *XcpChannel* the `GeneralPurposeConnection` is allowed to reference `PduTriggering`s of `GeneralPurposeIPdu`s with category XCP.⌋

6.4 IPdu Timing

AUTOSAR COM allows configuring statically two different transmission modes for each IPdu (True and False). `TransmissionModeDeclaration` uses a transmission mode selector, calculated from a number of individual `TransmissionModeCondition`s or `ModeDrivenTransmissionModeCondition`s to decide which of the two modes is selected. It is possible to switch between the transmission modes during runtime.

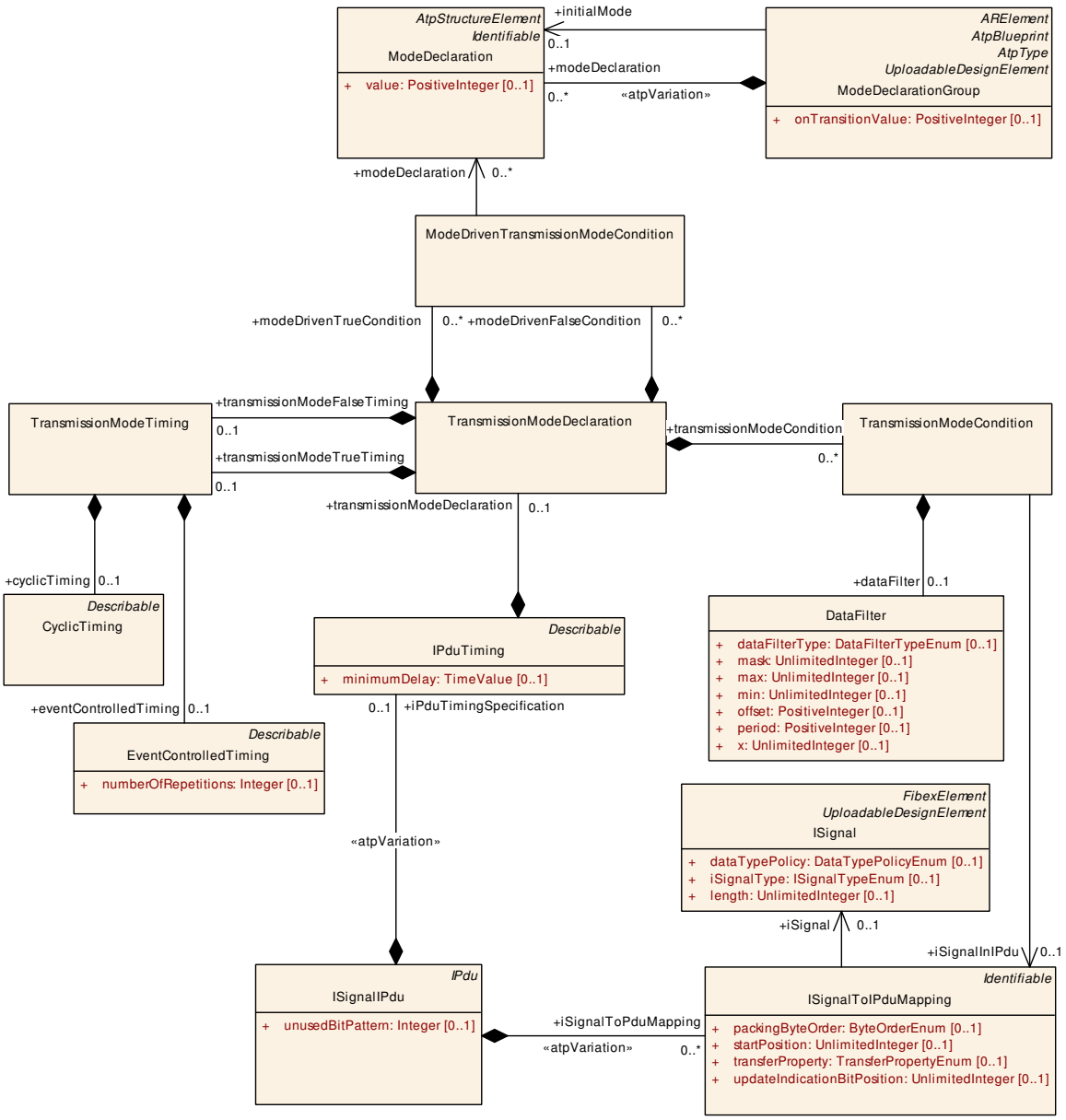

Figure 6.25: IPdu Timing

[TPS_SYST_01075] **Signal content evaluation via `TransmissionModeCondition`** ⌈The signal content can be evaluated as the transmission mode selector via the `TransmissionModeCondition`s.⌋

[TPS_SYST_01076] Mode evaluation via `modeDrivenTrueCondition` or `modeDrivenFalseCondition` ⌈Mode conditions can be evaluated as the transmission mode selector via the `modeDrivenTrueCondition`s or `modeDrivenFalseCondition`s.⌋

[constr_3045] Signal content evaluation vs. Mode evaluation

Imposition time: CP: IT_SysDesc

⌈The mode evaluation and the signal content evaluation shall not be used in the same IPdu. A mix of these two types is not allowed.⌋

To use the signal content evaluation a TransmissionModeCondition can be attached to each signal within an IPdu. Each TransmissionModeCondition contains a reference to a signal and to an assigned filter. The filter condition is used for the selection of the transmission mode. If at least one condition in the signal content evaluation is true, Transmission Mode "TRUE" shall be used for this IPdu. In all other cases, the Transmission Mode "FALSE" shall be used. More details can be found in the COM Specification [25].

[constr_3046] Consistency of TransmissionModeCondition.iSignalInIPdu

Imposition time: CP: IT_SysDesc

⌈The ISignalToIPduMapping referenced by the TransmissionModeCondition in the role iSignalInIPdu shall belong to the same ISignalIPdu as the TransmissionModeCondition.⌋

It is a use-case to define an unconditional TransmissionModeTiming for an ISignalIPdu, meaning that this ISignalIPdu will always have the same transmission behavior. This can be defined by having a TransmissionModeDeclaration without any TransmissionModeDeclaration.transmissionModeCondition or all of the aggregated TransmissionModeConditions have the TransmissionModeCondition.dataFilter set to DataFilter.dataFilterType = DataFilterTypeEnum.always. Also no TransmissionModeDeclaration.modeDrivenTrueCondition or TransmissionModeDeclaration.modeDrivenFalseCondition are to be defined in this use-case. In this setup only one TransmissionModeTiming is applicable (as there are no conditions defined where a distinction might be required) and this one TransmissionModeTiming shall be aggregated at the TransmissionModeDeclaration.transmissionModeTrueTiming.

[constr_9354] Definition of unconditional IPdu Timing

Imposition time: CP: IT_SysDesc

⌈If a TransmissionModeDeclaration has

- no TransmissionModeDeclaration.transmissionModeCondition or all of the aggregated TransmissionModeConditions have the TransmissionModeCondition.dataFilter set to DataFilter.dataFilterType = DataFilterTypeEnum.always, and
- no TransmissionModeDeclaration.modeDrivenTrueCondition defined, and
- no TransmissionModeDeclaration.modeDrivenFalseCondition defined,

then only one TransmissionModeTiming in the role TransmissionModeDeclaration.transmissionModeTrueTiming shall be defined.⌋

In the mode driven evaluation ModeDeclarations are evaluated. The transmissionModeFalseTiming is activated if all defined modeDrivenFalseConditions evaluate to true and the transmissionModeTrueTiming is activated if all defined modeDrivenTrueConditions evaluate to true. Each condition that is defined by ModeDrivenTransmissionModeCondition evaluates to true if one of the referenced ModeDeclarations is active.

The TransmissionModeDeclaration element aggregates the TransmissionModeTiming in two different roles: transmissionModeTrueTiming and transmissionModeFalseTiming. The available COM Transmission Mode Timings can be described by the CyclicTiming and EventControlledTiming elements (see [TPS_SYST_01077]) that are aggregated by the TransmissionModeTiming class.

[TPS_SYST_01077] Mapping of Com Transmission Modes to System Template elements ⌈

COM Transmission Modes	*Description*	*realization in System Template*
Periodic	Transmissions occur indefinitely with a fixed time period between them	CyclicTiming
Direct/n-times	Event driven transmission with n-1 repetitions	EventControlledTiming
Mixed	Periodic transmission with direct/n-times transmissions in between	EventControlledTiming and CyclicTiming
None	No transmission	no timing assigned

⌋

Class	**TransmissionModeDeclaration**			
Note	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES (True and False) for each I-PDU. As TransmissionMode selector the signal content can be evaluated via transmissionModeCondition (implemented directly in the COM module) or mode conditions can be defined with the modeDrivenTrue Condition or modeDrivenFalseCondition (evaluated by BswM and invoking Com_SwitchIpduTxMode COM API). If modeDrivenTrueCondition and modeDrivenFalseCondition are defined they shall never evaluate to true both at the same time. The mixing of Transmission Mode Switch via API and signal value is not allowed.			
Base	*ARObject*			
Aggregated by	IPduTiming.transmissionModeDeclaration			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**TransmissionModeDeclaration**			
modeDriven FalseCondition	ModeDriven TransmissionMode Condition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDriven FalseConditions evaluate to true (AND associated) the transmissionModeFalseTiming shall be activated. mode DrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time. **Tags:** atp.Splitkey=modeDrivenFalseCondition
modeDriven TrueCondition	ModeDriven TransmissionMode Condition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDriven TrueConditions evaluate to true (AND associated) the transmissionModeTrueTiming shall be activated. mode DrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time. **Tags:** atp.Splitkey=modeDrivenTrueCondition
transmission ModeCondition	TransmissionMode Condition	*	aggr	The Transmission Mode Selector evaluates the conditions for a subset of signals and decides which transmission mode should be used. In case only one transmission mode is used there is no need for the "TransmissionMode Condition" and its sub-structure. In case the transmission mode shall be switched using the COM-API "Com_Switch IpduTxMode" there is no need for the "TransmissionMode Condition" and its sub-structure. **Tags:** atp.Splitkey=transmissionModeCondition
transmission ModeFalse Timing	TransmissionMode Timing	0..1	aggr	Timing Specification if the COM Transmission Mode is false. The Transmission Mode Selector is defined to be false, if all Conditions evaluate to false. **Tags:** atp.Splitkey=transmissionModeFalseTiming
transmission ModeTrue Timing	TransmissionMode Timing	0..1	aggr	Timing Specification if the COM Transmission Mode is true. The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true. **Tags:** atp.Splitkey=transmissionModeTrueTiming

Table 6.56: TransmissionModeDeclaration

Class	**TransmissionModeCondition**			
Note	Possibility to attach a condition to each signal within an I-PDU. If at least one condition evaluates to true, TRANSMISSION MODE True shall be used for this I-Pdu. In all other cases, the TRANSMISSION MODE FALSE shall be used.			
Base	*ARObject*			
Aggregated by	TransmissionModeDeclaration.transmissionModeCondition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilter	DataFilter	0..1	aggr	Possibilities to define conditions **Tags:** atp.Splitkey=dataFilter
iSignalInIPdu	ISignalToIPduMapping	0..1	ref	Reference to a signal to which a condition is attached. **Tags:** atp.Splitkey=iSignalInIPdu

Table 6.57: TransmissionModeCondition

[constr_9185] Existence of `TransmissionModeCondition.dataFilter`

Imposition time: CP: `IT_SysDesc`

⌈For each `TransmissionModeCondition`, the aggregation of `DataFilter` in the role `dataFilter` shall exist.⌋

[constr_9186] Existence of TransmissionModeCondition.iSignalInIPdu

Imposition time: CP: IT_SysDesc

⌈For each TransmissionModeCondition, the reference to ISignalToIPduMapping in the role iSignalInIPdu shall exist.⌋

Class	ModeDrivenTransmissionModeCondition			
Note	The condition defined by this class evaluates to true if one of the referenced modeDeclarations (OR associated) is active. All referenced modeDeclarations shall be from the same ModeDeclarationGroup. The condition is used to define which TransmissionMode shall be activated using Com_SwitchIpduTx Mode.			
Base	*ARObject*			
Aggregated by	TransmissionModeDeclaration.modeDrivenFalseCondition, TransmissionModeDeclaration.modeDrivenTrueCondition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode Declaration	ModeDeclaration	*	ref	Reference to one modeDeclaration which is OR associated in the context of the ModeDrivenTransmissionModeCondition. **Tags:** atp.Splitkey=modeDeclaration

Table 6.58: ModeDrivenTransmissionModeCondition

[constr_9187] Existence of ModeDrivenTransmissionModeCondition.modeDeclaration

Imposition time: CP: IT_SysDesc

⌈For each ModeDrivenTransmissionModeCondition, the reference to ModeDeclaration in the role modeDeclaration shall exist.⌋

The ModeDeclaration and the ModeDeclarationGroup is described in more detail in the Software Component Template Specification [4].

Class	TransmissionModeTiming			
Note	If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)			
Base	*ARObject*			
Aggregated by	TransmissionModeDeclaration.transmissionModeFalseTiming, TransmissionModeDeclaration.transmissionModeTrueTiming			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cyclicTiming	CyclicTiming	0..1	aggr	Periodic Transmission Mode. **Tags:** atp.Splitkey=cyclicTiming
eventControlled Timing	EventControlledTiming	0..1	aggr	Direct Transmission Mode. **Tags:** atp.Splitkey=eventControlledTiming

Table 6.59: TransmissionModeTiming

6.4.1 Data Filter configuration

Data Filters are used on sender side to configure Transmission Mode Conditions (TMC). On receiver side Data Filters can be used as filtering mechanisms for signals (see ISignalPort element). More details about the usage of DataFilters can be found in the Software Component Template Specification [4].

[TPS_SYST_02013] Usage of dataFilters on GroupSignals on receiver side
⌈If the dataFilter of one GroupSignal evaluates to false the whole ISignal-Group in which the GroupSignal is contained shall be discarded.⌋

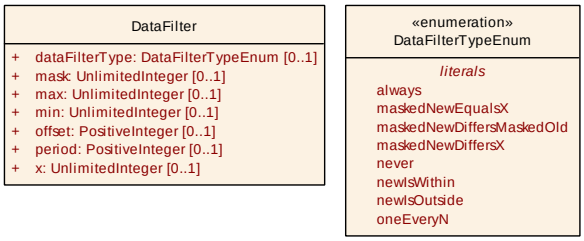

Figure 6.26: Data Filter

Class	**DataFilter**			
Note	Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.			
Base	*ARObject*			
Aggregated by	ISignalPort.dataFilter, NonqueuedReceiverComSpec.filter, NonqueuedSenderComSpec.dataFilter, SignalBasedEventElementToISignalTriggeringMapping.filter, SignalBasedFieldToISignalTriggering Mapping.filter, SignalServiceTranslationElementProps.filter, TransmissionModeCondition.dataFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataFilterType	DataFilterTypeEnum	0..1	attr	This attribute specifies the type of the filter.
mask	UnlimitedInteger	0..1	attr	Mask for old and new value.
max	UnlimitedInteger	0..1	attr	Value to specify the upper boundary
min	UnlimitedInteger	0..1	attr	Value to specify the lower boundary
offset	PositiveInteger	0..1	attr	Specifies the initial number of messages to occur before the first message is passed
period	PositiveInteger	0..1	attr	Specifies number of messages to occur before the message is passed again
x	UnlimitedInteger	0..1	attr	Value to compare with

Table 6.60: DataFilter

Enumeration	**DataFilterTypeEnum**
Note	This enum specifies the supported DataFilterTypes.
Aggregated by	DataFilter.dataFilterType
Literal	***Description***
always	No filtering is performed so that the message always passes. **Tags:** atp.EnumerationLiteralIndex=0

▽

△

Enumeration	**DataFilterTypeEnum**
maskedNewDiffersMaskedOld	Pass messages where the masked value has changed. (new_value&mask) !=(old_value&mask) new_value: current value of the message old_value: last value of the message (initialized with the initial value of the message, updated with new_value if the new message value is not filtered out) **Tags:** atp.EnumerationLiteralIndex=1
maskedNewDiffersX	Pass messages whose masked value is not equal to a specific value x (new_value&mask) != x new_value: current value of the message **Tags:** atp.EnumerationLiteralIndex=2
maskedNewEqualsX	Pass messages whose masked value is equal to a specific value x (new_value&mask) == x new_value: current value of the message **Tags:** atp.EnumerationLiteralIndex=3
never	The filter removes all messages. **Tags:** atp.EnumerationLiteralIndex=4
newIsOutside	Pass a message if its value is outside a predefined boundary. (min > new_value) OR (new_value > max) **Tags:** atp.EnumerationLiteralIndex=5
newIsWithin	Pass a message if its value is within a predefined boundary. min <= new_value <= max **Tags:** atp.EnumerationLiteralIndex=6
oneEveryN	Pass a message once every N message occurrences. Algorithm: occurrence % period == offset Start: occurrence = 0. Each time the message is received or transmitted, occurrence is incremented by 1 after filtering. Length of occurrence is 8 bit (minimum). **Tags:** atp.EnumerationLiteralIndex=7

Table 6.61: DataFilterTypeEnum

6.4.2 Cyclic Timing

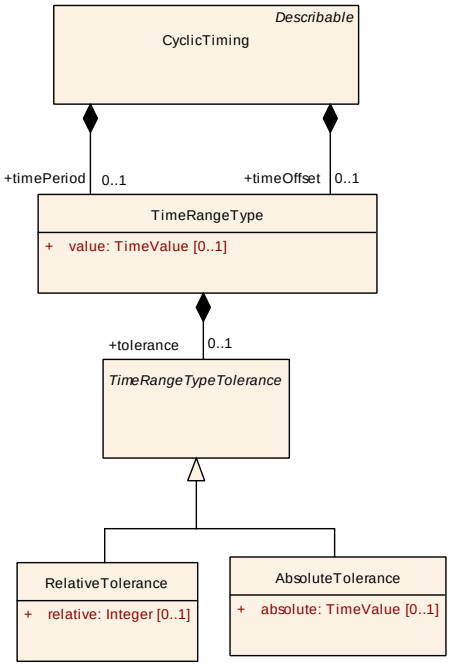

Figure 6.27: Cyclic Timing

Class	***TimeRangeTypeTolerance*** (abstract)			
Note	Maximum allowable deviation			
Base	*ARObject*			
Subclasses	AbsoluteTolerance, RelativeTolerance			
Aggregated by	TimeRangeType.tolerance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.62: TimeRangeTypeTolerance

Class	**CyclicTiming**			
Note	Specification of a cyclic sending behavior.			
Base	*ARObject*, *Describable*			
Aggregated by	TransmissionModeTiming.cyclicTiming			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
timeOffset	TimeRangeType	0..1	aggr	This attribute specifies the time until first transmission of this I-PDU. This attribute defines the time between Com_IpduGroupStart and the first transmission of the cyclic part of this transmission request for this I-PDU. **Tags:** atp.Splitkey=timeOffset
timePeriod	TimeRangeType	0..1	aggr	Period of the repetition of cyclic transmissions. **Tags:** atp.Splitkey=timePeriod

Table 6.63: CyclicTiming

[constr_9188] Existence of ModeDrivenTransmissionModeCondition.timePeriod

Imposition time: CP: IT_SysDesc

⌈For each CyclicTiming, the aggregation of TimeRangeType in the role timePeriod shall exist.⌋

6.4.3 EventControlled Timing

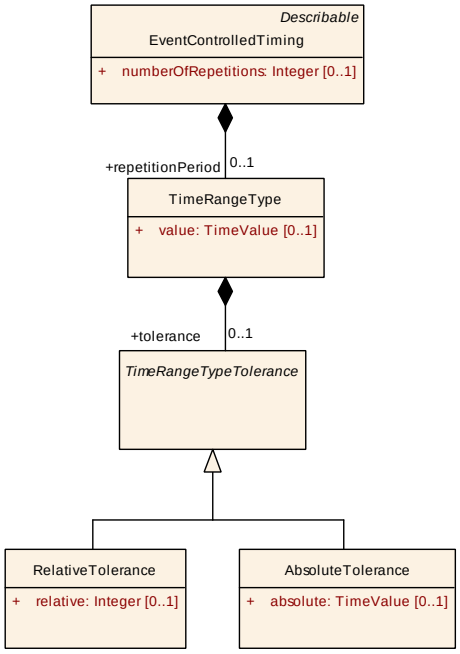

Figure 6.28: EventControlled Timing

Class	**EventControlledTiming**			
Note	Specification of a event driven sending behavior. The PDU is sent n (numberOfRepeat + 1) times separated by the repetitionPeriod. If numberOfRepeats = 0, then the Pdu is sent just once.			
Base	*ARObject*, *Describable*			
Aggregated by	TransmissionModeTiming.eventControlledTiming			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf Repetitions	Integer	0..1	attr	Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.
repetitionPeriod	TimeRangeType	0..1	aggr	The repetitionPeriod specifies the time in seconds that elapses before the pdu can be sent the next time (Minimum repeat gap between two pdus). The repetition Period is optional in case that no repetitions are configured. **Tags:** atp.Splitkey=repetitionPeriod

Table 6.64: EventControlledTiming

[constr_9189] Existence of EventControlledTiming.numberOfRepetitions

Imposition time: CP: IT_SysDesc

⌈For each EventControlledTiming, the attribute numberOfRepetitions shall exist.⌋

Class	TimeRangeType			
Note	The timeRange can be specified with the value attribute. Optionally a tolerance can be defined.			
Base	ARObject			
Aggregated by	CyclicTiming.timeOffset, CyclicTiming.timePeriod, EventControlledTiming.repetitionPeriod			
Attribute	Type	Mult.	Kind	Note
tolerance	TimeRangeType Tolerance	0..1	aggr	Optional specification of a tolerance. **Tags:** atp.Splitkey=tolerance
value	TimeValue	0..1	attr	Average value of a date (in seconds)

Table 6.65: TimeRangeType

[constr_9190] Existence of TimeRangeType.value

Imposition time: CP: IT_SysDesc

⌈For each TimeRangeType, the attribute value shall exist.⌋

Class	RelativeTolerance			
Note	Maximum allowable deviation			
Base	ARObject, TimeRangeTypeTolerance			
Aggregated by	TimeRangeType.tolerance			
Attribute	Type	Mult.	Kind	Note
relative	Integer	0..1	attr	Maximum allowable deviation in percent (percent of the corresponding TimeValue).

Table 6.66: RelativeTolerance

[constr_9191] Existence of RelativeTolerance.relative

Imposition time: CP: IT_SysDesc

⌈For each RelativeTolerance, the attribute relative shall exist.⌋

Class	AbsoluteTolerance			
Note	Maximum allowable deviation			
Base	ARObject, TimeRangeTypeTolerance			
Aggregated by	TimeRangeType.tolerance			
Attribute	Type	Mult.	Kind	Note
absolute	TimeValue	0..1	attr	Maximum allowable deviation in duration (in seconds)

Table 6.67: AbsoluteTolerance

[constr_9192] Existence of AbsoluteTolerance.absolute

Imposition time: CP: IT_SysDesc

⌈For each AbsoluteTolerance, the attribute absolute shall exist.⌋

6.4.4 Configuration of a trigger for COM_TriggerIPduSend API call

In the AUTOSAR BswM module a `BswMAction` with `BswMTriggerIPduSend` may be defined. The COM API `Com_TriggerIPDUSend` is called when this action is configured. By the call of `Com_TriggerIPDUSend` an IPdu with a given ID is triggered for transmission.

With such a configuration a single transmission of an IPdu can be configured that is independent of the configured COM transmission modes, e.g. in case of a vehicle mode change.

In a System Description the usage of the `Com_TriggerIPDUSend` API is defined with the `TriggerIPduSendCondition` that is aggregated by the `PduTriggering` in the role `triggerIPduSendCondition`. The `TriggerIPduSendCondition` defines the trigger for the `Com_TriggerIPDUSend` API call.

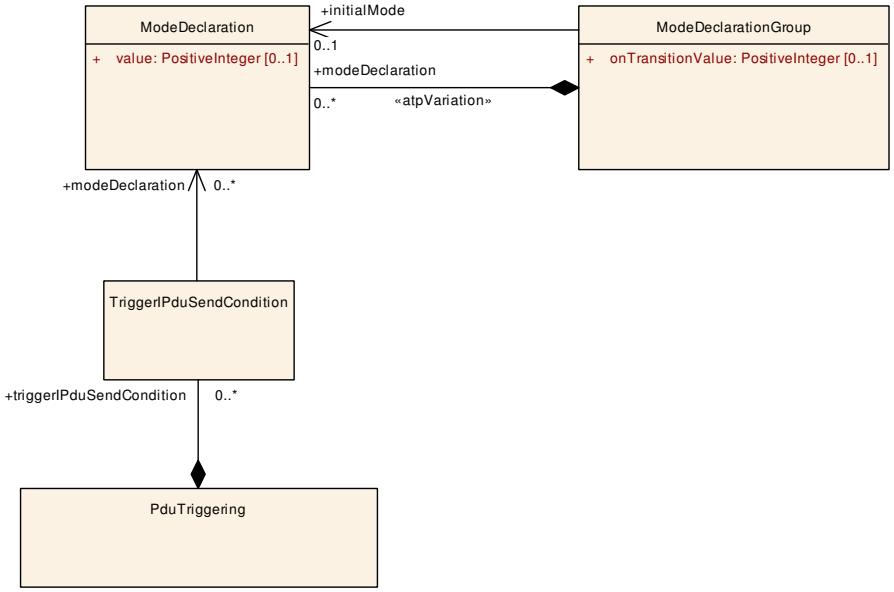

Figure 6.29: TriggerIPduSendCondition

Class	**TriggerIPduSendCondition**			
Note	The condition defined by this class evaluates to true if one of the referenced modeDeclarations (OR associated) is active. The condition is used to define when the Pdu is triggered with the Com_Trigger IPDUSend API call.			
Base	*ARObject*			
Aggregated by	PduTriggering.triggerIPduSendCondition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mode Declaration	ModeDeclaration	*	ref	Reference to one modeDeclaration which is OR associated in the context of the TriggerIPduSend Condition. **Tags:** atp.Splitkey=modeDeclaration

Table 6.68: TriggerIPduSendCondition

[constr_9193] Existence of `TriggerIPduSendCondition.modeDeclaration`

Imposition time: CP: `IT_SysDesc`

⌈For each `TriggerIPduSendCondition`, the reference to `ModeDeclaration` in role `modeDeclaration` shall exist.⌋

Only if all defined `TriggerIPduSendCondition`s evaluate to true (AND associated) the `Com_TriggerIPDUSend` API shall be called.

[constr_3211] `PduTriggering`s with `triggerIPduSendCondition`

Imposition time: CP: `IT_SysDesc`

⌈Only `PduTriggering`s with references to `ISignalIPdu`s are allowed to contain a `triggerIPduSendCondition`.⌋

Please note that OR Conditions defined by the `TriggerIPduSendCondition.modeDeclaration` are evaluated first. The AND Conditions defined by `PduTriggering.triggerIPduSendCondition`s are evaluated after the OR Conditions.

6.5 I-Pdu Multiplexer

Multiplexing is used to transport varying Com IPdus at the same position in a single multiplexed IPdu. A multiplexed IPdu consists of a dynamic part, a selector field and an optional static part. According to the value of the selector field the dynamic part can have a different layout.

[TPS_SYST_01078] Dynamic Part of a `MultiplexedIPdu` ⌈For each alternative of a `MultiplexedIPdu` there is exactly one Com IPdu that is transmitted in the dynamic part.⌋

[TPS_SYST_01079] Static Part of a `MultiplexedIPdu` ⌈The static part of a `MultiplexedIPdu` is the same regardless of the selector field and consists of exactly one Com IPdu.⌋

The `MultiplexedIPdu` element contains attributes that describe the position and the length of a selector within an `IPdu`. A selector is a bitfield of certain length, by the value of which the corresponding data region of the dynamic part shall be interpreted dynamically, i.e. at run-time.

[constr_3007] `selectorFieldCode`s for dynamic part alternatives

Imposition time: CP: `IT_SysDesc`

⌈The `selectorFieldCode`s for the dynamic part alternatives within one `MultiplexedIPdu` shall differ from each other.⌋

[constr_3097] Overlapping of segments of one `MultiplexedIPdu` is not allowed

Imposition time: CP: `IT_SysDesc`

⌈The segments defined by the `SegmentPosition` elements of one and the same `MultiplexedIPdu` - aggregated via `StaticPart` and `DynamicPart` - shall not overlap.⌋

[constr_3098] Defined segments of one `MultiplexedIPdu` shall not exceed the length of the `MultiplexedIPdu`

Imposition time: CP: `IT_SysDesc`

⌈The segments defined by the `SegmentPosition` elements of one and the same `MultiplexedIPdu` - aggregated via `StaticPart` and `DynamicPart` - shall not exceed the length of the `MultiplexedIPdu`.⌋

[constr_3099] Defined segments in a `DynamicPart` shall not exceed the length of any `DynamicPartAlternative.iPdu`

Imposition time: CP: `IT_SysDesc`

⌈The segments defined by the `SegmentPosition` elements aggregated in the `DynamicPart` of a `MultiplexedIPdu` shall not exceed the length of any `DynamicPartAlternative.iPdu`.⌋

[constr_3100] Defined segments in a `StaticPart` shall not exceed the length of the `StaticPart.iPdu`

Imposition time: CP: `IT_SysDesc`

⌈The segments defined by the `SegmentPosition` elements aggregated in the `StaticPart` of a `MultiplexedIPdu` shall not exceed the length of the `StaticPart.iPdu`⌋

[constr_3101] Signal representation of selector field for `DynamicPartAlternative`

Imposition time: CP: `IT_SysDesc`

⌈Every `ISignalIPdu` that is referenced by the `DynamicPartAlternative` shall contain an `ISignal` that represents the selector field. The selector field signal shall be located at the position that is described by the `selectorFieldLength` and `selectorFieldStartPosition`.⌋

[constr_5254] Value range of `MultiplexedIPdu.selectorFieldLength`

Imposition time: CP: `IT_SysDesc`

⌈The value of `MultiplexedIPdu.selectorFieldLength` shall be in the range of 1..16 Bits.⌋

It is assumed by the IPduM that the value of the `ISignal` representing the selector field value matches the value defined in `DynamicPartAlternative.selectorFieldCode`. The IPduM does not set or modify the selector field value. Therefore it is essential that the System Description is defined in a consistent way in order to get valid selector field value configurations.

There are two approaches how the selector field value is configured:

- static initialization of the `ISignal` representing the selector field value.
- giving applications the possibility to write the selector field value.

If the selector field value is initialized using the `ISignal` representing the selector field value then the consistency is defined in [TPS_SYST_02351]. In this case there is no change to the value of the `ISignal` representing the selector field value during runtime. Each `DynamicPartAlternative` has a corresponding `IPdu` with a correctly defined selector field value. Regardless which `DynamicPartAlternative` is triggered by COM, it always has the correct selector field value.

[TPS_SYST_02351] Selector field signal initial values in case no application writing the selector field signal ⌈If

- the `ISignal` representing the selector field is referenced by an `ISignalTriggering` and that `ISignalTriggering` refers to an `ISignalPort` where the `communicationDirection` is set to `out` and
- the `ISignal` representing the selector field is referring to a `SystemSignal` and that `SystemSignal` is neither

- referenced by a SenderReceiverToSignalMapping in the role systemSignal nor
- part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping,

then this ISignal representing the selector field shall have an initValue defined which corresponds to the DynamicPartAlternative.selectorFieldCode of the respective dynamic part alternative ISignalIPdu.⌋

If the application shall be able to write the selector field value then the following specification items apply: [TPS_SYST_02352], [constr_5232], [TPS_SYST_02353], [TPS_SYST_02355], [TPS_SYST_02356], [constr_5233].

One possible use-case for the application to be able to write the selector field value is to use the applications write access as the trigger to send out one specific DynamicPartAlternative, exactly that DynamicPartAlternative which matches the written selector field value. In order to achieve this functionality the COM module needs to be configured according to the rules defined below.

The IPduM on sender side gets transmission requests from COM (for the case of application writing the selector field value: the case of trigger transmit is excluded in [TPS_SYST_02353]).

[TPS_SYST_02355] defines that only the valid DynamicPartAlternative is actually triggered for transmission by the COM module, and thus is made available to the IPduM.

From the perspective of the IPduM there is not difference whether the selector field value has been defined by initializing the COM-Signal or whether the application has written the selector field value and the COM module has filtered the proper DynamicPartAlternative to be sent to the IPduM.

[TPS_SYST_02352] Triggering in case of application writing the selector field signal ⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or
  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping,

then this ISignal representing the selector field shall be the only ISignal that is mapped into the dynamic part alternative ISignalIPdu with a transferProperty set to an arbitrary value.⌋

[constr_5232] Triggering in case of application writing the selector field signal

Imposition time: CP: IT_SysDesc

⌈If

- the ISignal representing the selector field is referenced by an ISignal-Triggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role system-Signal or
  - part of a SystemSignalGroup that in turn is referenced by a Sender-ReceiverToSignalGroupMapping

then any ISignal other than the ISignal representing the selector field shall be mapped into that dynamic part alternative ISignalIPdu using the transfer-Property set to pending.⌋

In other words, if the selector field signal is written by the application software, then the selector field ISignal may be the only ISignal which triggers the dynamic part alternative ISignalIPdu. No triggering of a dynamic part alternative ISignalIPdu may be a use case as well, in such cases a cyclic transmission would still be possible.

[TPS_SYST_02353] No support for trigger transmit in case of application writing the selector field signal ⌈If

- the ISignal representing the selector field is referenced by an ISignal-Triggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role system-Signal or
  - part of a SystemSignalGroup that in turn is referenced by a Sender-ReceiverToSignalGroupMapping

then the dynamic part alternative ISignalIPdu, where the ISignal representing the selector field is mapped to, shall not be used in a trigger transmit COM stack configuration.⌋

[TPS_SYST_02354] No support for Just-In-Time update of dynamic parts in case of application writing the selector field signal ⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or
  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then the IpduMJitUpdate configuration parameter of the IpduMTxDynamicPart corresponding to the ISignalIPdu shall be configured to false during ECU Configuration.⌋

In other words, if the selector field signal is written by the application software, then the dynamic part alternative ISignalIPdu shall only be triggered by the application layer, no trigger transmit support allowed.

[TPS_SYST_02355] **TransmissionModeDeclaration** **in case of application writing the selector field signal** ⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or
  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then the dynamic part alternative ISignalIPdu, where the ISignal representing the selector field is mapped to, shall have an ISignalIPdu.iPduTimingSpecification with a TransmissionModeDeclaration where the TransmissionModeDeclaration.transmissionModeCondition defines a TransmissionModeCondition with

- TransmissionModeCondition.iSignalInIPdu refers to the ISignalToIPduMapping which maps the ISignal representing the selector field
- DataFilter.dataFilterType = maskedNewEqualsX,

- DataFilter.mask is set to $(2^{<selectorfield_bitsize>})$ - 1, where <selector-field_bitsize> is the size of the selector field signal in bits, i.e. ISignal.length of the ISignal representing the selector field,
- DataFilter.x corresponds to the DynamicPartAlternative.selector-FieldCode of the respective dynamic part alternative ISignalIPdu,
- TransmissionModeDeclaration.transmissionModeFalseTiming shall not be defined.

⌋

[TPS_SYST_02356] Only one TransmissionModeCondition in case of application writing the selector field signal ⌈If

- the ISignal representing the selector field is referenced by an ISignal-Triggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role system-Signal or
  - part of a SystemSignalGroup that in turn is referenced by a Sender-ReceiverToSignalGroupMapping

then the dynamic part alternative ISignalIPdu, where the ISignal representing the selector field is mapped to, shall have an ISignalIPdu.iPduTiming-Specification with a TransmissionModeDeclaration where there is only one TransmissionModeCondition defined, as specified in [TPS_SYST_02355].⌋

In other words, when the application software writes the selector field signal only that dynamic part alternative ISignalIPdu with the matching selectorFieldCode is actually available for sending.

[constr_5233] Usage of invalidValue in case of application writing the selector field signal

Imposition time: CP: IT_SysDesc

⌈If

- the ISignal representing the selector field is referenced by an ISignal-Triggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and
- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either
  - referenced by a SenderReceiverToSignalMapping in the role system-Signal or

– part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then

- the ISignal representing the selector field shall either
  - define no invalid value (ISignal.networkRepresentationProps.invalidValue) or
  - the invalidValue defined shall be different than any of the defined selector field values for that MultiplexedIPdu.

⌋

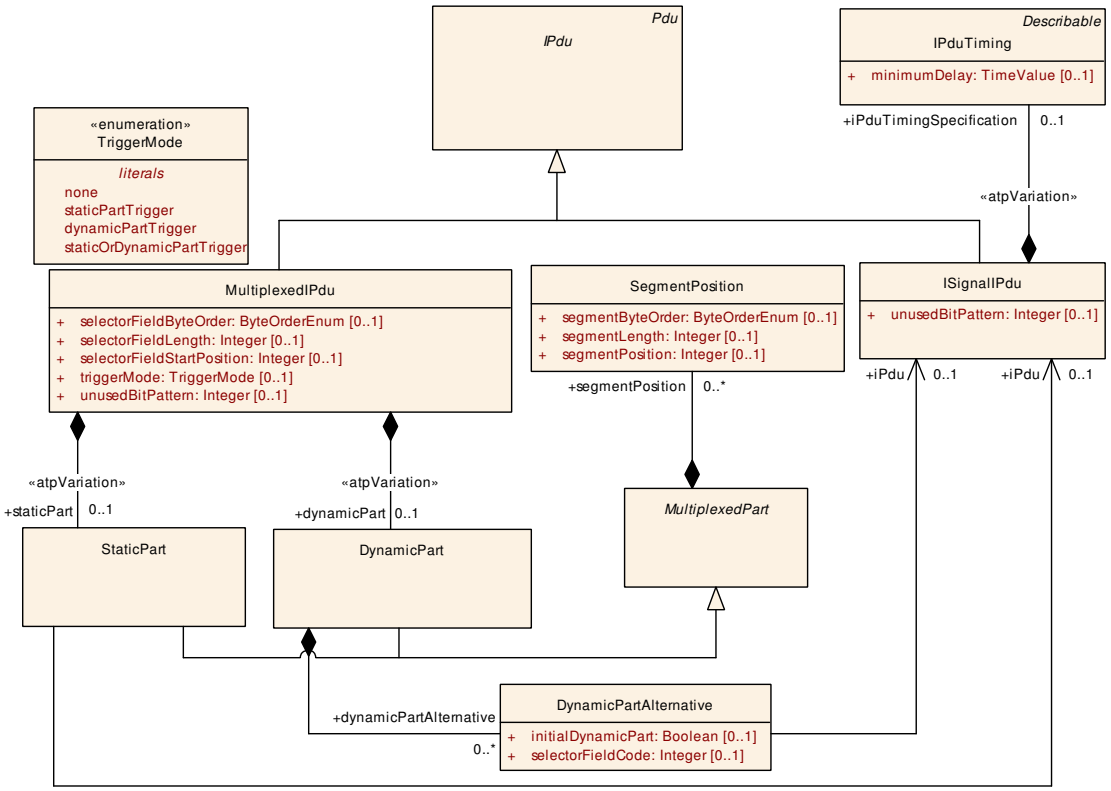

Figure 6.30: I-Pdu Multiplexer (FibexCore: IPDUMultiplexerOverview)

Enumeration	**TriggerMode**
Note	IPduM can be configured to send a transmission request for the new multiplexed I-PDU to the PDU-Router because of conditions/ modes.
Aggregated by	MultiplexedIPdu.triggerMode
Literal	***Description***
dynamicPartTrigger	IPduM sends a transmission request to the PduR if a dynamic part is received. **Tags:** atp.EnumerationLiteralIndex=0
none	IPduM does not trigger transmission because of receiving anything of this IPdu in case of Trigger Transmit. **Tags:** atp.EnumerationLiteralIndex=1
staticOrDynamic PartTrigger	IPduM sends a transmission request to the PduR if a static or dynamic part is received. **Tags:** atp.EnumerationLiteralIndex=2
staticPartTrigger	IPduM sends a transmission request to the PduR if a static part is received. **Tags:** atp.EnumerationLiteralIndex=3

Table 6.69: TriggerMode

Class	**MultiplexedIPdu**			
Note	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selector Field. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicPart	DynamicPart	0..1	aggr	According to the value of the selector field some parts of the IPdu have a different layout. In a complete System Description a MultiplexedIPdu shall contain a Dynamic Part. The following use cases support the multiplicity to be 0..1: • If a MultiplexedIPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedIPdu doesn't need to be described in the System Extract/ Ecu Extract. • If a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu then the dynamicPart does not need to be described in the System Extract/Ecu Extract. atpVariation: Content of a multiplexed PDU can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dynamicPart, dynamicPart.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**MultiplexedIPdu**			
selectorField ByteOrder	ByteOrderEnum	0..1	attr	This attribute defines the order of the bytes of the selector Field and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
selectorField Length	Integer	0..1	attr	The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
selectorField StartPosition	Integer	0..1	attr	This parameter is necessary to describe the position of the selector field within the IPdu. Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorFieldByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
staticPart	StaticPart	0..1	aggr	The static part of the multiplexed IPdu is the same regardless of the selector field. The static part is optional. atpVariation: Content of a multiplexed PDU can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticPart, staticPart.variationPoint.shortLabel vh.latestBindingTime=postBuild
triggerMode	TriggerMode	0..1	attr	IPduM can be configured to send a transmission request for the new multiplexed IPdu to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.

▽

△

Class	**MultiplexedIPdu**			
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPdu with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.

Table 6.70: MultiplexedIPdu

Class	**StaticPart**			
Note	Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.			
Base	*ARObject*, *MultiplexedPart*			
Aggregated by	MultiplexedIPdu.staticPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPdu	ISignalIPdu	0..1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu. **Tags:** atp.Splitkey=iPdu

Table 6.71: StaticPart

[constr_9176] Existence of `StaticPart.iPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `StaticPart`, the reference to `ISignalIPdu` in role `iPdu` shall exist.⌋

Class	**DynamicPart**			
Note	Dynamic part of a multiplexed I-Pdu. Reserved space which is used to transport varying SignalIPdus at the same position, controlled by the corresponding selectorFieldCode.			
Base	*ARObject*, *MultiplexedPart*			
Aggregated by	MultiplexedIPdu.dynamicPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicPart Alternative	DynamicPartAlternative	*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. **Tags:** atp.Splitkey=dynamicPartAlternative

Table 6.72: DynamicPart

[constr_9177] Existence of `DynamicPartAlternative.initialDynamicPart`

Imposition time: CP: `IT_SysDesc`

⌈For each `DynamicPartAlternative`, the attribute `initialDynamicPart` shall exist.⌋

Class	**DynamicPartAlternative**			
Note	One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.			
Base	*ARObject*			
Aggregated by	DynamicPart.dynamicPartAlternative			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialDynamic Part	Boolean	0..1	attr	Dynamic part that shall be used to initialize this multiplexed IPdu. Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.
iPdu	ISignalIPdu	0..1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu. **Tags:** atp.Splitkey=iPdu
selectorField Code	Integer	0..1	attr	The selector field is part of a multiplexed IPdu. It consists of contiguous bits. The value of the selector field selects the layout of the multiplexed part of the IPdu.

Table 6.73: DynamicPartAlternative

[constr_9178] Existence of DynamicPartAlternative.initialDynamicPart

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative the attribute initialDynamicPart shall exist.⌋

[constr_9179] Existence of DynamicPartAlternative.iPdu

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative, the reference to ISignalIPdu in role iPdu shall exist.⌋

[constr_9180] Existence of DynamicPartAlternative.selectorFieldCode

Imposition time: CP: IT_SysDesc

⌈For each DynamicPartAlternative, the attribute selectorFieldCode shall exist.⌋

Class	***MultiplexedPart*** (abstract)			
Note	The StaticPart and the DynamicPart have common properties. Both can be separated in multiple segments within the multiplexed PDU.			
Base	*ARObject*			
Subclasses	DynamicPart, StaticPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
segment Position	SegmentPosition	*	aggr	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. Therefore the StaticPart and the DynamicPart can contain multiple SegmentPositions. **Tags:** atp.Splitkey=segmentPosition

Table 6.74: MultiplexedPart

[constr_9181] Existence of MultiplexedPart.segmentPosition

Imposition time: CP: IT_SysDesc

⌈For each MultiplexedPart the aggregation of SegmentPosition in role segmentPosition shall exist.⌋

Class	SegmentPosition			
Note	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.			
Base	*ARObject*			
Aggregated by	*MultiplexedPart*.segmentPosition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
segmentByte Order	ByteOrderEnum	0..1	attr	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.
segmentLength	Integer	0..1	attr	Data Length of the segment in bits.
segment Position	Integer	0..1	attr	Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table 6.75: SegmentPosition

[constr_9182] Existence of SegmentPosition.segmentByteOrder

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentByteOrder shall exist.⌋

[constr_9183] Existence of SegmentPosition.segmentLength

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentLength shall exist.⌋

[constr_9184] Existence of SegmentPosition.segmentPosition

Imposition time: CP: IT_SysDesc

⌈For each SegmentPosition, the attribute segmentPosition shall exist.⌋

[constr_3247] Byte order mix within a MultiplexedIPdu is not allowed

Imposition time: CP: IT_SysDesc

⌈The segmentByteOrder of all SegmentPositions and the selectorField-ByteOrder shall have the same value in the MultiplexedIPdu.⌋

[constr_3223] No ByteOrderEnum.opaque allowed for MultiplexedIPdu.selectorFieldByteOrder

Imposition time: CP: IT_SysDesc

⌈The values of MultiplexedIPdu.selectorFieldByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

[constr_3224] No ByteOrderEnum.opaque allowed for SegmentPosition.segmentByteOrder.

Imposition time: CP: IT_SysDesc

⌈The values of SegmentPosition.segmentByteOrder are restricted to ByteOrderEnum.mostSignificantByteFirst and ByteOrderEnum.mostSignificantByteLast. I.e. the value ByteOrderEnum.opaque is not allowed.⌋

Figure 6.31 shows an example of an IPdu Multiplexer. The static part of the multiplexed IPdu contains ComIPduA. The value of the selector field in the dynamic part decides which content is transmitted. ComIPduB is transmitted if the selector field value is "0". ComIPduC is transmitted if the selector field value is "1".

The static and the dynamic part can consist of more than one element. These sub parts of the static or dynamic parts are called segments. In Figure 6.31 the dynamic Part is segmented into two parts. More details can be found in [26].

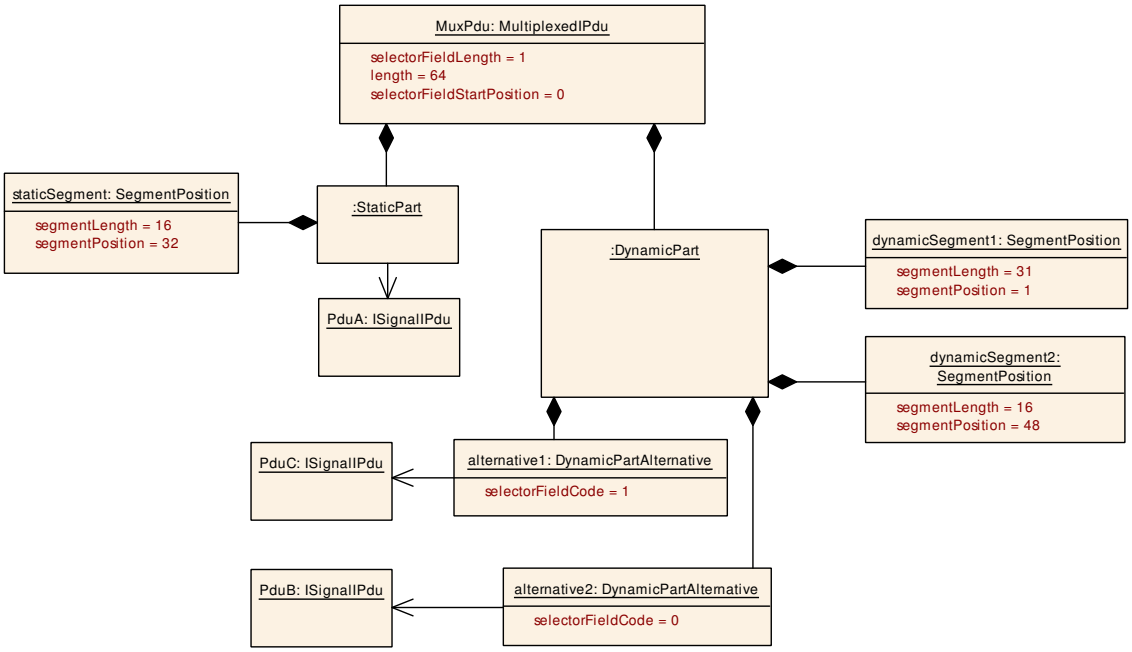

Figure 6.31: I-Pdu Multiplexer Example

Each of the following figures shows an example with an allowed IPduM configuration. Please note that the AUTOSAR IPduM module does not shift any part (static or dynamic) IPdu and just merges the payload. ISignalIPdus that are referenced by the different DynamicPartAlternatives in one MultiplexedIPdu shall always have

the same length. A configuration may be optimized with respect to unused data at end of a `StaticPart ISignalIPdu`. This is shown in figure 6.32 where the `ISignal-IPdu` that is referenced by the `StaticPart` is shorter than the `MultiplexedIPdu`. An optimization with respect to unused data at end of `DynamicPartAlternative ISignalIPdu`s is shown in figure 6.33.

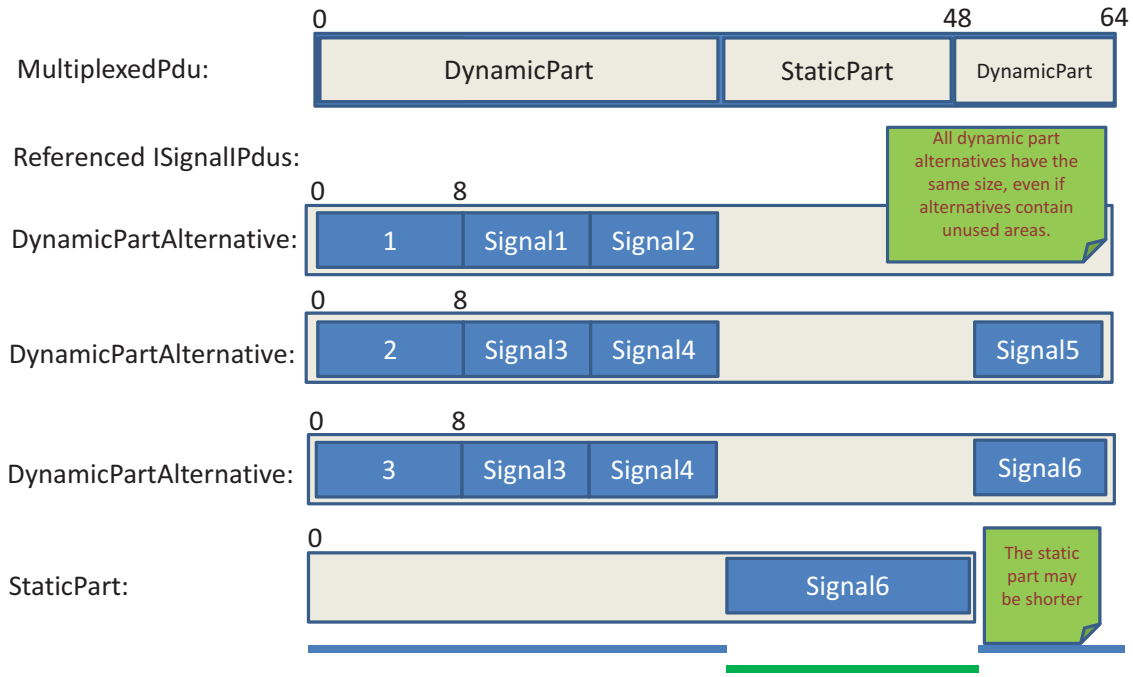

Figure 6.32: Multiplexer configuration example optimized with respect to unused data at end of static part Pdu

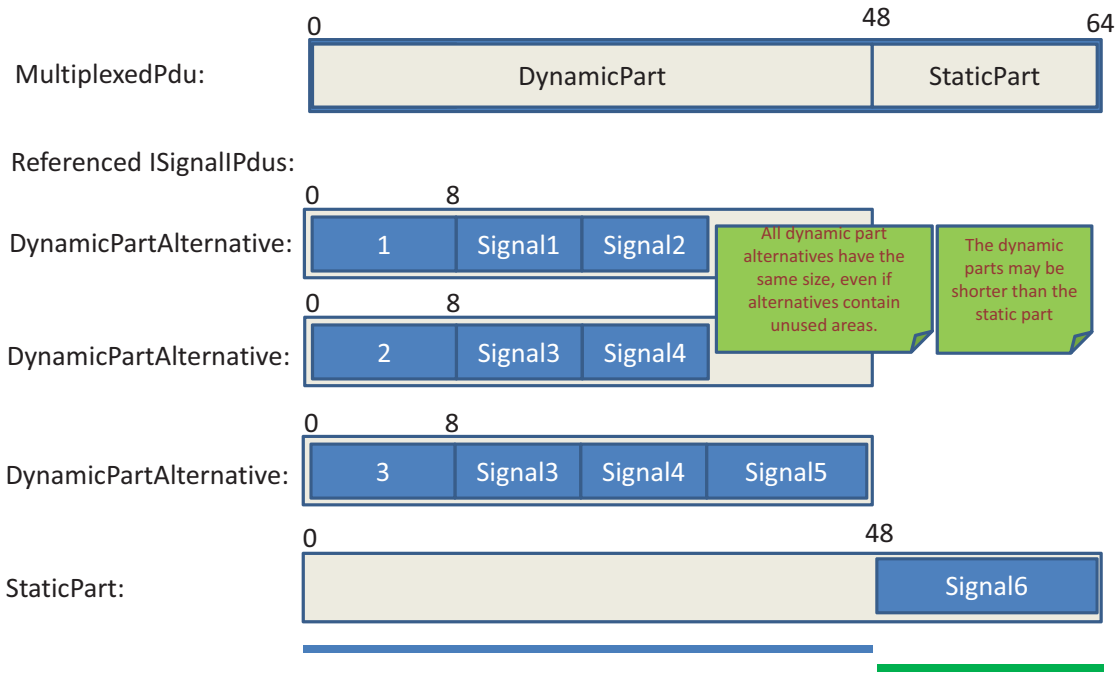

Figure 6.33: Multiplexer configuration example optimized with respect to unused data at end of dynamic part Pdus

6.5.1 I-Pdu Multiplexer in System Extract/ECU Extract

The processing in the ECU determines the description of `MultiplexedIPdu`s in the System Extract/Ecu Extract. In case that a Gateway ECU only routes a `MultiplexedIPdu` without being interested in the content leads to a reduced description in the System Extract/ECU Extract. The following items describe the different scenarios and the consequences for the System Extract/ECU Extract description. A complete System Description contains all information.

[TPS_SYST_01080] Sending or receiving of a `MultiplexedIPdu` in System Extract/ECU Extract ⌈

- all attributes of the `MultiplexedIPdu` are mandatory
- aggregated `DynamicPart` with associated `ISignalIPdu`s is mandatory in case
  - of sending
  - of receiving if at least one `DynamicPartAlternative` is received by one Ecu of the Extract.
- a `PduTriggering` shall be defined for the `MultiplexedIPdu`
- a `PduTriggering` shall be defined for all included `ISignalIPdu`s in the `DynamicPart` and `StaticPart`

⌋

The initial ECU Configuration Generator configures COM, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01081] Gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract ⌈

- `StaticPart` and `DynamicPart` definitions shall be omitted, thus no `ISignalIPdu` description shall be included
- all attributes of the `MultiplexedIPdu` shall be omitted.
- a `PduTriggering` shall be defined only for the gatewayed `MultiplexedIPdu`
- an `IPduMapping` between the source and the target `PduTriggering`s shall be defined

⌋

The initial ECU Configuration Generator configures PduR and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01082] Receiving and gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract ⌈

- all attributes of the `MultiplexedIPdu` are mandatory
- aggregated `DynamicPart` with associated `ISignalIPdu`s is mandatory in case at least one `DynamicPartAlternative` is received by one Ecu of the Extract.
- a `PduTriggering` shall be defined for the `MultiplexedIPdu`
- an `IPduMapping` between the source and the target `PduTriggering`s shall be defined
- a `PduTriggering` shall be defined for all included `ISignalIPdu`s in the `DynamicPart` and `StaticPart`

⌋

The initial ECU Configuration Generator configures Com, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

6.6 Frames

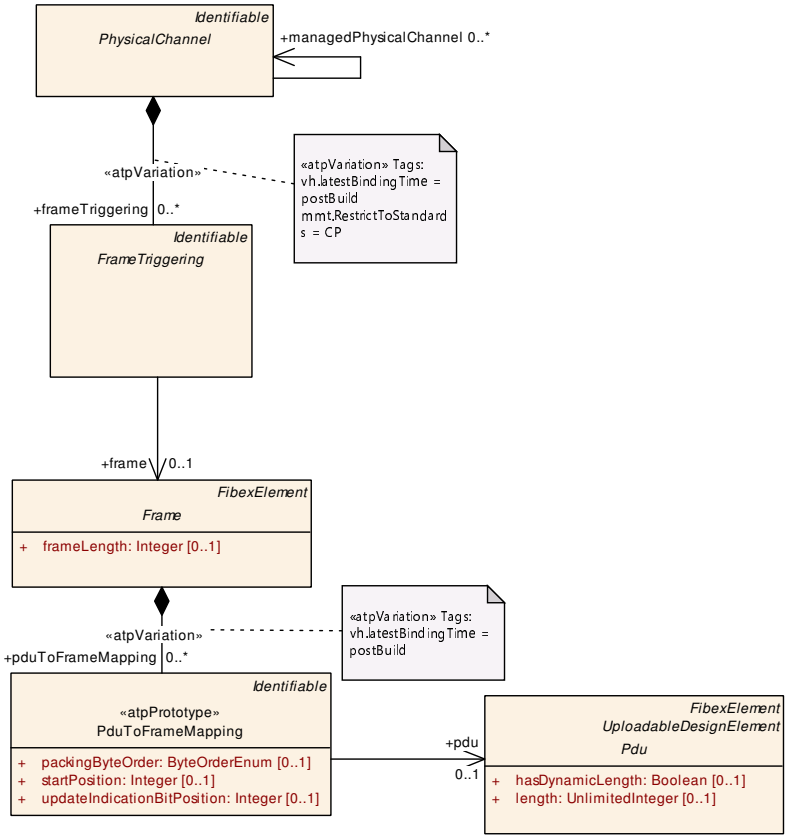

Figure 6.34: Frame Overview (FibexCore: FrameOverview)

[TPS_SYST_01083] **`Frame`** ⌈A `Frame` represents a general design object that is used to describe the layout of the included `Pdu`s as a reusable asset.⌋

[TPS_SYST_01084] **`FrameTriggering`** ⌈The `FrameTriggering` implements the reusable definition of a `Frame` within a concrete context and thus defines a `Frame`'s send behavior and identification on a certain `PhysicalChannel`.⌋

[TPS_SYST_02255] **`Frame.frameLength` usage for `FlexrayFrame`s and `CanFrame`s** ⌈The `frameLength` for a `FlexrayFrame` shall be equal or larger than the combined length of all `Pdu`s that are mapped to the frame.

The `frameLength` for a `CanFrame` is used to describe the minimum length of a received L-PDU to be accepted by a data length check. Therefore, it is possible to configure a `frameLength` which is smaller than the mapped `Pdu` to this frame. If data length check is not needed the `frameLength` of a `CanFrame` may be left undefined. The reason for that is that if the `CanFrame.frameLength` is larger than the `Pdu.length` of the mapped `Pdu` and data length check is used, a received `Pdu` will always be discarded due to a failing minimum length check.⌋

Class	***Frame*** (abstract)			
Note	Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*AbstractEthernetFrame*, CanFrame, FlexrayFrame, *LinFrame*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
frameLength	Integer	0..1	attr	The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay). The frameLength of zero bytes is allowed. Please consider also TPS_SYST_02255.
pduToFrame Mapping	PduToFrameMapping	*	aggr	A frames layout as a sequence of Pdus. atpVariation: The content of a frame can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduToFrameMapping, pduToFrame Mapping.shortName, pduToFrameMapping.variation Point.shortLabel vh.latestBindingTime=postBuild

Table 6.76: Frame

Class	***FrameTriggering*** (abstract)			
Note	The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/ in is handled by the Bus interface.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanFrameTriggering, EthernetFrameTriggering, FlexrayFrameTriggering, LinFrameTriggering			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
frame	Frame	0..1	ref	One frame can be triggered several times, e.g. on different channels. If a frame has no frame triggering, it won't be sent at all. A frame triggering has assigned exactly one frame, which it triggers. **Tags:** atp.Splitkey=frame
framePort	FramePort	*	ref	References to the FramePort on every ECU of the system which sends and/or receives the frame. References for both the sender and the receiver side shall be included when the system is completely defined. **Tags:** atp.Splitkey=framePort
pduTriggering	PduTriggering	*	ref	This reference provides the relationship to the Pdu Triggerings that are implemented by the FrameTriggering. The reference is optional since no PduTriggering can be defined for NmPdus and XCP Pdus. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.77: FrameTriggering

[constr_9131] Existence of `FrameTriggering.frame`

Imposition time: CP: `IT_SysDesc`

⌈For each `FrameTriggering`, the reference to `Frame` in the role `frame` shall exist.⌋

6.7 Specialized Attributes of the Communication Entities

In the Basic Software the timing of bus frames can be controlled by send requests of the RTE in combination with the Transmission Mode and Transfer Property parameters in COM. On the other hand the timing can be controlled by the FlexRay Interface and LIN Interface.

This chapter describes the protocol specific extensions to the communication elements.

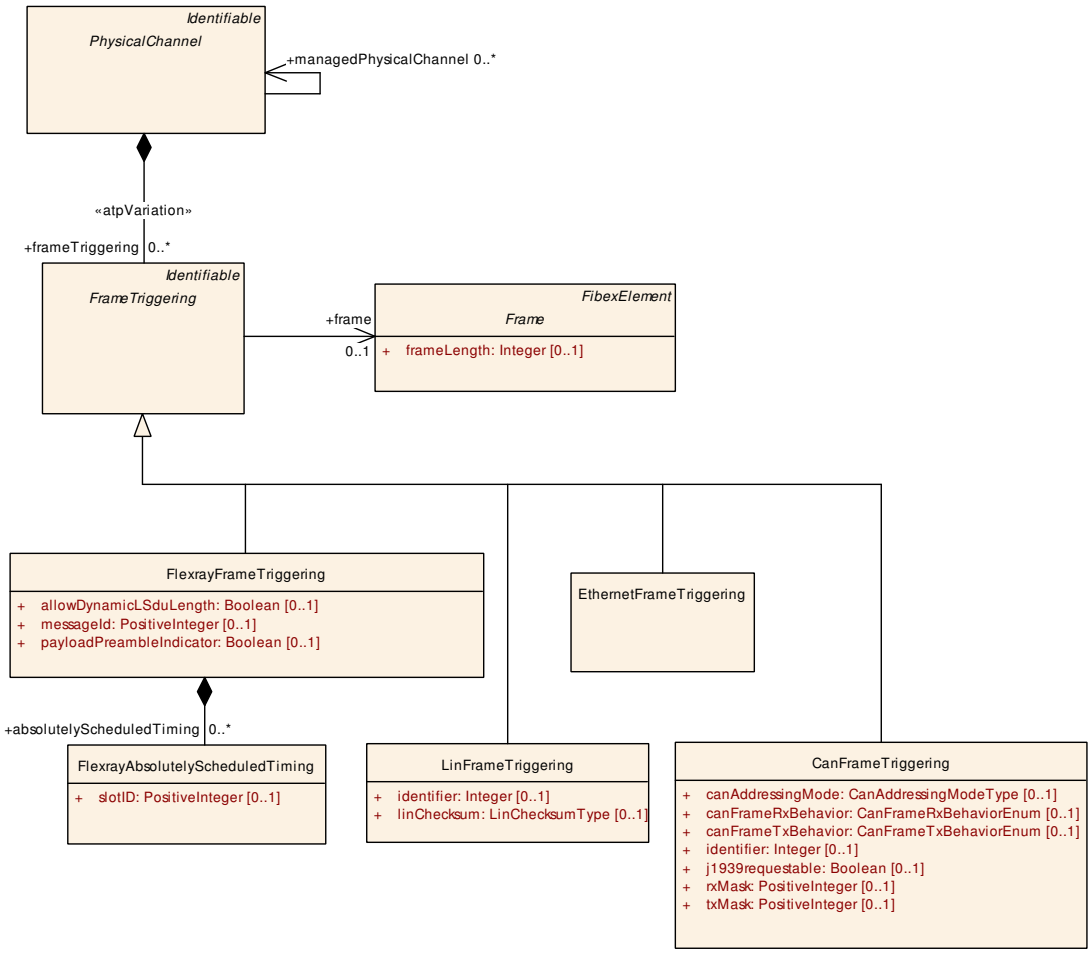

Figure 6.35: Frame Triggering

6.7.1 FlexRay specific description

[TPS_SYST_01128] Communication over FlexRay ⌈The System Template supports the description of communication over FlexRay.⌋

In the following, the elements necessary to describe the FlexRay communication are specified.

FlexRay static segment parameters: Each `FlexrayFrameTriggering` is identified by its `slotID` and `communicationCycle`. In the static segment all communication

slots are of identical, statically configured duration and all FrameTriggerings are of identical, statically configured length.

The sending behavior where the exact time for the FlexrayFrameTriggerings transmission is guaranteed is provided in the System Template by the usage of FlexrayAbsolutelyScheduledTiming.

In the cycle counter field of every frame, the current value of the cycle counter is transmitted (see FlexRay frame format). This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.

[TPS_SYST_01085] Transmission of a FrameTriggering multiple times within one communication cycle ⌈In the static segment FlexrayFrameTriggerings can be sent multiple times within one communication cycle. For describing this case multiple FlexrayAbsolutelyScheduledTimings shall be used.⌋

FlexRay dynamic segment parameters: In the dynamic segment the duration of communication slots may vary in order to accommodate frames of varying length. Furthermore, in the dynamic part, the slotID is equivalent to a priority. The higher the number the lower is the priority.

The frames in the static and in the dynamic segment are described in the same way. Each FlexrayFrameTriggering is identified by its slotID and communicationCycle. A description is provided by the usage of FlexrayAbsolutelyScheduledTiming.

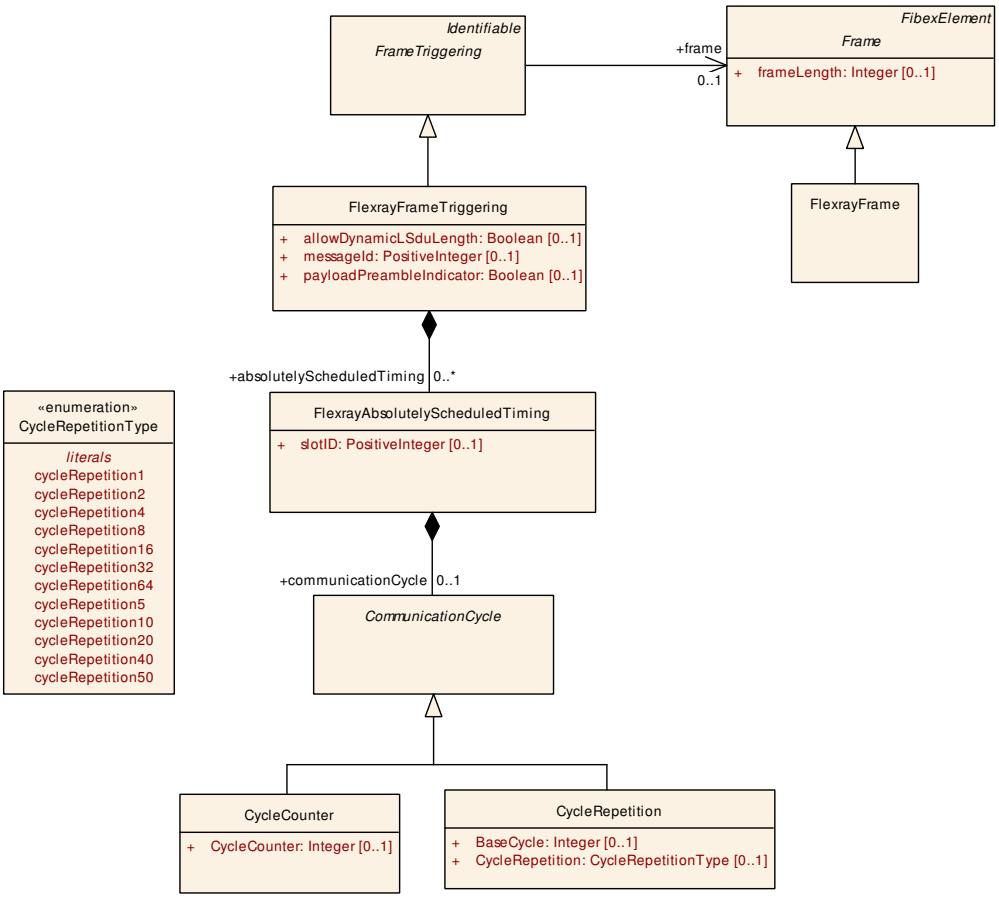

Figure 6.36: FlexRay Absolutely Scheduled Timing (Fibex4FlexRay:FlexrayAbsolutelyScheduledTiming)

Class	**FlexrayFrame**			
Note	FlexRay specific Frame element. **Tags:** atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.78: FlexrayFrame

Class	**FlexrayFrameTriggering**			
Note	FlexRay specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
absolutely Scheduled Timing	FlexrayAbsolutely ScheduledTiming	*	aggr	Specification of a sending behaviour where the exact time for the frames transmission is guaranteed. **Tags:** atp.Splitkey=absolutelyScheduledTiming

△

Class	**FlexrayFrameTriggering**			
allowDynamic LSduLength	Boolean	0..1	attr	Allows L-PDU length reduction and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU. If this attribute is set to true than the referenced Frame length attribute defines the max. length.
messageId	PositiveInteger	0..1	attr	The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.
payload Preamble Indicator	Boolean	0..1	attr	Switching the Payload Preamble bit.

Table 6.79: FlexrayFrameTriggering

[constr_9124] Existence of FlexrayFrameTriggering.allowDynamicLSduLength

Imposition time: CP: IT_SysDesc

⌈For each FlexrayFrameTriggering, the attribute allowDynamicLSduLength shall exist.⌋

[constr_9125] Existence of FlexrayFrameTriggering.payloadPreambleIndicator

Imposition time: CP: IT_SysDesc

⌈For each FlexrayFrameTriggering, the attribute payloadPreambleIndicator shall exist.⌋

Class	**FlexrayAbsolutelyScheduledTiming**			
Note	Each frame in FlexRay is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming. In the static segment a frame can be sent multiple times within one communication cycle. For describing this case multiple AbsolutelyScheduledTimings have to be used. The main use case would be that a frame is sent twice within one communication cycle.			
Base	*ARObject*			
Aggregated by	FlexrayFrameTriggering.absolutelyScheduledTiming			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Cycle	CommunicationCycle	0..1	aggr	The communication cycle where the frame is sent. **Tags:** atp.Splitkey=communicationCycle

▽

△

Class	FlexrayAbsolutelyScheduledTiming			
slotID	PositiveInteger	0..1	attr	In the static part the SlotID defines the slot in which the frame is transmitted. The SlotID also determines, in combination with FlexrayCluster::numberOfStaticSlots, whether the frame is sent in static or dynamic segment. In the dynamic part, the slot id is equivalent to a priority. Lower dynamic slot ids are all sent until the end of the dynamic segment. Higher numbers, which were ignored that time, have to wait one cycle and then shall try again. minValue: 1 maxValue: 2047

Table 6.80: FlexrayAbsolutelyScheduledTiming

[constr_9126] Existence of FlexrayAbsolutelyScheduledTiming.slotID

Imposition time: CP: IT_SysDesc

⌈For each FlexrayAbsolutelyScheduledTiming, the attribute slotID shall exist.⌋

[constr_9127] Existence of FlexrayAbsolutelyScheduledTiming.communicationCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayAbsolutelyScheduledTiming, the aggregation of CommunicationCycle in the role communicationCycle shall exist.⌋

Class	*CommunicationCycle* (abstract)			
Note	The communication cycle where the frame is sent.			
Base	*ARObject*			
Subclasses	CycleCounter, CycleRepetition			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.81: CommunicationCycle

The communication cycle can be described by the CycleCounter or by the CycleRepetition:

Class	CycleCounter			
Note	The communication cycle where the frame is send is described by the attribute "cycleCounter".			
Base	*ARObject*, *CommunicationCycle*			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**CycleCounter**			
CycleCounter	Integer	0..1	attr	The communication cycle where the frame described by this timing is sent. If a timing is given in this way the referencing FlexrayCluster shall specify the cycleCount Max as upper bound and point of total repetition. This value is incremented at the beginning of each new cycle, ranging from 0 to cycleCountMax, and is reset to 0 after a sequence of cycleCountMax+1 cycles.

Table 6.82: CycleCounter

[constr_9128] Existence of `CycleCounter.CycleCounter`

Imposition time: CP: `IT_SysDesc`

⌈For each `CycleCounter`, the attribute `CycleCounter` shall exist.⌋

Class	**CycleRepetition**			
Note	The communication cycle where the frame is send is described by the attributes baseCycle and cycle Repetition.			
Base	*ARObject*, *CommunicationCycle*			
Aggregated by	FlexrayAbsolutelyScheduledTiming.communicationCycle			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
BaseCycle	Integer	0..1	attr	The first communication cycle where the frame is sent. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.
CycleRepetition	CycleRepetitionType	0..1	attr	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.

Table 6.83: CycleRepetition

Enumeration	**CycleRepetitionType**
Note	The number of communication cycles (after the first cycle) whenever the frame is sent again. The FlexRay communication controller allows only determined values.
Aggregated by	CycleRepetition.CycleRepetition
Literal	***Description***
cycleRepetition1	Attribute cycleRepetition value="1" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=0
cycleRepetition10	Attribute cycleRepetition value="10" to support FlexRay 3.0 **Tags:** atp.EnumerationLiteralIndex=1
cycleRepetition16	Attribute cycleRepetition value="16" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=2
cycleRepetition2	Attribute cycleRepetition value="2" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=3

▽

△

Enumeration	CycleRepetitionType
cycleRepetition20	Attribute cycleRepetition value="20" to support FlexRay 3.0 **Tags:** atp.EnumerationLiteralIndex=4
cycleRepetition32	Attribute cycleRepetition value="32" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=5
cycleRepetition4	Attribute cycleRepetition value="4" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=6
cycleRepetition40	Attribute cycleRepetition value="40" to support FlexRay 3.0 **Tags:** atp.EnumerationLiteralIndex=7
cycleRepetition5	Attribute cycleRepetition value="5" to support FlexRay 3.0 **Tags:** atp.EnumerationLiteralIndex=8
cycleRepetition50	Attribute cycleRepetition value="50" to support FlexRay 3.0 **Tags:** atp.EnumerationLiteralIndex=9
cycleRepetition64	Attribute cycleRepetition value="64" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=10
cycleRepetition8	Attribute cycleRepetition value="8" valid only for FlexRay Protocol 2.1 Rev A **Tags:** atp.EnumerationLiteralIndex=11

Table 6.84: CycleRepetitionType

[constr_9129] Existence of CycleRepetition.BaseCycle

Imposition time: CP: IT_SysDesc

⌈For each CycleRepetition, the attribute BaseCycle shall exist.⌋

[constr_9130] Existence of CycleRepetition.CycleRepetition

Imposition time: CP: IT_SysDesc

⌈For each CycleRepetition the attribute CycleRepetition shall exist.⌋

[constr_3012] Overlapping of Pdus is prohibited

Imposition time: CP: IT_SysDesc

⌈Pdus mapped to a FlexrayFrame shall NOT overlap.⌋

[constr_3013] FlexrayFrame length shall not be exceeded

Imposition time: CP: IT_SysDesc

⌈The combined length of all Pdus that are mapped into a FlexrayFrame shall not exceed the defined FlexrayFrame length.⌋

[constr_3014] Overlapping of updateIndicationBits for Pdus is prohibited

Imposition time: CP: IT_SysDesc

⌈The updateIndicationBitPosition for a Pdu in a FlexrayFrame shall NOT overlap with other updateIndicationBitPositions and Pdu locations.⌋

[constr_5104] Assignment of a FlexrayFrame where allowDynamicLSduLength is set to true

Imposition time: CP: IT_SysDesc

⌈FlexrayFrames which are referenced by a FlexrayFrameTriggering where allowDynamicLSduLength is set to true shall always be assigned to the dynamic segment.⌋

[constr_5105] Mapping of Pdu with dynamic length in a FlexrayFrame

Imposition time: CP: IT_SysDesc

⌈Only the last Pdu in a FlexrayFrame is allowed to qualify according to [TPS_SYST_03085] to be of dynamic length.⌋

Note: Please be aware that the dynamic Pdu at the end of the FlexRay Frame may need to provide some mechanism to determine its actual length (e.g. length field, termination). Otherwise the receiver is not able to determine the actual sent length.

6.7.2 LIN specific description

LIN is a protocol that is based on a single master - multiple slave principle. In the following, the parameters will be specified, which are necessary to describe the LIN Schedule Tables and the LIN Frames.

[TPS_SYST_01129] Communication over LIN ⌈The System Template supports the description of communication over LIN.⌋

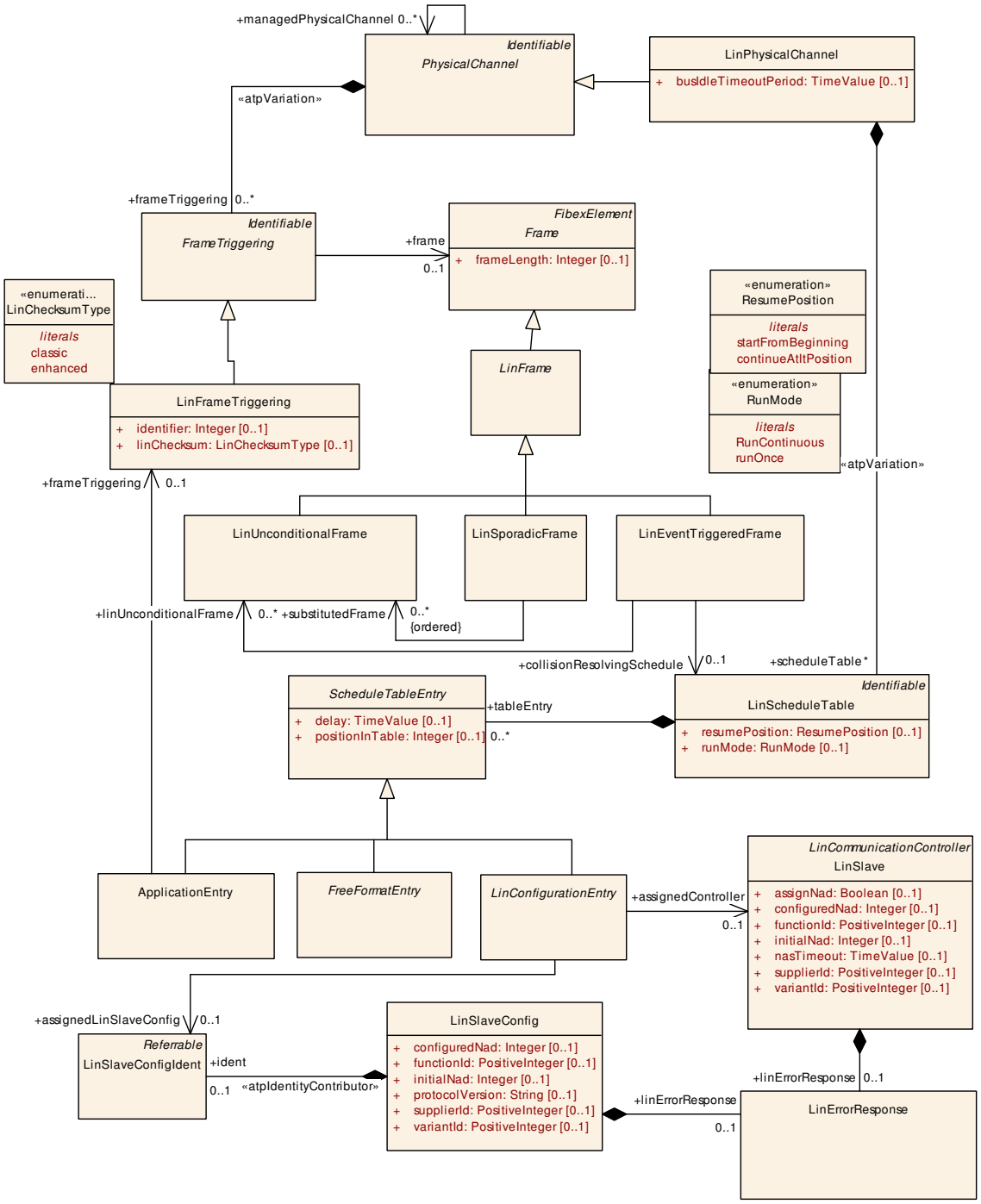

Figure 6.37: LIN Schedule Table (Fibex4Lin:LinScheduleTable)

6.7.2.1 LIN Frames

One LIN Frame consists of two parts: header and response. The header is always sent by a `LinMaster`, while the response is sent by only one dedicated `LinSlave`. There are three different ways of transmitting frames on the bus: unconditional, event triggered, and sporadic frames.

Class	**LinFrame** (abstract)			
Note	Lin specific Frame element. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	LinEventTriggeredFrame, LinSporadicFrame, LinUnconditionalFrame			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.85: LinFrame

Class	**LinFrameTriggering**			
Note	LIN specific attributes to the FrameTriggering This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
identifier	Integer	0..1	attr	To describe a frames identifier on the communication system, usually with a fixed identifierValue. For Lin SporadicFrames the attribute shall be ignored.
linChecksum	LinChecksumType	0..1	attr	Type of checksum that the frame is using. This attribute is optional because in case of sporadic frames it should not be set.

Table 6.86: LinFrameTriggering

Enumeration	**LinChecksumType**
Note	Use of classic or enhanced checksum is managed by the master node and it is determined per frame identifier; This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	LinFrameTriggering.linChecksum
Literal	**Description**
classic	Classic in communication with LIN 1.3 slave nodes **Tags:** atp.EnumerationLiteralIndex=0
enhanced	Enhanced in communication with LIN 2.0 slave nodes. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.87: LinChecksumType

[TPS_SYST_02095] **`LinFrameTriggering.linChecksum` for `LinUnconditionalFrames`** ⌈The `linChecksum` attribute of a `LinFrameTriggering` that references a `LinUnconditionalFrame` shall be set.⌋

Class	LinUnconditionalFrame			
Note	Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data. **Tags:** atp.recommendedPackage=Frames This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table 6.88: LinUnconditionalFrame

[constr_3225] LinFrameTriggering.linChecksum not allowed for LinSporadicFrames

Imposition time: CP: IT_SysDesc

⌈The linChecksum attribute of a LinFrameTriggering that references a LinSporadicFrame shall not be set.⌋

[constr_3226] LinFrameTriggering.linChecksum for LinEventTriggeredFrames

Imposition time: CP: IT_SysDesc

⌈Within a PhysicalChannel the linChecksum attribute of a LinFrameTriggering that references a LinEventTriggeredFrame shall have the same value as the linChecksum attribute of each LinFrameTriggering that references a LinUnconditionalFrame that in turn is referenced by that LinEventTriggeredFrame.⌋

[constr_3203] LinFrameTriggering to LinSporadicFrame reference restriction in LinSporadicFrame context

Imposition time: CP: IT_SysDesc

⌈Within a PhysicalChannel a LinUnconditionalFrame shall be referenced by only one LinFrameTriggering to allow a derivation of the identifier of a substituted Frame if the LinUnconditionalFrame is referenced by a LinSporadicFrame in the role substitutedFrame.⌋

Class	LinSporadicFrame			
Note	A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus. **Tags:** atp.recommendedPackage=Frames This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**LinSporadicFrame**			
substituted Frame (ordered)	LinUnconditionalFrame	*	ref	Reference to a group of unconditional frames that share the same frame slot. In case that more than one of the declared frames needs to be transferred, the one first listed shall be chosen. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. A LinUnconditionalFrame associated with a LinSporadic Frame may not be allocated in the same LinSchedule Table as the sporadic frame.

Table 6.89: LinSporadicFrame

[constr_3204] LinUnconditionalFrames associated with a LinSporadicFrame

Imposition time: CP: IT_SysDesc

⌈A LinUnconditionalFrame associated with a LinSporadicFrame shall not be allocated in the same LinScheduleTable as the LinSporadicFrame.⌋

[constr_3205] Existence of FramePort for a FrameTriggering that references a LinSporadicFrame

Imposition time: CP: IT_SysDesc

⌈A FrameTriggering that references a LinSporadicFrame shall not have a reference to a FramePort.⌋

[constr_9132] Existence of LinSporadicFrame.substitutedFrame

Imposition time: CP: IT_SysDesc

⌈For each LinSporadicFrame, at least one reference to LinUnconditionalFrame in the role substitutedFrame shall exist.⌋

Instead of the LinSporadicFrame a LinUnconditionalFrame is sent in the timeslot on the bus and therefore the FrameTriggering that references a LinSporadicFrame does not need to have a reference to a FramePort.

Class	**LinEventTriggeredFrame**
Note	An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response. The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable. The event controlled frame shall not contain any Pdus. **Tags:** atp.recommendedPackage=Frames This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	**LinEventTriggeredFrame**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
collision Resolving Schedule	LinScheduleTable	0..1	ref	Reference to the schedule table, which resolves a collision. **Tags:** atp.Splitkey=collisionResolvingSchedule
linUnconditional Frame	LinUnconditionalFrame	*	ref	A list of slaves can respond to the master request if at least one of the signals carried in its unconditional frame is updated. For each response a LinFrameTriggering and a LinUnconditionalFrame shall be defined. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. The Unconditional frames associated with an event triggered frame shall: • have equal length. • use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed). • reserve the first data field to its protected identifier (even if the associated unconditional frame is scheduled as a unconditional frame in the same or another schedule table). • be published by different slave nodes. • shall not be included directly in the same schedule table as the event triggered frame is scheduled. **Tags:** atp.Splitkey=linUnconditionalFrame

Table 6.90: LinEventTriggeredFrame

[TPS_SYST_02077] Subscribers of a `LinEventTriggeredFrame` ⌈For each subscriber of a `LinEventTriggeredFrame` a `LinUnconditionalFrame` and a `LinFrameTriggering` that points to this `LinUnconditionalFrame` shall be defined.⌋

[constr_3202] `LinFrameTriggering` to `LinUnconditionalFrame` reference restriction in `LinEventTriggeredFrame` context

Imposition time: CP: `IT_SysDesc`

⌈Within a `PhysicalChannel` a `LinUnconditionalFrame` shall be referenced by only one `LinFrameTriggering` to allow a derivation of the identifier of a substituted Frame if the `LinUnconditionalFrame` is referenced by a `LinEventTriggeredFrame` in the role `linUnconditionalFrame`.⌋

[constr_3206] Existence of `FramePort` for a `FrameTriggering` that references a `LinEventTriggeredFrame`

Imposition time: CP: `IT_SysDesc`

⌈A `FrameTriggering` that references a `LinEventTriggeredFrame` shall not have a reference to a `FramePort`.⌋

[constr_9133] Existence of LinEventTriggeredFrame.linUnconditionalFrame

Imposition time: CP: IT_SysDesc

⌈For each LinEventTriggeredFrame, at least one reference to LinUnconditionalFrame in the role linUnconditionalFrame shall exist.⌋

A LinUnconditionalFrame is sent as the response of a LinEventTriggeredFrame on the bus instead and therefore the FrameTriggering that references a LinEventTriggeredFrame does not need to have a reference to a FramePort.

[TPS_SYST_02078] LinUnconditionalFrames associated with a LinEventTriggeredFrame ⌈The LinUnconditionalFrames associated with a LinEventTriggeredFrame shall:

- have equal length
- use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed)
- reserve the first data field to its protected identifier (even if the associated LinUnconditionalFrame is scheduled as a LinUnconditionalFrame in the same or another schedule table)
- be published by different slave nodes
- not be included directly in the same LinScheduleTable as the associated LinEventTriggeredFrame.

⌋

6.7.2.2 LIN Schedule Table

The LinMaster uses one or more predefined scheduling tables to start the sending and receiving to the LIN bus. These scheduling tables contain at least the relative timing that defines the message sending.

[constr_1657] Existence of LinPhysicalChannel.scheduleTable

Imposition time: CP: IT_SysDesc

⌈In any given Ecu Extract that contains a LinSlave, the LinPhysicalChannel that relates to the respective LinSlave via commConnector.commController shall not aggregate a LinScheduleTable.⌋

Class	**LinScheduleTable**			
Note	The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LinPhysicalChannel.scheduleTable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
resumePosition	ResumePosition	0..1	attr	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
runMode	RunMode	0..1	attr	The schedule table can be executed in two different modes.
tableEntry	ScheduleTableEntry	*	aggr	The scheduling table consists of table entries, which contain Frame slots. **Tags:** atp.Splitkey=tableEntry

Table 6.91: LinScheduleTable

Enumeration	**RunMode**
Note	The schedule table can be executed in two different modes. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	LinScheduleTable.runMode
Literal	***Description***
RunContinuous	RUN_CONTINUOUS run mode **Tags:** atp.EnumerationLiteralIndex=0
runOnce	RUN_ONCE run mode **Tags:** atp.EnumerationLiteralIndex=1

Table 6.92: RunMode

Enumeration	**ResumePosition**
Note	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	LinScheduleTable.resumePosition
Literal	***Description***
continueAtItPosition	Continue at IT Point. **Tags:** atp.EnumerationLiteralIndex=0
startFromBeginning	Start from the beginning **Tags:** atp.EnumerationLiteralIndex=1

Table 6.93: ResumePosition

Class	***ScheduleTableEntry*** (abstract)
Note	Table entry in a LinScheduleTable. Specifies what will be done in the frame slot. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*
Subclasses	ApplicationEntry, *FreeFormatEntry*, *LinConfigurationEntry*

△

Class	**ScheduleTableEntry** (abstract)			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
delay	TimeValue	0..1	attr	Relative delay between this tableEntry and the start of the successor in the schedule table in seconds.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the schedule table entry. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-10
positionInTable	Integer	0..1	attr	Relative position in the schedule table. The first entry index in the schedule table is 0.

Table 6.94: ScheduleTableEntry

[constr_9134] Existence of ScheduleTableEntry.delay

Imposition time: CP: IT_SysDesc

⌈For each ScheduleTableEntry the attribute delay shall exist.⌋

[constr_9135] Existence of ScheduleTableEntry.positionInTable

Imposition time: CP: IT_SysDesc

⌈For each ScheduleTableEntry, the attribute positionInTable shall exist.⌋

Class	**ApplicationEntry**			
Note	Schedule table entry for application messages. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frameTriggering	LinFrameTriggering	0..1	ref	Specifies the LinFrame that will be transmitted in this frame slot. **Tags:** atp.Splitkey=frameTriggering

Table 6.95: ApplicationEntry

[constr_9136] Existence of ApplicationEntry.frameTriggering

Imposition time: CP: IT_SysDesc

⌈For each ApplicationEntry, the reference to FrameTriggering in the role frameTriggering shall exist.⌋

Class	**FreeFormatEntry** (abstract)
Note	FreeFormat transmits a fixed master request frame with the eight data bytes provided. This may for instance be used to issue user specific fixed frames. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *ScheduleTableEntry*

▽

△

Class	***FreeFormatEntry*** (abstract)			
Subclasses	FreeFormat			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.96: FreeFormatEntry

Class	***LinConfigurationEntry*** (abstract)			
Note	A ScheduleTableEntry which contains LIN specific assignments. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *ScheduleTableEntry*			
Subclasses	AssignFrameId, AssignFrameIdRange, AssignNad, ConditionalChangeNad, DataDumpEntry, Save ConfigurationEntry, UnassignFrameId			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assigned Controller	LinSlave	0..1	ref	The LIN slaves controller who is target of this assignment. Optional in case LinConfigurationEntry.assignedLinSlave Config exists. **Tags:** atp.Splitkey=assignedController
assignedLin SlaveConfig	LinSlaveConfigIdent	0..1	ref	The LIN slave that is target of this assignment. Please note that this reference is redundant to the assignedController reference. In an Ecu Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the LinMaster. **Tags:** atp.Splitkey=assignedLinSlaveConfig

Table 6.97: LinConfigurationEntry

6.7.2.3 Configuration Services

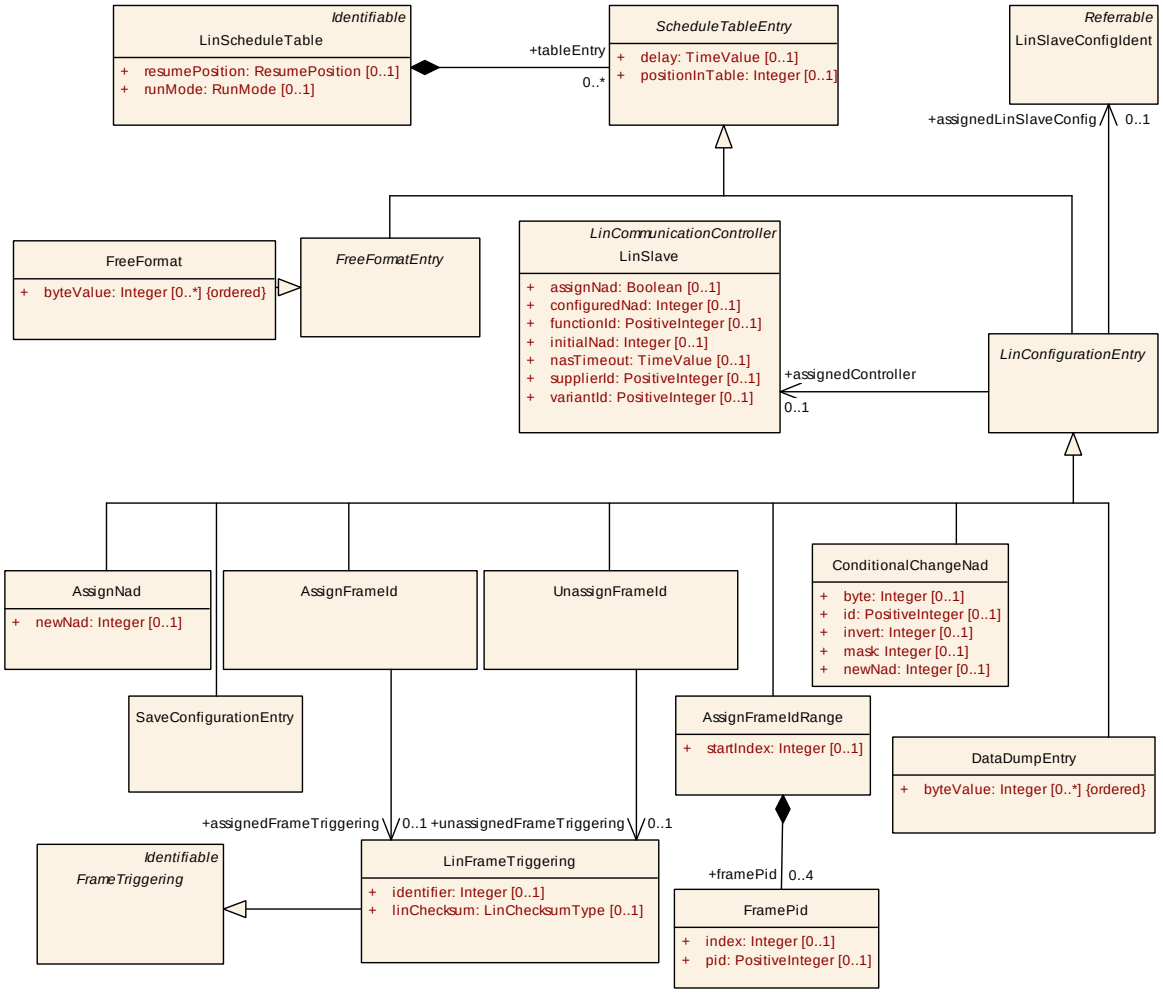

Figure 6.38: LIN Configuration Entries (Fibex4Lin:LinConfigurationEntries)

LIN only supports 64 identifiers. That creates the need for extending the address space. Hence the frames are identified by message ids from a much larger address space that is additionally separated by supplier ids. During runtime the master assigns a LinId to the frame. In case of identical parts within a cluster the initial node ID (oldNad) is used to differentiate such nodes.

To support that in System Template the `AssignFrameId` is introduced as a LIN specific extension. For the assignment a relation to the `LinSlave` is used. The `LinSlave` element is referenced by a `LinCommunicationConnector` element that contains a list of frames processed by the slave node.

Class	**AssignFrameId**
Note	Schedule entry for an Assign Frame Id master request. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*

△

Class	AssignFrameId			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	Type	Mult.	Kind	Note
assignedFrame Triggering	LinFrameTriggering	0..1	ref	The frame whose identifier is set by this assignment. **Tags:** atp.Splitkey=assignedFrameTriggering

Table 6.98: AssignFrameId

Class	UnassignFrameId			
Note	Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	Type	Mult.	Kind	Note
unassigned FrameTriggering	LinFrameTriggering	0..1	ref	The frame whose identifier is reset by this assignment. **Tags:** atp.Splitkey=unassignedFrameTriggering

Table 6.99: UnassignFrameId

[constr_9137] Existence of AssignFrameId.assignedFrameTriggering

Imposition time: CP: IT_SysDesc

⌈For each AssignFrameId, the reference to LinFrameTriggering in the role assignedFrameTriggering shall exist.⌋

[constr_9138] Existence of UnassignFrameId.unassignedFrameTriggering

Imposition time: CP: IT_SysDesc

⌈For each UnassignFrameId, the reference to LinFrameTriggering in the role unassignedFrameTriggering shall exist.⌋

[TPS_SYST_02363] messageId of AssignFrameId and UnassignFrameId

⌈In case that the AssignFrameId or UnassignFrameId refers to a LinSlave in the role assignedController the messageId of the AssignFrameId/UnassignFrameId can be derived from the messageId of the LinConfigurableFrame that references the same LinFrame as the LinFrameTriggering that is referenced by the AssignFrameId/UnassignFrameId and that is aggregated by the LinCommunicationConnector in role linConfigurableFrame that points to this LinSlave in the role commController.

In case that the AssignFrameId/UnassignFrameId refers to a LinSlaveConfigIdent in the role assignedLinSlaveConfig the messageId of the AssignFrameId/UnassignFrameId can also be derived from the messageId of the LinConfigurableFrame that references the same LinFrame as the LinFrameTriggering that is referenced by the AssignFrameId/UnassignFrameId and that is aggregated by the referenced LinSlaveConfig.⌋

The Assign frame ID configuration service is replaced in LIN 2.1 by the Assign frame ID range configuration service. AssignFrameIdRange is used to set or disable Protected Identifiers up to four frames. For the assignment a relation to the LinSlave is used. The LinSlave element is referenced by a LinCommunicationConnector element that contains a list of frames processed by the slave node.

Class	**AssignFrameIdRange**			
Note	AssignFrameIdRange generates an assign frame PID range request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
framePid	FramePid	0..4	aggr	Optional assignment of frame_PID values that are included in the request. The frame_PIDs are ordered. **Tags:** atp.Splitkey=framePid
startIndex	Integer	0..1	attr	The startIndex sets the index to the first frame to assign a PID.

Table 6.100: AssignFrameIdRange

[constr_9139] Existence of AssignFrameIdRange.startIndex

Imposition time: CP: IT_SysDesc

⌈For each AssignFrameIdRange, the attribute startIndex shall exist.⌋

Class	**FramePid**			
Note	Frame_PIDs that are included in the request. The "pid" attribute describes the value and the "index" attribute the position of the frame_PID in the request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	AssignFrameIdRange.framePid			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
index	Integer	0..1	attr	This attribute is used to order the frame_PIDs. The values of index shall be unique within one AssignFrameIdRange.
pid	PositiveInteger	0..1	attr	Frame_PID value.

Table 6.101: FramePid

[constr_9140] Existence of FramePid.index

Imposition time: CP: IT_SysDesc

⌈For each FramePid, the attribute index shall exist.⌋

[constr_9141] Existence of FramePid.pid

Imposition time: CP: IT_SysDesc

⌈For each FramePid, the attribute pid shall exist.⌋

[constr_5031] Uniqueness of FramePid.index

Imposition time: CP: IT_SysDesc

⌈FramePid.index shall always be set and be unique in the context of the aggregating AssignFrameIdRange.⌋

Assign NAD is used to resolve conflicting NADs in LIN clusters built using off-the-shelves slave nodes or reused slave nodes. This request uses the initial NAD. The NAD used for the response shall be the same as in the request, i.e. the initial NAD.

Class	**AssignNad**			
Note	Schedule entry for an Assign NAD master request. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
newNad	Integer	0..1	attr	The newly assigned NAD value.

Table 6.102: AssignNad

[constr_9142] Existence of AssignNad.newNad

Imposition time: CP: IT_SysDesc

⌈For each AssignNad, the attribute newNad shall exist.⌋

The conditional change NAD is used to detect unknown slave nodes in a cluster and to separate their NADs.

Class	**ConditionalChangeNad**			
Note	Generates an conditional change NAD request. See ISO 17987 protocol specification for more information. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
byte	Integer	0..1	attr	Byte Position of Data Byte that should be used for the bitwise XOR with Invert and the bitwise AND with Mask.
id	PositiveInteger	0..1	attr	Byte Position of Id.
invert	Integer	0..1	attr	Byte Position of Invert.
mask	Integer	0..1	attr	Byte Position of Mask.
newNad	Integer	0..1	attr	The newly assigned NAD value (Byte Position).

Table 6.103: ConditionalChangeNad

[constr_9143] Existence of ConditionalChangeNad.byte

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute byte shall exist.⌋

[constr_9144] Existence of ConditionalChangeNad.id

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute id shall exist.⌋

[constr_9145] Existence of ConditionalChangeNad.invert

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute invert shall exist.⌋

[constr_9146] Existence of ConditionalChangeNad.mask

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute mask shall exist.⌋

[constr_9147] Existence of ConditionalChangeNad.newNad

Imposition time: CP: IT_SysDesc

⌈For each ConditionalChangeNad, the attribute newNad shall exist.⌋

The Save Configuration service tells the slave node that the slave application shall save the current configuration.

Class	**SaveConfigurationEntry**			
Note	This service is used to notify a slave node to store its configuration. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.104: SaveConfigurationEntry

The Data Dump service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific.

Class	**DataDumpEntry**			
Note	This service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
byteValue (ordered)	Integer	*	attr	Supplier specific format.

Table 6.105: DataDumpEntry

[constr_9148] Existence of DataDumpEntry.byteValue

Imposition time: CP: IT_SysDesc

⌈For each DataDumpEntry, 5 byteValues shall be defined.⌋

With the FreeFormat a scheduling of fixed data content within a diagnostic frame is defined. For that specification FreeFormat is introduced.

Class	**FreeFormat**			
Note	Representing freely defined data. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *FreeFormatEntry*, *ScheduleTableEntry*			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
byteValue (ordered)	Integer	*	attr	The integer Value of a freely defined data byte.

Table 6.106: FreeFormat

[constr_9149] Existence of FreeFormat.byteValue

Imposition time: CP: IT_SysDesc

⌈For each FreeFormat, 8 byteValues shall be defined.⌋

In order to be consistent with the rest of the communication configuration, it is required that the diagnostic LIN Frames (Master Request Frame, Slave Response Frame) are explicitly modeled as Frame elements. LinFrameTriggerings dealing with diagnostic Frames thus reference this diagnostic frames.

[TPS_SYST_02276] Modeling of LIN master request frames ⌈A LIN master request frame shall be modeled per LinPhysicalChannel in terms of a LinFrameTriggering referencing a LinUnconditionalFrame while the following rules apply:

- The LinFrameTriggering has identifier set to 60 (0x3C) and linChecksum set to classic.
- The LinFrameTriggering has a reference to an “out” FramePort of the LinMaster and “in” FramePorts of the respective LinSlaves in the scope of the same LinPhysicalChannel.
- The LinFrameTriggering references a LinUnconditionalFrame with frameLength set to 8.
- The LinFrameTriggering references a PduTriggering that in turn references a NPdu that is mapped to the LinUnconditionalFrame and that has its length set to 8. The NPdu is referenced by a LinTpConnection in the role dataPdu.
- The PduTriggering has a reference to an “out” IPduPort of the LinMaster and “in” IPduPorts of the respective LinSlaves in the scope of the same LinPhysicalChannel.

⌋

[TPS_SYST_02277] Modeling of LIN slave response frames ⌈A LIN slave response frame shall be modeled per LinPhysicalChannel in terms of a LinFrameTriggering referencing a LinUnconditionalFrame while the following rules apply:

- The LinFrameTriggering has identifier set to 61 (0x3D) and linChecksum set to classic.
- The LinFrameTriggering has a reference to an "in" FramePort of the LinMaster and "out" FramePorts of the respective LinSlaves in the scope of the same LinPhysicalChannel.
- The LinFrameTriggering references a LinUnconditionalFrame with frameLength set to 8.
- The LinFrameTriggering references a PduTriggering that in turn references a NPdu that is mapped to the LinUnconditionalFrame and that has its length set to 8. The NPdu is referenced by a LinTpConnection in the role dataPdu.
- The PduTriggering has a reference to an "in" IPduPort of the LinMaster and "out" IPduPorts of the respective LinSlaves in the scope of the same LinPhysicalChannel.

⌋

6.7.3 CAN specific description

This chapter describes additions to the CAN definition of FrameTriggerings.

[TPS_SYST_01130] Communication over CAN ⌈The System Template supports the description of communication over CAN.⌋

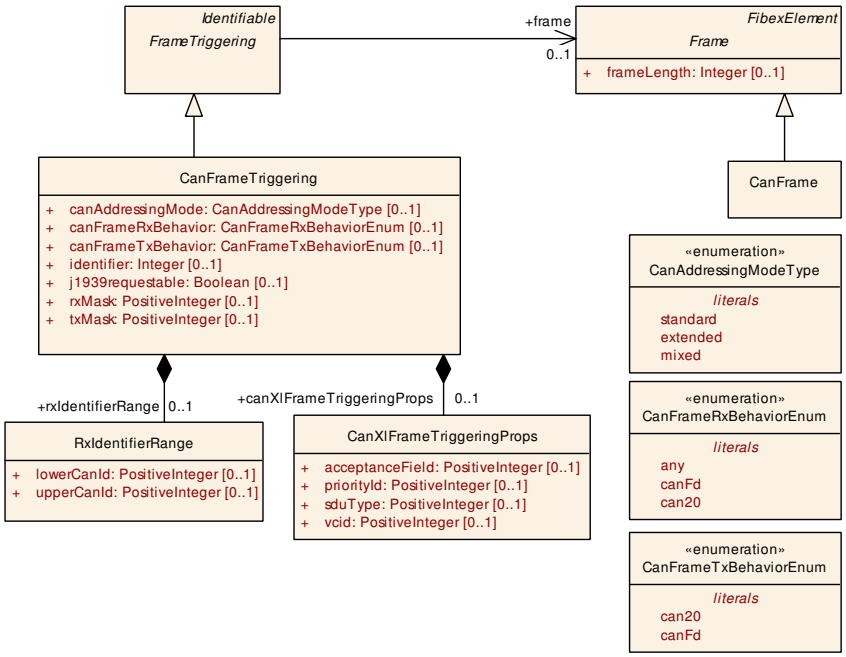

Figure 6.39: CanFrameTriggering (Fibex4Can:CanCommunication)

Class	**CanFrame**			
Note	CAN specific Frame element. **Tags:** atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.107: CanFrame

Class	**CanFrameTriggering**			
Note	CAN specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canAddressing Mode	CanAddressingMode Type	0..1	attr	The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.
canFrameRx Behavior	CanFrameRxBehavior Enum	0..1	attr	Defines which CAN protocol shall be expected for frame reception.
canFrameTx Behavior	CanFrameTxBehavior Enum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
canXlFrame TriggeringProps	CanXlFrameTriggering Props	0..1	aggr	Definition of CAN XL specific attributes in case the frame is a CAN XL frame. **Tags:** atp.Splitkey=canXlFrameTriggeringProps
identifier	Integer	0..1	attr	This attribute is used to define the identifier this frame shall use on the CAN network.
j1939requestable	Boolean	0..1	attr	Frame can be triggered by the J1939 request message.
rxIdentifier Range	RxIdentifierRange	0..1	aggr	Optional definition of a CanId range. **Tags:** atp.Splitkey=rxIdentifierRange
rxMask	PositiveInteger	0..1	attr	Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range.
txMask	PositiveInteger	0..1	attr	Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.

Table 6.108: CanFrameTriggering

[constr_9100] Existence of `CanFrameTriggering.canAddressingMode`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanFrameTriggering`, the attribute `canAddressingMode` shall exist.⌋

Enumeration	**CanAddressingModeType**
Note	Indicates whether standard or extended CAN identifiers are used
Aggregated by	CanFrameTriggering.canAddressingMode, IEEE1722TpAcfCanPart.canAddressingMode
Literal	***Description***
extended	Extended 29-bit-identifiers are used (CAN 2.0B) **Tags:** atp.EnumerationLiteralIndex=0
mixed	The type of CANIDs can be both Standard or Extended. **Tags:** atp.EnumerationLiteralIndex=2
standard	Standard 11-bit-identifiers are used (CAN 2.0A) **Tags:** atp.EnumerationLiteralIndex=1

Table 6.109: CanAddressingModeType

Class	**RxIdentifierRange**			
Note	Optional definition of a CanId range to reduce the effort of specifying every possible FrameTriggering within the defined Id range during reception. All frames received within a range are mapped to the same Pdu that is passed to a upper layer module (e.g. Nm, CDD, PduR).			
Base	*ARObject*			
Aggregated by	CanFrameTriggering.rxIdentifierRange, CanXlNmNodeProps.rxIdentifierRange, IEEE1722TpAcfCanPart.canIdentifierRange			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lowerCanId	PositiveInteger	0..1	attr	This attribute can be used together with the upperCanId attribute to define a range of CanIds.
upperCanId	PositiveInteger	0..1	attr	This attribute can be used together with the lowerCanId attribute to define a range of CanIds.

Table 6.110: RxIdentifierRange

[constr_9101] Existence of RxIdentifierRange.lowerCanId

Imposition time: CP: IT_SysDesc

⌈For each RxIdentifierRange, the attribute lowerCanId shall exist.⌋

[constr_9102] Existence of RxIdentifierRange.upperCanId

Imposition time: CP: IT_SysDesc

⌈For each RxIdentifierRange, the attribute upperCanId shall exist.⌋

Enumeration	**CanFrameRxBehaviorEnum**
Note	Defines different CAN protocols for frame reception behavior.
Aggregated by	CanFrameTriggering.canFrameRxBehavior
Literal	***Description***
any	This CAN frame may be received as both, CAN 2.0 and CAN FD. **Tags:** atp.EnumerationLiteralIndex=0
can20	This CAN frame shall be received as CAN 2.0 only. In case the CAN frame is received as CAN FD it is discarded during reception. **Tags:** atp.EnumerationLiteralIndex=1
canFd	This CAN frame shall be received as CAN FD only. In case the CAN frame is received as CAN 2.0 it is discarded during reception. **Tags:** atp.EnumerationLiteralIndex=2

Table 6.111: CanFrameRxBehaviorEnum

There exist use-cases where the CanFrameTriggering is used as a placeholder for a variant number of actual Can frames and therefore no dedicated CAN identifier can be defined (e.g. MetaData handling, Bus Mirroring).

[TPS_SYST_02201] Existence of CanFrameTriggering.identifier ⌈In a System with category SYSTEM_DESCRIPTION the identifier may be omitted if the value is computed during runtime. In a System with category ECU_EXTRACT, ECU_SYSTEM_DESCRIPTION or SYSTEM_EXTRACT for the transmitter the identifier may be omitted if the value is computed during runtime. In an System with category ECU_EXTRACT, ECU_SYSTEM_DESCRIPTION or SYSTEM_EXTRACT for the receiver the identifier may be omitted if rxIdentifierRange is defined.⌋

[constr_9318] Reception of CanFrameTriggerings with the same identifier by an EcuInstance

Imposition time: CP: IT_SysDesc

⌈For all CanFrameTriggerings on the same PhysicalChannel that refer to a FramePort with the communicationDirection = in of the same EcuInstance the condition applies that no two of these CanFrameTriggerings shall have the same identifier and the same canAddressingMode assigned.⌋

In other words an EcuInstance shall not receive two CanFrameTriggerings with the same identifier and the same canAddressingMode in a variant bound model. This constraint was introduced to the System Template to detect misconfiguration issues in the CanIf module in an early development phase.

[TPS_SYST_02256] Allowed CanFrame.frameLength settings ⌈For a CanFrame it is allowed to configure a smaller frameLength than the length of the Pdu which is mapped to this CanFrame. This is used to model the minimum length of the received L-PDU of a CanFrame to be accepted, if the data length check is enabled.⌋

The CanFrameTriggering.canFrameRxBehavior allows to define a tolerant CAN FD reception strategy. With the setting any the respective CAN frame is accepted for reception, regardless whether it is received with CAN FD or CAN 2.0 protocol.

Enumeration	**CanFrameTxBehaviorEnum**
Note	Defines different CAN protocols for frame transmission behavior.
Aggregated by	CanFrameTriggering.canFrameTxBehavior, IEEE1722TpAcfCanPart.canFrameTxBehavior
Literal	*Description*
can20	This CAN frame shall be sent as CAN 2.0 only. **Tags:** atp.EnumerationLiteralIndex=0
canFd	This CAN frame shall be sent as CAN FD. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.112: CanFrameTxBehaviorEnum

Note that the transmission behavior of CanFrameTriggering.canFrameTxBehavior may still be redefined in the communication stack on driver level.

[TPS_SYST_02168] MetaData support required if CanFrameTriggering.txMask is used ⌈The usage of CanFrameTriggering.txMask requires the support of COM Stack MetaData.⌋

Please note that the MetaData support in [TPS_SYST_02168] is required to calculate CAN-Ids at run-time.

[TPS_SYST_02169] MetaData support may be required if CanFrameTriggering.rxMask is used ⌈The usage of CanFrameTriggering.rxMask may require the support of COM Stack MetaData.⌋

Please note that the MetaData support in [TPS_SYST_02169] is required if the upper layer is interested in the masked part of CAN-Id, e.g. J1939. In some cases the upper layer is not interested in the masked part of CAN-Id, e.g. for CanNm the MetaData is not required.

6.7.3.1 SAE J1939 Protocol specific description

J1939 is a protocol and application layer standard of the SAE (Society of Automotive Engineers) based on the CAN technology. It defines parameters uniquely identified by the SPN (Suspect Parameter Number). These are mapped to parameter groups that are uniquely identified by a PGN (Parameter Group Number). Parameters are simply handled as `SystemSignal`s which have a name derived from the name of the SPNs. A Parameter Group (PG) corresponds to an `IPdu`.

J1939 uses extended 29 bit CAN identifiers to encode a priority, the source address of the frame, and a frame ID which is based on the PGN (Parameter Group Number) and may contain the destination address.

J1939 supports `IPdu`s with more than 8 bytes, and `IPdu`s with variable length that may exceed 8 bytes. As soon as an `IPdu` has more than 8 bytes, it does not fit in a single CAN frame and a transport protocol shall be used. Variable length `IPdu`s will always be handled by the J1939 TP, regardless of the actual length. The J1939 Transport Protocol is described in chapter 6.8.7.

[TPS_SYST_01132] Communication over SAE J1939 ⌈The System Template supports the description of communication over SAE J1939.⌋

[constr_3209] `CanFrameTriggering`s with identical PGN

Imposition time: CP: `IT_SysDesc`

⌈For all `CanFrameTriggering`s where the attribute `identifier` contains the identical PGN (as defined in section 5.2 Protocol Data Unit in [27]) the attribute `j1939requestable` shall also have an identical value.⌋

6.7.3.2 CAN XL specific description

CAN XL is a continued development of CAN FD and standardized by the *Special Interest Group (SIG) CAN XL* of the *CAN in Automation (CiA)* association. It features a transmission speed of up to 20 MBit/s and large payloads (up to 2048 bytes) to offer a cost-effective alternative to Ethernet 10BASE network technology.

To ease the migration of an Ethernet based communication design to CAN XL, the feature of tunneling Ethernet frames through CAN XL based physical connections has also been standardized by the CiA association. The required system-level configuration for this use case is described in Ch. 3.3.1.2.2.

[TPS_SYST_03075] Communication over CAN XL ⌈The System Template supports the description of communication over CAN XL.⌋

Since CAN XL requires definition of additional specific configuration parameters on the physical layer which are relevant on system level, a few extensions to some elements of the System Template have been defined. They are described in the following.

[TPS_SYST_03076] Definition of CAN XL frame triggering attributes ⌈If a `CanFrameTriggering` relates to a CAN XL frame, then `canXlFrameTriggeringProps` as well as the relevant attributes contained in the aggregated class `CanXlFrameTriggeringProps` shall be defined accordingly. The absence of `CanXlFrameTriggeringProps` indicates a `CanFrameTriggering` for a non-CAN-XL frame.⌋

Please note that the relevance of the attributes of `CanXlFrameTriggeringProps` depends on the SDU Type and is specified by the *CiA* standardization of CAN XL. Additionally, a `CanXlFrameTriggeringProps` defined without any attributes is also valid meaning that the associated CAN frame is a CAN XL frame and gets these attribute values by meta-data. The existence of corresponding meta-data configuration has to be ensured during runtime.

[constr_3704] Existence of `CanXlFrameTriggeringProps`

Imposition time: CP: `IT_SysDesc`

⌈If the class `CanXlFrameTriggeringProps` is aggregated by a `CanFrameTriggering`, then the `CanCommunicationController` – which is referenced through `commController` by a `CanCommunicationConnector` which in turn is referenced through `commConnector` by a `CanPhysicalChannel` that aggregates the aforementioned `CanFrameTriggering` – shall aggregate at least one of

- `CanControllerConfiguration` with `CanControllerXlConfiguration` aggregated or
- `CanControllerConfigurationRequirements` with `CanControllerXlConfigurationRequirements` aggregated.

⌋

[constr_3713] Allowed values for `acceptanceField`

Imposition time: CP: `IT_SysDesc`

⌈The value for `acceptanceField` shall be in the range between 0 and 4294967295.⌋

[constr_3705] Allowed values for `priorityId`

Imposition time: CP: `IT_SysDesc`

⌈The value for `priorityId` shall be in the range between 0 and 2047.⌋

[constr_3706] Allowed values for `sduType`

Imposition time: CP: `IT_SysDesc`

⌈The value for `sduType` shall be in the range between 0 and 255.⌋

[constr_3707] Allowed values for `vcid`

Imposition time: CP: `IT_SysDesc`

⌈The value for `vcid` shall be in the range between 0 and 255.⌋

Please note: Since it is possible to concurrently operate CAN FD and CAN XL nodes in the same network, the attributes of `CanFrameTriggering` related to CAN FD are also relevant in case this mixed network use case is to be realized.

Please note that figure 6.39 shows the modeling of `CanXlFrameTriggeringProps`.

6.7.4 Ethernet specific description

This chapter specifies how the data communication between nodes in an IP network over TCP and UDP protocols is described with an AUTOSAR model.

[TPS_SYST_01091] Definition of SoAdConfig ⌈The SoAdConfig in the System Template is defined per EthernetPhysicalChannel which represents a VLAN.⌋

The SoAdConfig element is the entry point for the description of the IP communication on a VLAN since it contains a collection of SocketAddresses of nodes that are able to receive and transmit information over the VLAN. Each node is represented by an EcuInstance. The SocketAddress defines a communication endpoint (IP Unicast or IP Multicast) and assigns it to one EthernetCommunicationConnector of an EcuInstance in case of IP Unicast communication and to one or several EthernetCommunicationConnectors in case of IP Multicast.

Class	**SoAdConfig**			
Note	SoAd Configuration for one specific Physical Channel.			
Base	*ARObject*			
Aggregated by	EthernetPhysicalChannel.soAdConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
socketAddress	SocketAddress	*	aggr	Collection of SoAdAddresses. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=socketAddress, socketAddress.shortName, socketAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.113: SoAdConfig

The SocketAddress is the element that is used to establish the link between the EcuInstance and the NetworkEndpoint. The SocketAddress has a connector reference to the EthernetCommunicationConnector. The SocketAddress also aggregates the ApplicationEndpoint that in turn references the NetworkEndpoint.

Class	**SocketAddress**			
Note	This meta-class represents a socket address towards the rest of the meta-model. The actual semantics of the represented socket address, however, is contributed by aggregation of an ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SoAdConfig.socketAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowedIPv6Ext Headers	IPv6ExtHeaderFilterList	0..1	ref	Reference to a list of IPv6 Extension Headers allowed for this SocketConnection. If no list is referenced all IPv6 Extension Headers are allowed and processed. **Tags:** atp.Splitkey=allowedIPv6ExtHeaders

▽

△

Class	**SocketAddress**			
allowedTcp Options	TcpOptionFilterList	0..1	ref	Reference to a list of TCP options allowed for this Socket Connection. **Tags:** atp.Splitkey=allowedTcpOptions This Attribute is only used by the AUTOSAR Classic Platform.
application Endpoint	ApplicationEndpoint	0..1	aggr	Application addressing **Tags:** atp.Splitkey=applicationEndpoint, application Endpoint.shortName
connector	EthernetCommunication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP unicast address for an ECU that is part of the model. **Tags:** atp.Splitkey=connector
differentiated ServiceField	PositiveInteger	0..1	attr	The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.
flowLabel	PositiveInteger	0..1	attr	The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.
multicast Connector	EthernetCommunication Connector	*	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP multicast address, i.e. if the aggregated ApplicationEndpoint references a NetworkEndpoint that describes an IP Address in the IP multicast range. Such a SocketAddress contains references to those Ecus (via the multicastConnector reference) in the model that will receive multicast messages via the SocketAddress that is defined by the aggregated ApplicationEndpoint and NetworkEndpoint, i.e. IP Address and UDP Port combination. **Tags:** atp.Splitkey=multicastConnector
pduCollection MaxBufferSize	PositiveInteger	0..1	attr	Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.
pduCollection Timeout	TimeValue	0..1	attr	Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.
staticSocket Connection	StaticSocketConnection	*	aggr	Definition of a static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticSocketConnection, staticSocket Connection.shortName, staticSocketConnection.variation Point.shortLabel vh.latestBindingTime=postBuild
ttl	PositiveInteger	0..1	attr	This attribute defines a value set in the header of an Internet Protocol (IP) packet that tells network devices the maximum number of router hops the packet can make before it is discarded. The TTL value is a counter that is decremented by 1 every time the packet passes through a router.

▽

△

Class	**SocketAddress**			
udpChecksum Handling	UdpChecksum CalculationEnum	0..1	attr	Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksum Disabled) on the related socket connection.

Table 6.114: SocketAddress

The communication endpoint itself is defined by the ApplicationEndpoint that is aggregated by the SocketAddress and by the NetworkEndpoint that in turn is referenced by the ApplicationEndpoint and is defined on the VLAN. The ApplicationEndpoint is the endpoint in terms of application addressing and defines the transport layer configuration. The IP-address that is connected to the transport layer is defined by the referenced NetworkEndpoint.

[constr_5061] EthernetCommunicationConnectors and referencing SocketAddresses shall be in the same VLAN

Imposition time: CP: IT_SysDesc

⌈Each EthernetCommunicationConnector that is referenced by a SocketAddress in the role connector or multicastConnector shall be referenced by the same EthernetPhysicalChannel that aggregates the SoAdConfig that in turn aggregates the SocketAddress.⌋

[constr_5326] Each local SocketAddress of an EcuInstance shall reference an EthernetCommunicationConnector in the role connector or multicastConnector

Imposition time: CP: IT_SysDesc

⌈If an EcuInstance uses a SocketAddress as local address, the SocketAddress shall refer to an EthernetCommunicationConnector of the EcuInstance, either via SocketAddress.connector if the SocketAddress represents a unicast address, or via SocketAddress.multicastConnector if the SocketAddress represents a multicast address.⌋

[constr_3311] Usage of SocketAddress.flowLabel

Imposition time: CP: IT_SysDesc

⌈SocketAddress.flowLabel shall only be used if the aggregated ApplicationEndpoint refers to a NetworkEndpoint with an Ipv6Configuration.⌋

[TPS_SYST_02140] SocketAddress.udpChecksumHandling default value ⌈If SocketAddress.udpChecksumHandling is not used the value udpChecksumEnabled shall be assumed.⌋

Enumeration	**UdpChecksumCalculationEnum**
Note	This enumeration defines the UDP checksum calculation.
Aggregated by	SocketAddress.udpChecksumHandling
Literal	***Description***
udpChecksum Disabled	Udp checksum handling shall be disabled **Tags:** atp.EnumerationLiteralIndex=1
udpChecksum Enabled	Udp checksum handling shall be enabled **Tags:** atp.EnumerationLiteralIndex=0

Table 6.115: UdpChecksumCalculationEnum

[TPS_SYST_02141] Semantics of `udpChecksumHandling` ⌈The semantics of `udpChecksumHandling` is different for the sending and the receiving side:

TX - calculation of UDP checksum:

- `udpChecksumEnabled` means that the UDP checksum is calculated on the transmission side.
- `udpChecksumDisabled` means that the UDP checksum is not calculated but set to zero on the transmission side.

RX - handling of UDP checksum of zero:

- `udpChecksumEnabled` means that the UDP checksum of zero is treated as invalid checksum on receiver side (causing the UDP datagram to be dropped by the receiver). A valid non-zero checksum is accepted and the UDP datagram is forwarded to the upper layer.
- `udpChecksumDisabled` means the UDP checksum of zero is treated as valid checksum on the receiver side (causing the UDP datagram to be forwarded to the upper layer). A valid non-zero checksum is accepted and the UDP datagram is forwarded to the upper layer as well.

⌋

[TPS_SYST_02142] Reception of invalid checksum ⌈On Rx side an invalid checksum should always cause the related UDP datagram to be discarded independent of the `udpChecksumHandling` value.⌋

To enable the IPv6 packet filtering the attribute `allowedIPv6ExtHeaders` allows to define a permitted list of IPv6 Extension Headers that are allowed for a `SocketAddress`. Lists of IPv6 Extension Headers can be defined with the `IPv6ExtHeaderFilterList` element and can be collected in `IPv6ExtHeaderFilterSet`s.

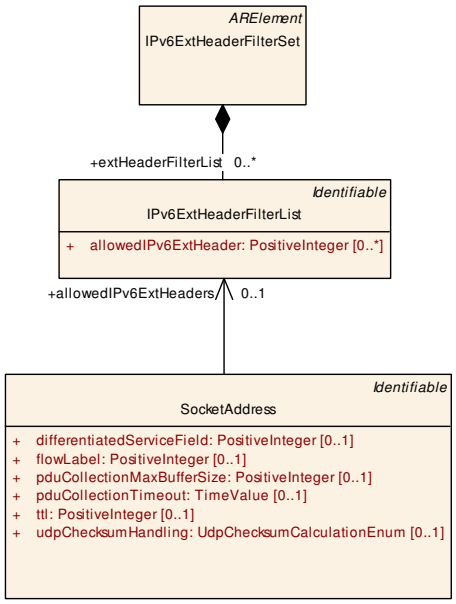

Figure 6.40: IPv6 Extension Header Filter Set

Class	**IPv6ExtHeaderFilterSet**			
Note	Set of IPv6 Extension Header Filters. **Tags:** atp.recommendedPackage=IPv6ExtHeaderFilterSets This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
extHeaderFilter List	IPv6ExtHeaderFilterList	*	aggr	In order to permit or deny certain types of IPv6 extension headers a permitted list of IPv6 extension headers can be configured. **Tags:** atp.Splitkey=extHeaderFilterList, extHeaderFilter List.shortName

Table 6.116: IPv6ExtHeaderFilterSet

Class	**IPv6ExtHeaderFilterList**			
Note	Permitted list for the filtering of IPv6 extension headers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPv6ExtHeaderFilterSet.extHeaderFilterList			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowedIPv6Ext Header	PositiveInteger	*	attr	IPv6 Extension Header type allowed by this filter.

Table 6.117: IPv6ExtHeaderFilterList

[constr_3276] Prohibition of usage of `allowedIPv6ExtHeaders` in IPv4 `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈IPv4 `SocketAddress` shall not define `allowedIPv6ExtHeaders`. An IPv4 `SocketAddress` aggregates an `ApplicationEndpoint` that refers to a `NetworkEndpoint` that has an `Ipv4Configuration` as `networkEndpointAddress`.⌋

[constr_3277] Restriction of usage of `IPv6ExtHeaderFilterList`s in IPv6 `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈All `SocketAddress`es related to the same IPv6 `NetworkEndpoint` shall all reference either no or exactly the same `IPv6ExtHeaderFilterList` with the `allowedIPv6ExtHeaders` attribute.⌋

[constr_9109] Existence of `IPv6ExtHeaderFilterList.allowedIPv6ExtHeader`

Imposition time: CP: `IT_SysDesc`

⌈For each `IPv6ExtHeaderFilterList`, the attribute `allowedIPv6ExtHeader` shall exist.⌋

To enable the filtering of Tcp options the attribute `allowedTcpOptions` defines a permitted list of Tcp options that are allowed for a `SocketAddress`. Lists of Tcp Option filters can be defined with the `TcpOptionFilterList` element and can be collected in `TcpOptionFilterSet`s.

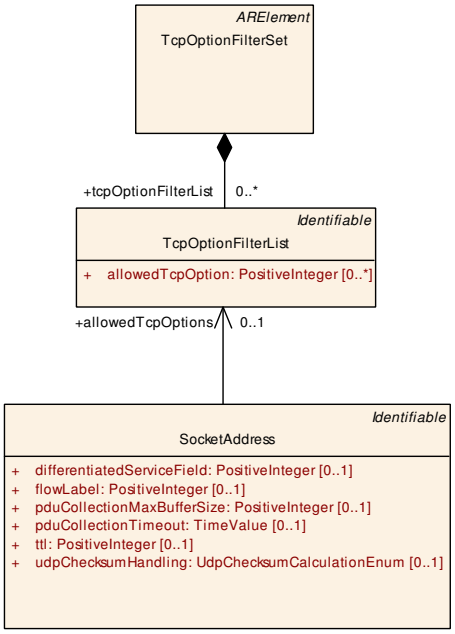

Figure 6.41: Tcp Option Filter Set

Class	TcpOptionFilterSet			
Note	Set of TcpOptionFilterLists. **Tags:** atp.recommendedPackage=TcpOptionFilterSets This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpOptionFilter List	TcpOptionFilterList	*	aggr	Collection of permitted lists for the filtering of TCP options. **Tags:** atp.Splitkey=tcpOptionFilterList, tcpOptionFilter List.shortName

Table 6.118: TcpOptionFilterSet

Class	TcpOptionFilterList			
Note	Permitted list for the filtering of TCP options. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TcpOptionFilterSet.tcpOptionFilterList			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allowedTcp Option	PositiveInteger	*	attr	TCP option kind allowed by this filter.

Table 6.119: TcpOptionFilterList

[constr_9110] Existence of `TcpOptionFilterList.allowedTcpOption`

Imposition time: CP: `IT_SysDesc`

⌈For each `TcpOptionFilterList`, the attribute `allowedTcpOption` shall exist.⌋

[constr_3297] Prohibition of usage of `allowedTcpOptions` in Udp `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈Udp `SocketAddress` shall not define `allowedTcpOptions`. A Udp `SocketAddress` aggregates an `ApplicationEndpoint` that has a `UdpTp` defined as `tpConfiguration`.⌋

6.7.4.1 ApplicationEndpoint

Class	ApplicationEndpoint			
Note	An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SocketAddress.applicationEndpoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**ApplicationEndpoint**			
maxNumberOf Connections	PositiveInteger	0..1	attr	This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.
network Endpoint	NetworkEndpoint	0..1	ref	Reference to the network address. **Tags:** atp.Splitkey=networkEndpoint
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
tlsCrypto Mapping	TlsCryptoService Mapping	0..1	ref	This reference identifies the applicable TlsCryptoService Mapping that adds the ability for TLS-based encryption on the enclosing ApplicationEndpoint. **Tags:** atp.Splitkey=tlsCryptoMapping
tpConfiguration	TransportProtocol Configuration	0..1	aggr	Configuration of the used transport protocol. **Tags:** atp.Splitkey=tpConfiguration

Table 6.120: ApplicationEndpoint

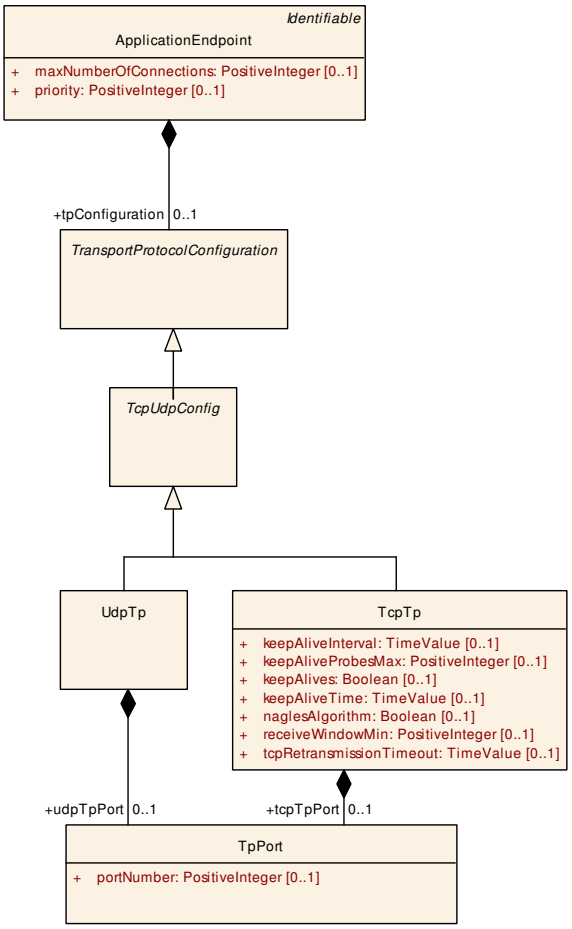

Figure 6.42: Application Endpoint

Class	**TransportProtocolConfiguration** (abstract)			
Note	Transport Protocol configuration.			
Base	*ARObject*			
Subclasses	*TcpUdpConfig*			
Aggregated by	ApplicationEndpoint.tpConfiguration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.121: TransportProtocolConfiguration

The following Transport Protocols are supported by the System Template:

Class	**TcpUdpConfig** (abstract)			
Note	Tcp or Udp Transport Protocol Configuration.			
Base	*ARObject*, *TransportProtocolConfiguration*			
Subclasses	TcpTp, UdpTp			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.122: TcpUdpConfig

Class	**UdpTp**			
Note	Content Model for UDP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
udpTpPort	TpPort	0..1	aggr	Udp Port configuration. **Tags:** atp.Splitkey=udpTpPort

Table 6.123: UdpTp

Class	**TcpTp**			
Note	Content Model for TCP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
keepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
keepAlive ProbesMax	PositiveInteger	0..1	attr	Maximum number of times that TCP retransmits an individual data segment before aborting the connection.
keepAlives	Boolean	0..1	attr	Indicates if Keep-Alive messages are sent.
keepAliveTime	TimeValue	0..1	attr	Specifies the time in seconds between the last data packet sent and the first keepalive probe.
naglesAlgorithm	Boolean	0..1	attr	Indicates if Nagle's Algorithm is used.
receiveWindow Min	PositiveInteger	0..1	attr	Minimum size of the TCP receive window in bytes.



△

Class	**TcpTp**			
tcp Retransmission Timeout	TimeValue	0..1	attr	Defines the timeout in seconds before an unacknowledged TCP segment is sent again. If the tcp RetransmissionTimeout is not defined or set to "INF", no TCP segments shall be re-transmitted.
tcpTpPort	TpPort	0..1	aggr	TCP Port configuration. **Tags:** atp.Splitkey=tcpTpPort

Table 6.124: TcpTp

Class	**TpPort**			
Note	Dynamic or direct assignment of a PortNumber.			
Base	*ARObject*			
Aggregated by	TcpTp.tcpTpPort, UdpTp.udpTpPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
portNumber	PositiveInteger	0..1	attr	Port Number.

Table 6.125: TpPort

For the usable value range of `portNumber`, restrictions may be imposed on the members of an `EthernetCluster`. If all the participants of the `EthernetCluster` are fully statically configured with respect to their used `portNumber`s, no need for reserved `portNumber` ranges may exist.

But with the introduction of the AUTOSAR Adaptive Platform, more dynamic communication use-cases and protocols have been introduced. Some `portNumber` ranges might have a dedicated usage in these scenarios.

An operating system implementation might utilize a certain set of `portNumber`s, assuming these `portNumber`s are free to use in an ephemeral way. If now an AUTOSAR Service defines a Service Instance to use a `portNumber` that has already been allocated by the operating system (maybe already during boot), then the AUTOSAR Service Instance will not be able to be instantiated.

To avoid clashing `portNumber`s, special care has to be put into defining the usable `portNumber` ranges for AUTOSAR communication. The source of the restrictions on the usage of `portNumber`s is [28, IETF RFC 6335], especially section 8.1.2 “Variances for Specific Port Number Ranges”.

[advisory_02005] Usable `portNumber`s for AUTOSAR Communication

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈It is advised not to use the following values for `portNumber`:

- 1023, 1024, 49151: reserved by IANA
- 49152 .. 65535: this is the Dynamic Port range and might be allocated by the operating system or other applications to be used as ephemeral ports

⌋

[TPS_SYST_02215] Usage of portNumber with value 0 on a remote port ⌈The configuration of the portNumber to 0 on a remote port means that the portNumber is assigned dynamically at runtime.⌋

[TPS_SYST_02443] Usage of portNumber with value 0 on a local port ⌈The configuration of the portNumber to 0 on a local port means that the portNumber is an ephemeral port and is assigned dynamically at runtime by the IP-stack.⌋

In the following cases the portNumber on a local port on an ApplicationEndpoint can be set to value = 0:

1. ProvidedServiceInstance.localUnicastAddress: In this case the Service is provided on an SocketAddress that includes the ephemeral port number and is used by the Service Discovery in the offer Message.
2. ConsumedServiceInstance.localUnicastAddress: In this case the Service is consumed on an SocketAddress that includes the ephemeral port number and is used by the Service Discovery in the SubscribeEventGroup Message.
3. ConsumedEventGroup.eventMulticastAddress: If the multicast address is determined at runtime, the portNumber = 0 indicates that the portNumber will be reconfigured at runtime as well, together with the multicast address, according to the values received from the service provider via Service Discovery.

In the following cases the portNumber on a remote port on an ApplicationEndpoint can be set to value = 0:

1. ProvidedServiceInstance.remoteUnicastAddress: The portNumber = 0 indicates that clients can access the service using arbitrary client port numbers. The concrete portNumber will be set at runtime.
2. EventHandler.eventMulticastAddress: In this case the Multicast Address that is used for event communication in the IP-Multicast case includes the ephemeral port number that is used by the Service Discovery in the SubscribeEventGroupAck Message. This is the address to which the EventHandler sends the events to in case IP Multicast is used.

[advisory_03005] ConsumedServiceInstance.remoteUnicastAddress should not reference an ApplicationEndpoint with portNumber = 0

Imposition time: CP: IT_SysDesc

⌈ConsumedServiceInstance should not reference an ApplicationEndpoint in the role remoteUnicastAddress that defines a UdpTp or TcpTp tpConfiguration with portNumber = 0.⌋

The reason for [advisory_03005] is that according to [TPS_SYST_02396] the Service Offer and Service Find do not need to be performed if ConsumedServiceInstance.remoteUnicastAddress is used. But the ConsumedServiceInstance needs to know the address of the Service provider since the methods of this Service can be called.

[TPS_SYST_01131] TCP/IP and UDP/IP communication over Ethernet ⌈The System Template supports the description of TCP/IP and UDP/IP communication over Ethernet.⌋

[TPS_SYST_01089] `ApplicationEndpoint` priority ⌈The `priority` at the `ApplicationEndpoint` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership` and the `priority` that is defined at the `NetworkEndpoint`.⌋

[constr_9111] Existence of `ApplicationEndpoint.networkEndpoint`

Imposition time: CP: `IT_SysDesc`

⌈For each `ApplicationEndpoint`, the reference to `NetworkEndpoint` in the role `networkEndpoint` shall exist.⌋

[constr_9367] Definition of several local `ApplicationEndpoint`s with identical properties (IP, Port, Protocol) is forbidden

Imposition time: CP: `IT_SysDesc`

⌈It is not allowed to define several local `ApplicationEndpoint`s (referenced by `ProvidedServiceInstance`s in the role `localUnicastAddress` or by `ConsumedServiceInstance`s in the role `localUnicastAddress`) that use the same port number and protocol via `tpConfiguration` and reference the same `NetworkEndpoint` in the role `networkEndpoint` in the context of an `EthernetPhysicalChannel`.⌋

[constr_9367] was introduced to forbid scenarios in which local `ApplicationEndpoint`s are duplicated because of the `maxNumberOfConnections` attribute setting. If several Service Instances (`ProvidedServiceInstance`s or `ConsumedServiceInstance`s) are located on the same `SocketAddress` (defined by `ApplicationEndpoint` and `NetworkEndpoint`), then all those Service Instances shall reference the same `ApplicationEndpoint`. The `maxNumberOfConnections` attribute of the local `ApplicationEndpoint` defines the maximum number of connections shared by all Service Instances of this `ApplicationEndpoint`.

If for example ProvidedServiceInstance_1 has two potential client `EcuInstance`s and ProvidedServiceInstance_2 has three different potential client `EcuInstance`s then the `maxNumberOfConnections` attribute of the `ApplicationEndpoint` needs to be set to 5.

[constr_9113] Existence of `UdpTp.udpTpPort`

Imposition time: CP: `IT_SysDesc`

⌈For each `UdpTp`, the aggregation of `TpPort` in the role `udpTpPort` shall exist.⌋

[constr_9114] Existence of TcpTp.tcpTpPort

Imposition time: CP: IT_SysDesc

⌈For each TcpTp, the aggregation of TpPort in the role tcpTpPort shall exist.⌋

6.7.4.2 NetworkEndpoint

Class	**NetworkEndpoint**			
Note	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetPhysicalChannel.networkEndpoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
fullyQualified DomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.
infrastructure Services	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed. **Tags:** atp.Splitkey=infrastructureServices This Attribute is only used by the AUTOSAR Classic Platform.
ipSecConfig	IPSecConfig	0..1	aggr	Optional IPSec configuration that provides security services for IP packets. **Tags:** atp.Splitkey=ipSecConfig
network Endpoint Address	NetworkEndpoint Address	*	aggr	Definition of a Network Address. **Tags:** atp.Splitkey=networkEndpointAddress, networkEndpoint Address.ipv4Address, networkEndpointAddress.ipv6 Address, networkEndpointAddress.macMulticastGroup xml.namePlural=NETWORK-ENDPOINT-ADDRESSES
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.

Table 6.126: NetworkEndpoint

The NetworkEndpoint defines the network addressing. The network endpoint may have a priority and a FQDN (Fully Qualified Domain Name) that is used for the Service Discovery (e.g. some.example.host.).

[constr_9122] Existence of NetworkEndpoint.networkEndpointAddress

Imposition time: CP: IT_SysDesc

⌈For each NetworkEndpoint, the aggregation of NetworkEndpointAddress in the role networkEndpointAddress shall exist.⌋

[TPS_SYST_01090] valid NetworkEndpoint ⌈To build a valid NetworkEndpoint at least one of the following options shall be defined in the role NetworkEndpoint.networkEndpointAddress:

- a MacMulticastConfiguration with a reference to a MacMulticastGroup
- Ipv4Configuration
- Ipv6Configuration

⌋

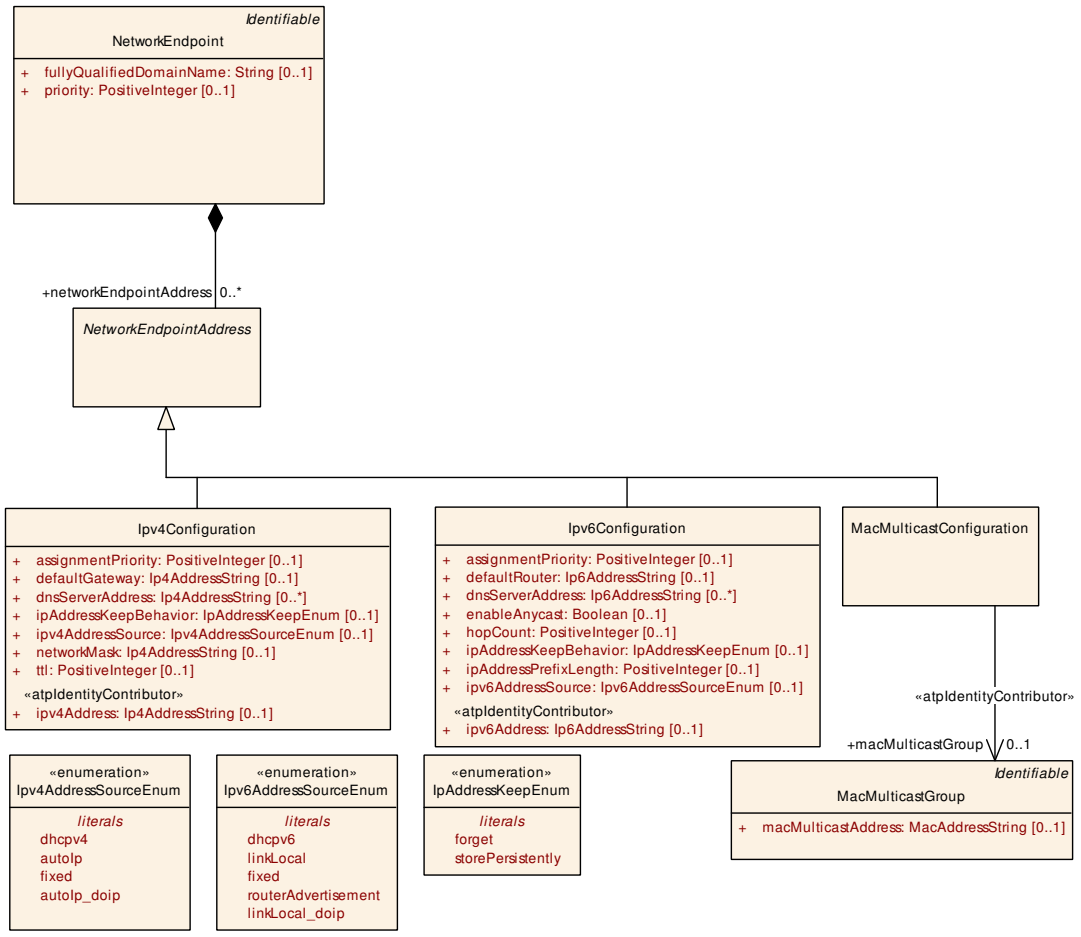

Figure 6.43: Network Endpoint

The attribute `NetworkEndpoint.networkEndpointAddress` defines whether an **IPv4**, **IPv6** or **MAC multicast** address is assigned to the `NetworkEndpoint`.

The reference of the `MacMulticastConfiguration.macMulticastGroup` defines the mapping of IP multicast to MAC multicast.

[TPS_SYST_01088] `NetworkEndpoint` priority ⌈The `priority` at the `NetworkEndpoint` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership`.⌋

Class	***NetworkEndpointAddress*** (abstract)
Note	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.
Base	*ARObject*
Subclasses	Ipv4Configuration, Ipv6Configuration, MacMulticastConfiguration

▽

△

Class	***NetworkEndpointAddress*** (abstract)			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.127: NetworkEndpointAddress

Class	**Ipv4Configuration**			
Note	Internet Protocol version 4 (IPv4) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipv4Address	Ip4AddressString	0..1	attr	IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used. **Stereotypes:** atpIdentityContributor
ipv4Address Source	Ipv4AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.
networkMask	Ip4AddressString	0..1	attr	Network mask. Notation 255.255.255.255
ttl	PositiveInteger	0..1	attr	Lifespan of data (0..255). The purpose of the TimeToLive field is to avoid a situation in which an undeliverable datagram keeps circulating on a system.

Table 6.128: Ipv4Configuration

Enumeration	**Ipv4AddressSourceEnum**
Note	Defines how the node obtains its IPv4-Address.
Aggregated by	Ipv4Configuration.ipv4AddressSource
Literal	***Description***
autoIp	AutoIP is used to dynamically assign IP addresses at device startup. **Tags:** atp.EnumerationLiteralIndex=0
autoIp_doip	Linklocal IPv4 Address Assignment using DoIP Parameters **Tags:** atp.EnumerationLiteralIndex=2 xml.name=AUTO-IP--DOIP
dhcpv4	DHCP is a service for the automatic IP configuration of a client. **Tags:** atp.EnumerationLiteralIndex=3
fixed	The IP Address shall be declared manually. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.129: Ipv4AddressSourceEnum

Enumeration	**IpAddressKeepEnum**
Note	Defines the behavior after a dynamic IP address has been assigned.
Aggregated by	Ipv4Configuration.ipAddressKeepBehavior, Ipv6Configuration.ipAddressKeepBehavior
Literal	***Description***
forget	After a dynamic IP address has been assigned just use it for this session. **Tags:** atp.EnumerationLiteralIndex=0
storePersistently	After a dynamic IP address has been assigned store the address persistently. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.130: IpAddressKeepEnum

Class	**Ipv6Configuration**			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultRouter	Ip6AddressString	0..1	attr	IP address of the default router.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of pre configured DNS servers. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
enableAnycast	Boolean	0..1	attr	This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).
hopCount	PositiveInteger	0..1	attr	The distance between two hosts. The hop count n means that n gateways separate the source host from the destination host (Range 0..255)
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipAddressPrefix Length	PositiveInteger	0..1	attr	IPv6 prefix length defines the part of the IPv6 address that is the network prefix.
ipv6Address	Ip6AddressString	0..1	attr	IPv6 Address. Notation: FFFF:...:FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used. **Stereotypes:** atpIdentityContributor
ipv6Address Source	Ipv6AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.

Table 6.131: Ipv6Configuration

[constr_5263] `NetworkEndpoint.networkEndpointAddress` restriction for IPv4

Imposition time: CP: `IT_SysDesc`

⌈A `NetworkEndpoint` shall not aggregate several `Ipv4Configuration`s that have their `ipv4AddressSource` set to fixed.⌋

[constr_5264] `NetworkEndpoint.networkEndpointAddress` restriction for IPv6

Imposition time: CP: `IT_SysDesc`

⌈A `NetworkEndpoint` shall not aggregate several `Ipv6Configuration`s that have their `ipv6AddressSource` set to fixed.⌋

[constr_5265] `NetworkEndpoint.networkEndpointAddress` restriction

Imposition time: CP: `IT_SysDesc`

⌈A `NetworkEndpoint` shall not aggregate an `Ipv4Configuration` and an `Ipv6Configuration` as `networkEndpointAddress` at the same time.⌋

[TPS_SYST_03002] Keep behavior of DHCP clients ⌈The attribute `IpAddressKeepEnum` defines for the DHCP client to either

- persistently store an assigned IP address (`storePersistently`) after it has been fetched, or
- learn it after each start-up (`forget`).

⌋

[constr_3298] `Ipv6Configuration.ipv6Address` range in case of `enableAnycast`

Imposition time: CP: `IT_SysDesc`

⌈If `Ipv6Configuration.enableAnycast` is set to true then the `Ipv6Configuration.ipv6Address` needs to be in the unicast addressing range.⌋

Enumeration	**Ipv6AddressSourceEnum**
Note	Defines how the node obtains its IPv6-Address.
Aggregated by	Ipv6Configuration.ipv6AddressSource
Literal	***Description***
dhcpv6	DHCP is a service for the automatic IP configuration of a client. **Tags:** atp.EnumerationLiteralIndex=0
fixed	The IP Address shall be declared manually. **Tags:** atp.EnumerationLiteralIndex=1
linkLocal	LinkLocal is intended only for communications within the segment of a local network (a link) or a point-to-point connection that a host is connected to. **Tags:** atp.EnumerationLiteralIndex=2
linkLocal_doip	Linklocal IPv6 Address Assignment using DoIP Parameters **Tags:** atp.EnumerationLiteralIndex=3 xml.name=LINK-LOCAL--DOIP
router Advertisement	IPv6 Stateless Autoconfiguration. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.132: Ipv6AddressSourceEnum

Class	**MacMulticastConfiguration**			
Note	References a per cluster globally defined MAC-Multicast-Group.			
Base	*ARObject*, *NetworkEndpointAddress*			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
macMulticast Group	MacMulticastGroup	0..1	ref	Reference to a macMulticastGroup. **Stereotypes:** atpIdentityContributor

Table 6.133: MacMulticastConfiguration

[constr_9123] Existence of MacMulticastConfiguration.macMulticast-Group

Imposition time: CP: IT_SysDesc

⌈For each MacMulticastConfiguration, the reference to MacMulticastGroup in the role macMulticastGroup shall exist.⌋

Primitive	**Ip4AddressString**
Note	This is used to specify an IP4 address. Notation: 255.255.255.255 **Tags:** xml.xsd.customType=IP4-ADDRESS-STRING xml.xsd.pattern=(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)|ANY xml.xsd.type=string

Table 6.134: Ip4AddressString

Primitive	**Ip6AddressString**
Note	This is used to specify an IP6 address. Notation: FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF Alternative notations, short-cuts with duplicate colons like ::, etc. or mixtures using colons and dots, are not allowed. **Tags:** xml.xsd.customType=IP6-ADDRESS-STRING xml.xsd.pattern=[0-9A-Fa-f]{1,4}(:[0-9A-Fa-f]{1,4}){7,7}|ANY xml.xsd.type=string

Table 6.135: Ip6AddressString

In addition, infrastructure services may be provided or consumed by the Network-Endpoints.

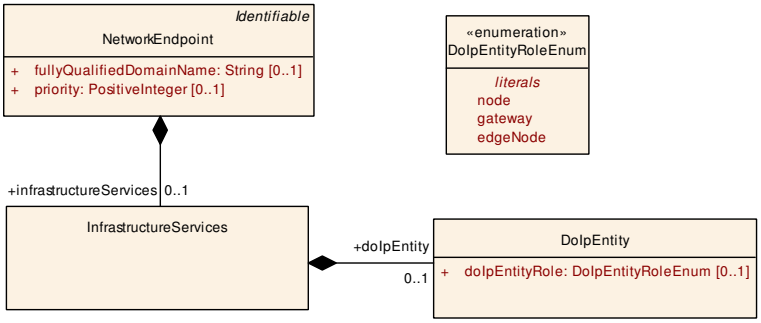

Figure 6.44: Network Endpoint Infrastructure Services

Class	**InfrastructureServices**			
Note	Defines the network infrastructure services provided or consumed.			
Base	*ARObject*			
Aggregated by	NetworkEndpoint.infrastructureServices			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
doIpEntity	DoIpEntity	0..1	aggr	Defines whether a infrastructure service that runs on the network endpoint is a DoIP-Entity. **Tags:** atp.Splitkey=doIpEntity This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.136: InfrastructureServices

The `DoIpEntity` (Diagnostics over Internet Protocol, ISO 13400) defines the DoIp role this `NetworkEndpoint` has.

Class	**DoIpEntity**			
Note	ECU providing this infrastructure service is a DoIP-Entity.			
Base	*ARObject*			
Aggregated by	InfrastructureServices.doIpEntity			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
doIpEntityRole	DoIpEntityRoleEnum	0..1	attr	Identifies the role in terms of DoIP this network-node has.

Table 6.137: DoIpEntity

Enumeration	**DoIpEntityRoleEnum**
Note	DoIP role a network-node has.
Aggregated by	DoIpEntity.doIpEntityRole
Literal	*Description*
edgeNode	Network node is a DoIP gateway that accepts external connections. **Tags:** atp.EnumerationLiteralIndex=0
gateway	Network node is a Gateway between the DoIP network and other networks. **Tags:** atp.EnumerationLiteralIndex=1
node	Network node is a DoIp node. **Tags:** atp.EnumerationLiteralIndex=2

Table 6.138: DoIpEntityRoleEnum

6.7.4.3 Service Instances

The AUTOSAR protocol Service Discovery protocol is used to communicate the availability of functional entities called services in the in-vehicle communication and to control the send behavior of event messages to receivers that subscribed to receive these events (Publish/Subscribe). A Service may also provide methods that a client is able to call (Request/Response).

Please note that the AUTOSAR Classic Platform does not support Service Interfaces. To mimic a Service Interface in the classic platform any combination of ClientServerInterfaces, SenderReceiverInterfaces or TriggerInterfaces may be used to describe a service to which later a Service Id is assigned.

The assignment of Service Ids is done on the Service Instance level in the System Description. An AbstractServiceInstance collected in the ServiceInstanceCollectionSet is describing such a Service Instance.

Class	**ServiceInstanceCollectionSet**			
Note	Collection of ServiceInstances **Tags:** atp.recommendedPackage=ServiceInstanceCollectionSets			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
serviceInstance	AbstractServiceInstance	*	aggr	ServiceInstances that are part of the collection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceInstance, serviceInstance.shortName, serviceInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.139: ServiceInstanceCollectionSet

Class	***AbstractServiceInstance*** (abstract)			
Note	Provided and Consumed Ethernet Service Instances that are available at the ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ConsumedServiceInstance, *DdsCpServiceInstance*, ProvidedServiceInstance			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
capability Record	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=capabilityRecord, capabilityRecord.variationPoint.shortLabel vh.latestBindingTime=postBuild
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.
method Activation RoutingGroup	PduActivationRoutingGroup	0..1	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=methodActivationRoutingGroup, methodActivationRoutingGroup.shortName, methodActivationRoutingGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.140: AbstractServiceInstance

It is possible to specify additional information about the AbstractServiceInstance with the Capability Record that allows to transport arbitrary configuration strings (key/-value pairs). This allows to encode additional information like the name of a service or its configuration.

[TPS_SYST_02216] Configuration of capabilityRecords ⌈A Capability Record (key/value pair) is configurable with the AbstractServiceInstance.capability-Record and the two attributes key and value.⌋

Class	**TagWithOptionalValue**			
Note	A tagged value is a combination of a tag (key) and a value that gives supplementary information that is attached to a model element. Please note that keys without a value are allowed.			
Base	*ARObject*			
Aggregated by	*AbstractServiceInstance*.capabilityRecord, Machine.environmentVariable, ProvidedSomeipService Instance.capabilityRecord, RequiredSomeipServiceInstance.capabilityRecord, Startup Config.environmentVariable, StateDependentStartupConfig.environmentVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
key	String	0..1	attr	Defines a key.
sequenceOffset	Integer	0..1	attr	The sequenceOffset attribute supports the use case where TagWithOptionalValue is aggregated as splitable. If multiple aggregations define the same value of attribute key then the order in which the value collection is merged might be significant. As an example consider the modeling of the $PATH environment variable by means of a meta class TagWithOptionalValue. The sequenceOffset describes the relative position of each contribution in the concatenated value. The contributions are sorted in increasing integer order.
value	String	0..1	attr	Defines the corresponding value.

Table 6.141: TagWithOptionalValue

[TPS_SYST_01094] allowed key/value TagWithOptionalValue combinations ⌈The following key/value combinations are supported:

- key present, with no value (e.g. "passreq" -> password required for this service)
- key present, with empty value (e.g. "PlugIns=" -> server supports plugins, but none are presently installed)
- key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")

⌋

[constr_3322] Consistent setting of SoConIPduIdentifier.pduCollection-Semantics in the context of one SocketAddress

Imposition time: CP: IT_SysDesc

⌈The value of the attribute SoConIPduIdentifier.pduCollectionSemantics shall be identical for all referenced SoConIPduIdentifiers within the context of a given SocketAddress.⌋

6.7.4.3.1 SOME/IP Service Instances

A SOME/IP serialized message is represented by an ISignal that aggregates the SOMEIPTransformationISignalProps that in turn references the TransformationTechnology with the protocol set to SOMEIP. The ISignal is mapped into an ISignalIPdu and the PduTriggering that instantiates the ISignalIPdu on the VLAN is related to a ProvidedServiceInstance or ConsumedServiceInstance via a PduActivationRoutingGroup. The ProvidedServiceInstance.majorVersion or ConsumedServiceInstance.majorVersion describes the Service Interface Version that is transported in the Header of the SOME/IP message to identify the source of the message.

[TPS_SYST_02377] Consistent setting of Service Interface Version ⌈The Service Interface Version represented by the SOMEIPTransformation (either via SOMEIPTransformationISignalProps.interfaceVersion of the ISignal or via SOMEIPTransformationDescription.interfaceVersion of the TransformationTechnology that is referenced by the SOMEIPTransformationISignalProps of the ISignal) shall be equal to the corresponding ProvidedServiceInstance.majorVersion or ConsumedServiceInstance.majorVersion.⌋

The following chapters are describing the configuration of SOME/IP ProvidedServiceInstances and ConsumedServiceInstances in more detail. Please note that currently the communication between a ProvidedServiceInstance and a ConsumedServiceInstance is restricted to a VLAN (see also [constr_5079]).

Note: classes called ProvidedServiceInstances and ConsumedServiceInstances are specific for SOME/IP.

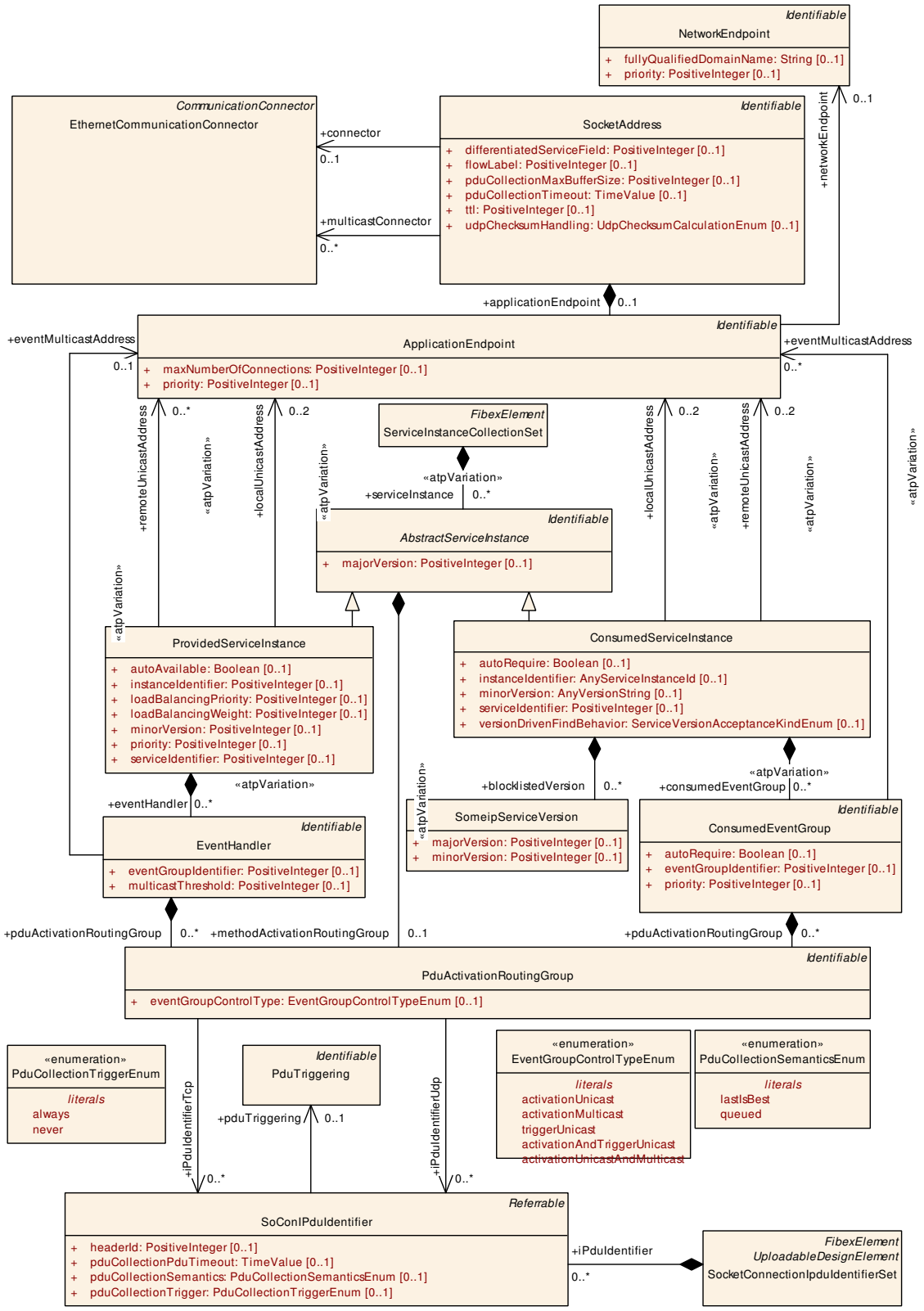

Figure 6.45: Provided and Consumed Service Instances

6.7.4.4 Multicast Subscription

The established approach for service subscription is that a client uses an *unicast* endpoint to subscribe to a server and the server acknowledges the subscription providing a *multicast* endpoint. The client has a *unicast* socket where the events are received in case the server distributes its events in *unicast* mode and the client has a *multicast* socket where the events are received in case the server switches to *multicast* sending (based on the `multicastThreshold`).

An extended approach is to allow a client to subscribe using a *multicast* endpoint (combination of multicast IP-Address and port). Thus, in the event group subscription a *multicast* endpoint is sent from the client to the server. The server handles this *multicast* endpoint as it would be an *unicast* endpoint. But the server collects identical *multicast* endpoints and handles them as ONE client.

The goal of this approach is to be able to have a group of clients subscribing to a common *multicast* endpoint, while other clients still are able to subscribe using a *unicast* endpoint.

This feature is called **`multicast subscription`**.

If, for example, THREE clients subscribe to an event group with the same *multicast* endpoint and ONE client subscribes with an *unicast* endpoint to the same event group, then the server would consider this as having TWO clients. The events of this event group will be sent every time once to the *multicast* endpoint (where it actually would be received by THREE clients) and once to the *unicast* endpoint for the *unicast* client.

Note that, in case `PduActivationRoutingGroup.eventGroupControlType` is defined as `triggerUnicast` or `activationAndTriggerUnicast` there will be an initial event sent every time another client subscribes using the same *multicast* endpoint. Thus, already subscribed clients on that *multicast* endpoint will get that initial event as well.

Even if further clients subscribe to the already used *multicast* endpoint the number of clients counted for the calculation of the `multicastThreshold` is not increased. In the sketched example there would still be TWO clients registered for subscription.

Of course, if further clients subscribe to the very same event group with different *multicast* or *unicast* endpoints, these additional clients are considered in the `multicastThreshold` count. Thus, at exceeding the `multicastThreshold`, count the server will switch to the one *multicast* endpoint which the server distributed to all of its clients in the subscribe acknowledge message. At this point ALL subscribed clients (regardless whether they subscribed using an *unicast* or *multicast* endpoint) will receive the events on the subscribe acknowledge *multicast* endpoint provided by the server.

The server keeps track of how many clients subscribed to a dedicated *multicast* endpoint.

In case of unsubscribe or expiration of a subscription which was subscribed using a *multicast* endpoint the server needs to decrement the count of subscriptions for that

specific *multicast* endpoint. If the last subscriber unsubscribes the subscription to that *multicast* endpoint is removed.

Note that each *multicast* endpoint has an own subscription count to be kept by the server.

For details on the multicast subscription feature refer to the protocol specification [29] and functional specification [30].

Enabling the multicast subscription:

[TPS_SYST_03064] Enabling of multicast subscription ⌈The multicast subscription is enabled when a ConsumedServiceInstance defines a eventMulticastSubscriptionAddress reference to an ApplicationEndpoint.⌋

[constr_3669] eventMulticastSubscriptionAddress shall refer to a multicast address

Imposition time: CP: IT_SysDesc

⌈The reference ConsumedServiceInstance.eventMulticastSubscriptionAddress shall refer to an ApplicationEndpoint which in turn refers to a NetworkEndpoint that represents a multicast address.⌋

If the multicast subscription is enabled for a client then it is not allowed to have also a unicast subscription defined.

[constr_3670] No support for parallel localUnicastAddress and eventMulticastSubscriptionAddress

Imposition time: CP: IT_SysDesc

⌈If a eventMulticastSubscriptionAddress is defined for a ConsumedServiceInstance then there shall not be a localUnicastAddress defined at the same ConsumedServiceInstance.⌋

In case of a static configuration it is also possible to define a set of predefined *multicast* client endpoints at the server. This is supported also in a mixed configuration, where one part of the statically defined clients is defined using ProvidedServiceInstance.remoteUnicastAddress and the other part of the statically defined clients is defined using ProvidedServiceInstance.remoteMulticastSubscriptionAddress. The server takes the union of both sets as the set of statically defined clients.

[TPS_SYST_03065] Static definition of multicast subscription at the server ⌈The ProvidedServiceInstance.remoteMulticastSubscriptionAddress defines a set of remote multicast addresses which are handled by the server as predefined subscribed clients.⌋

[TPS_SYST_03066] Mix of static definition consisting of multicast subscription clients and unicast subscription clients at the server ⌈It is well supported to define both, a set of ProvidedServiceInstance.remoteUnicastAddress and

a set of ProvidedServiceInstance.remoteMulticastSubscriptionAddress references. The server is then configured to handle the union of both sets as the predefined clients.⌋

[constr_3671] remoteMulticastSubscriptionAddress shall refer to a multicast address

Imposition time: CP: IT_SysDesc

⌈The reference ProvidedServiceInstance.remoteMulticastSubscriptionAddress shall refer to an ApplicationEndpoint which in turn refers to a NetworkEndpoint that represents a multicast address.⌋

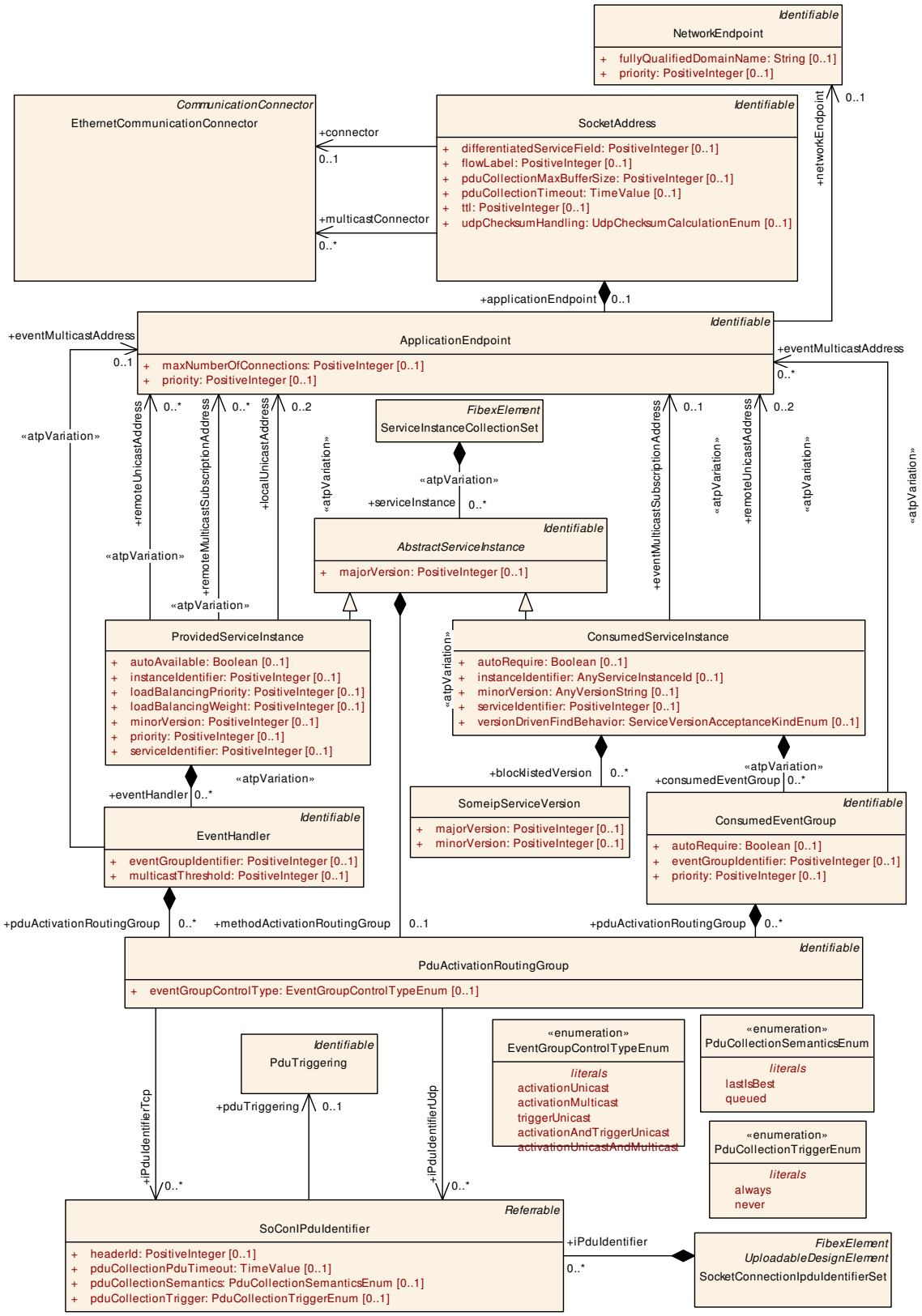

Figure 6.46: Multicast Subscribe

There are several constrains to be considered when using the `multicast subscription` feature:

If a ConsumedServiceInstance has methods defined, then the multicast subscription is not possible.

[constr_3672] No support for methods in multicast subscription at the client

Imposition time: CP: IT_SysDesc

⌈If a ConsumedServiceInstance aggregates a PduActivationRoutingGroup in the role methodActivationRoutingGroup, then the ConsumedServiceInstance shall not define a eventMulticastSubscriptionAddress.⌋

A server is able to define a set of statically configured client addresses using the ProvidedServiceInstance.remoteMulticastSubscriptionAddress reference. But if the server also provides methods it is not possible to define static *multicast* receivers for that server.

[constr_3673] No support for methods in multicast subscription at the server static configuration

Imposition time: CP: IT_SysDesc

⌈If a ProvidedServiceInstance aggregates a PduActivationRoutingGroup in the role methodActivationRoutingGroup, then the ProvidedServiceInstance.remoteMulticastSubscriptionAddress shall not be defined.⌋

There are some features of AUTOSAR which need to be carefully considered when used in combination with multicast subscription:

- communication which relies on sequence counting might not operate properly (e.g. E2E or SecOC) if initial events are sent on every subscription to an *multicast* endpoint
- communication which requires point-to-point interaction may not operate properly (e.g. IPSec or TLS)

6.7.4.5 SOME/IP specific description

6.7.4.5.1 SOME/IP ProvidedServiceInstance

The ProvidedServiceInstance is used to define a SOME/IP Service provider.

[TPS_SYST_02217] SOME/IP Service offer ⌈The EcuInstance on which the ProvidedServiceInstance is deployed offers the Service Instance over SOME/IP with the serviceIdentifier and instanceIdentifier. The version of the Service that is offered is described by the attributes majorVersion and minorVersion.⌋

[constr_5062] SOME/IP ProvidedServiceInstances of the same serviceInterface on one EcuInstance

Imposition time: CP: IT_SysDesc

⌈Different ProvidedServiceInstances with the same serviceIdentifier and the same majorVersion and different instanceIdentifiers shall not be mapped to the same UDP/TCP port number and IP address combination that is represented by referenced ApplicationEndpoint and its referenced NetworkEndpoint.⌋

The reason for this restriction is that the Instance IDs are only used for Service Discovery but are not contained in the SOME/IP header. So if for example two ProvidedServiceInstances of the same ServiceInterface are provided on the same EcuInstance and a client wants to call a method of one of these ProvidedServiceInstances the only possibility for the client to distinguish the ProvidedServiceInstances is the port number over which the individual ProvidedServiceInstances are provided.

[constr_5063] ProvidedServiceInstance.serviceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.serviceIdentifier is mandatory.⌋

[constr_5064] ProvidedServiceInstance.majorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.majorVersion is mandatory.⌋

[constr_5065] ProvidedServiceInstance.minorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.minorVersion is mandatory.⌋

[constr_5066] ProvidedServiceInstance.instanceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.instanceIdentifier is mandatory.⌋

[constr_5067] ProvidedServiceInstance shall be unique in respect of serviceIdentifier, instanceIdentifier, majorVersion

Imposition time: CP: IT_SysDesc

⌈On a VLAN each ProvidedServiceInstance shall have a different serviceIdentifier, instanceIdentifier and majorVersion value combination.⌋

The reason for this constraint is that the Service Discovery messages have to be unambiguous on the VLAN.

In other words no two ProvidedServiceInstances in a variant bound model shall have the same serviceIdentifier, instanceIdentifier and majorVersion value combination.

Class	**ProvidedServiceInstance**			
Note	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowedService Consumer	NetworkEndpoint	*	ref	NetworkEndpoints on which the ConsumedService Instances that are communicating with this Provided ServiceInstance are allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=allowedServiceConsumer, allowedService Consumer.networkEndpoint, allowedService Consumer.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
autoAvailable	Boolean	0..1	attr	Defines that this ProvidedServiceInstance shall be offered by the service discovery at ECU start.
eventHandler	EventHandler	*	aggr	Collection of event groups provided by the Provided ServiceInstance **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventHandler, eventHandler.shortName, eventHandler.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.
loadBalancing Priority	PositiveInteger	0..1	attr	Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.
loadBalancing Weight	PositiveInteger	0..1	attr	Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the PSI is provided (udp, tcp or both). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
minorVersion	PositiveInteger	0..1	attr	Minor Version of the Service that is provided by this ProvidedServiceInstance.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
remoteMulticast Subscription Address	ApplicationEndpoint	*	ref	This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=remoteMulticastSubscriptionAddress, remoteMulticastSubscriptionAddress.applicationEndpoint, remoteMulticastSubscriptionAddress.variationPoint.short Label vh.latestBindingTime=postBuild



△

Class	ProvidedServiceInstance			
remoteUnicast Address	ApplicationEndpoint	*	ref	This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=remoteUnicastAddress, remoteUnicast Address.applicationEndpoint, remoteUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdServerTimer Config	SomeipSdServer ServiceInstanceConfig	0..1	ref	Server specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdServerTimerConfig, sdServerTimer Config.someipSdServerServiceInstanceConfig, sdServer TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is offered.

Table 6.142: ProvidedServiceInstance

[TPS_SYST_02218] **`ProvidedServiceInstance` deployment** ⌈The deployment of a `ProvidedServiceInstance` to an `EcuInstance` is realized with the `localUnicastAddress` reference. The referenced `ApplicationEndpoint` defines the TCP/UDP Port and the `NetworkEndpoint` that is referenced by the `ApplicationEndpoint` defines the IP Address on which the Service can be reached. This Endpoint information is transported in the SOME/IP Service Offer message to the clients. With the `SocketAddress.connector` reference the deployment to an `EcuInstance` is achieved.⌋

The AUTOSAR BswM is used to aggregate the availability of all entities which make up a service instance. When all entities are available, the service instance as such is available. When a service instance becomes available the SD Module will usually send an announcement message so other ECUs can learn about the availability and the location (IP address and UDP or TCP Port) of that service instance.

Please note that the Service provider and the Service consumer do not need to find each other if the `ProvidedServiceInstance` contains a `remoteUnicastAddress` reference to an `ApplicationEndpoint`. If this reference is set, the Server will not get the client's address from the connection request.

The server knows the address of the Client from the configuration and all necessary socket connections can be set up from this configuration information. The client still needs to subscribe to event groups that are of interest for it in this setup.

[TPS_SYST_02219] Static configuration between `ProvidedServiceInstance` and `ConsumedServiceInstance` ⌈If the `ProvidedServiceInstance` contains a `remoteUnicastAddress` reference to an `ApplicationEndpoint` the SoAd will setup one single SocketConnection between the `localUnicastAddress` and the `remoteUnicastAddress`.⌋

[TPS_SYST_02220] Maximal number of clients that may connect to the local server address ⌈If the ProvidedServiceInstance references an ApplicationEndpoint in the role localUnicastAddress then the attribute ApplicationEndpoint.maxNumberOfConnections defines the maximal number of clients (with a dynamic address) that will be able to connect to this local server address. The SoAd will setup a SocketConnection for each potential client with localUnicastAddress and ANY remote Address (ANY Port and ANY IP-Address).⌋

[constr_5068] ProvidedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the ProvidedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_5069] ProvidedServiceInstance.remoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈The ProvidedServiceInstance.remoteUnicastAddress shall point to an IP Unicast address.⌋

In other words the localUnicastAddress and remoteUnicastAddress are not allowed to point to an IP Multicast address. Please note that an IP Unicast address can be defined as a fixed address or be retrieved via dynamic mechanisms, e.g. DHCP.

Please note that a Service may provide some portions (Events, Methods) over TCP and other portions over UDP. This is the reason why the ProvidedServiceInstance is able to reference up to two localUnicastAddresses.

[TPS_SYST_02221] ProvidedServiceInstance.localUnicastAddress reference target ⌈The ProvidedServiceInstance is allowed to have:

- a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port,
- a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port,
- a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port and a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port.

⌋

[constr_3379] Multiple `SocketAddress` entries with the same IP Address, Protocol and Port in the context of a given `EcuInstance`

Imposition time: CP: `IT_SysDesc`

⌈If there are two or more `SocketAddress` entities within the scope of one `SoAdConfig` in the scope of one `EcuInstance` that have the same static (fixed at configuration time) IP Address, Protocol and Port in the aggregated `ApplicationEndpoint` and `NetworkEndpoint`, (e.g., 192.168.1.1, Tcp and 10000, respectively) then only one of these `SocketAddress` elements shall be referenced by `ProvidedServiceInstance`s/`ConsumedServiceInstance`s in the role `localUnicastAddress`.⌋

Rationale for [constr_3379]: There can be only one representation of the `ProvidedServiceInstance`/`ConsumedServiceInstance` using the given IP Address, Protocol and Port in the Sd module configuration in the context of a given `EcuInstance`. Therefore, defining `ProvidedServiceInstance`/`ConsumedServiceInstance` and assign it to several `ApplicationEndpoint`s would in this case require a merge of potentially different attribute values of the `ProvidedServiceInstance`s and/or `ConsumedServiceInstance`s in the System Description and such situation is avoided by this constraint.

Class	**PduActivationRoutingGroup**			
Note	Group of Pdus that can be activated or deactivated for transmission over a socket connection.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AbstractServiceInstance*.methodActivationRoutingGroup, ConsumedEventGroup.pduActivationRoutingGroup, EventHandler.pduActivationRoutingGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventGroupControlType	EventGroupControlTypeEnum	0..1	attr	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.
iPduIdentifierTcp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Tcp in case that the referencing PduActivationRoutingGroup is activated. **Tags:** atp.Splitkey=iPduIdentifierTcp
iPduIdentifierUdp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Udp in case that the referencing PduActivationRoutingGroup is activated. **Tags:** atp.Splitkey=iPduIdentifierUdp

Table 6.143: PduActivationRoutingGroup

Enumeration	**EventGroupControlTypeEnum**
Note	Types of a RoutingGroups for the event communication.
Aggregated by	PduActivationRoutingGroup.eventGroupControlType
Literal	***Description***

▽

△

Enumeration	**EventGroupControlTypeEnum**
activationAnd TriggerUnicast	Activate the data path for unicast events and triggered unicast events that are sent out after a client got subscribed. **Tags:** atp.EnumerationLiteralIndex=0
activationMulticast	Activate the data path for multicast events of an EventGroup. **Tags:** atp.EnumerationLiteralIndex=1
activationUnicast	Activate the data path for unicast events of an EventGroup. **Tags:** atp.EnumerationLiteralIndex=2
activationUnicast AndMulticast	Activate the data path for unicast and multicast events of an EventGroup if they use the identical data path. **Tags:** atp.EnumerationLiteralIndex=4
triggerUnicast	Activate the data path for triggered unicast events that are sent out after a client got subscribed. **Tags:** atp.EnumerationLiteralIndex=3

Table 6.144: EventGroupControlTypeEnum

Class	**SoConIPduIdentifier**			
Note	Identification of Pdu content on a socket connection. This Identifier is required in case that multiple Pdus are transmitted over the same socket connection.			
Base	*ARObject*, *Referrable*			
Aggregated by	SocketConnectionIpduIdentifierSet.iPduIdentifier			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
headerId	PositiveInteger	0..1	attr	If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.
pduCollection PduTimeout	TimeValue	0..1	attr	Defines the timeout in seconds the PDU collection shall be transmitted at the latest after this PDU has been put into the buffer.
pduCollection Semantics	PduCollection SemanticsEnum	0..1	attr	Specifies if the referenced PduTriggering shall be collected using a queued (i.e. all PDU instances) or last-is-best (i.e. only the last PDU instance) semantics. If this attribute is not present the behavior of "queued" is assumed.
pduCollection Trigger	PduCollectionTrigger Enum	0..1	attr	Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.
pduTriggering	PduTriggering	0..1	ref	Reference to a Pdu that is transmitted over a socket connection. **Tags:** atp.Splitkey=pduTriggering

Table 6.145: SoConIPduIdentifier

Class	**SocketConnectionIpduIdentifierSet**			
Note	Collection of PduIdentifiers used for transmission over a Socket Connection with the header option. **Tags:** atp.recommendedPackage=SocketConnectionIpduIdentiferSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**SocketConnectionIpduIdentifierSet**			
iPduIdentifier	SoConIPduIdentifier	*	aggr	Collection of IPduIdentifiers that are transmitted over Socket Connections. **Tags:** atp.Splitkey=iPduIdentifier, iPduIdentifier.shortName

Table 6.146: SocketConnectionIpduIdentifierSet

Enumeration	**PduCollectionSemanticsEnum**
Note	Defines the collection semantics for the PDU collection feature.
Aggregated by	SoConIPduIdentifier.pduCollectionSemantics
Literal	***Description***
lastIsBest	Only the latest PDU instances are transmitted. **Tags:** atp.EnumerationLiteralIndex=0
queued	All instances of PDUs are transmitted. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.147: PduCollectionSemanticsEnum

[TPS_SYST_02222] Usage of `headerId` ⌈If multiple `SoConIPduIdentifier`s are referenced by `PduActivationRoutingGroup` in the role `iPduIdentifierTcp` or in the role `iPduIdentifierUdp` then the `headerId` information shall be used to distinguish between the different `Pdu`s.⌋

Please note that a Method Call and a Method Return may use the same `headerId` for identification since the communication direction is different in this case.

[TPS_SYST_02223] Activation/Deactivation of `PduActivationRoutingGroup`s ⌈The routing of `Pdu`s to and from a socket may be activated or deactivated with a `PduActivationRoutingGroup` depending on the availability of `AbstractServiceInstance`s, `EventHandler`s or `ConsumedEventGroup`s that send or receive the data.⌋

The Routing Group Activation Table is controlled in AUTOSAR by the Service Discovery module.

[TPS_SYST_02224] Methods provided by a `ProvidedServiceInstance` ⌈If the `ProvidedServiceInstance` is offered by the Service Discovery protocol then the `PduActivationRoutingGroup` that is aggregated in the role `methodActivationRoutingGroup` is activated. All Methods that are provided by the `ProvidedServiceInstance` can by called by the interested clients and the server will respond to these calls. If the `ProvidedServiceInstance` offer is stopped then the `PduActivationRoutingGroup` is deactivated.⌋

Please note that according to [TPS_SYST_02151] the relationship between the call and return is achieved by means of meta data items attached to the `Pdu`s by the Socket Adapter.

[TPS_SYST_02081] PduTriggering that is used for ClientServer Communication ⌈A PduTriggering that points to an ISignalIPdu that aggregates an ISignalToIPduMapping that in turn references an ISignal that refers to a ClientServerToSignalMapping.callSignal or to ClientServerToSignalMapping.returnSignal is designated as PduTriggering that is used for ClientServer Communication.⌋

[TPS_SYST_02225] Service methods provided over UDP ⌈Method Pdus (Call and Return ISignalIPdus) of a ProvidedServiceInstance that are offered for access over UDP are described by SoConIPduIdentifiers referenced by the iPduIdentifierUdp reference from the PduActivationRoutingGroup that is aggregated in the role methodActivationRoutingGroup by the ProvidedServiceInstance.⌋

[TPS_SYST_02226] Service methods provided over TCP ⌈Method Pdus (Call and Return ISignalIPdus) of a ProvidedServiceInstance that are offered for access over TCP are described by SoConIPduIdentifiers referenced by the iPduIdentifierTcp reference from the PduActivationRoutingGroup that is aggregated in the role methodActivationRoutingGroup by the ProvidedServiceInstance.⌋

[TPS_SYST_02227] Publishing of a SOME/IP Event group ⌈A ProvidedServiceInstance publishes an event group for each EventHandler that is aggregated in the role eventHandler. The eventGroupIdentifier identifies the SOME/IP Event Group in the context of the ProvidedServiceInstance. With the publishing of an event group the server offers to push notifications about updates to all clients that are subscribed to the event group.⌋

Class	**EventHandler**			
Note	This element represents an event group as part of the Provided Service Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ProvidedServiceInstance.eventHandler			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventGroup Identifier	PositiveInteger	0..1	attr	Unique Identifier that identifies the EventGroup in SOME/IP. This Identifier is sent as Eventgroup ID in SOME/IP Service Discovery messages.

▽

△

Class	EventHandler			
eventMulticast Address	ApplicationEndpoint	0..1	ref	Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the mulicastThreshold is exceeded. This address is transmitted in the SD-SubscribeEvent GroupAck Message to client (answer to SD-Subscribe EventGroup). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastAddress, eventMulticast Address.applicationEndpoint, eventMulticast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
multicast Threshold	PositiveInteger	0..1	attr	Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast. If configured to 0 only unicast will be used. If configured to 1 the first client will be already served by multicast. If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast. This does not influence the handling of initial events, which are served using unicast only.
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events. **Tags:** atp.Splitkey=pduActivationRoutingGroup, pdu ActivationRoutingGroup.shortName
sdServerEg TimingConfig	SomeipSdServerEvent GroupTimingConfig	0..1	ref	Server Timing configuration settings that are EventGroup specific. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdServerEgTimingConfig, sdServerEg TimingConfig.someipSdServerEventGroupTimingConfig, sdServerEgTimingConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.148: EventHandler

[TPS_SYST_02228] Transmission of events over UDP/TCP Port ⌈The events of an event group that are described by an `EventHandler` are transmitted either:

- via IP Unicast on the UDP/TCP Port that is defined by the `ApplicationEndpoint` that is referenced by the `ProvidedServiceInstance` in the role `localUnicastAddress` if the `eventMulticastAddress` is not set or the `multicastThreshold` is not reached, or
- via IP Multicast on the local UDP Port that is defined by the `ApplicationEndpoint` that is referenced by the `ProvidedServiceInstance` in the role `localUnicastAddress` to the IP Multicast remote Address that is defined by the `ApplicationEndpoint` that is referenced by the `EventHandler` in the role `eventMulticastAddress` if the `multicastThreshold` is reached.

⌋

[constr_5331] No IP multicast in case of TCP

Imposition time: CP: IT_SysDesc

⌈The ApplicationEndpoint that is referenced in the role eventMulticastAddress from an EventHandler is only allowed to aggregate UdpTp in the role tpConfiguration.⌋

[TPS_SYST_02180] Usage of EventHandler.multicastThreshold ⌈The switching between IP-Unicast and IP-Multicast is guided by the server with the EventHandler.multicastThreshold attribute and by the number of subscribed clients to the EventHandler.

The Server will change the transmission of events to Multicast if the multicastThreshold of the corresponding EventHandler is reached by the number of subscribed clients. If the number of subscribed clients is smaller then the configured multicastThreshold, the transmission of events takes place via unicast communication.⌋

[constr_5380] Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of one VLAN

Imposition time: CP: IT_SysDesc

⌈SoConIPduIdentifiers with the same headerId shall not be referenced by PduActivationRoutingGroups of different EventHandlers if

- one or several of these EventHandlers has the multicastThreshold set to a value>0 and
- all these EventHandlers are aggregated by ProvidedServiceInstances that reference ApplicationEndpoints with the localUnicastAddress reference that in turn are aggregated by SocketAddresses which contain a reference to the same EthernetCommunicationConnector in the role connector (i.e. the EventHandlers are located on the same VLAN).

except for the case that all these EventHandlers have the multicastThreshold set to the value 1.⌋

[constr_5380] is introduced to forbid scenarios on one VLAN in which the Service Provider is forced to transmit the same event over IP Unicast and IP Multicast in cases where the same event is assigned to several EventHandlers (EventGroups) and the multicastThreshold is reached for one EventHandler, but not for the other ones.

[constr_5071] EventHandler.eventMulticastAddress reference target

Imposition time: CP: IT_SysDesc

⌈The ApplicationEndpoint that is referenced by an EventHandler in the role eventMulticastAddress shall reference a NetworkEndpoint that defines an IP Multicast Address.⌋

[constr_5072] `EventHandler` without defined `eventMulticastAddress`

Imposition time: CP: `IT_SysDesc`

⌈If an `EventHandler` that is aggregated by a `ProvidedServiceInstance` does not have a defined `eventMulticastAddress` then the `multicastThreshold` shall be set to the value 0 (IP Unicast only).⌋

[TPS_SYST_02229] Event groups provided by a `ProvidedServiceInstance`
⌈If the `ProvidedServiceInstance` publishes an event group then the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by an `EventHandler` is activated. The interested clients can subscribe to the published event groups until the offer is stopped.⌋

[TPS_SYST_02230] `PduActivationRoutingGroup`s for event groups ⌈The `PduActivationRoutingGroup` that is aggregated by the `EventHandler` in the role `pduActivationRoutingGroup` enables the routing of a group of `Pdu`s that are related with the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifier`s.

- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` enables the routing over IP Unicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` enables the routing over IP Multicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `triggerUnicast` enables the routing of initial events that are sent out by the server immediately after a client got subscribed. This `PduActivationRoutingGroup` can used for SOME/IP Field notifiers.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationAndTriggerUnicast` enables the routing over IP Unicast and makes sure that initial events are sent out by the server immediately after a client got subscribed. This `PduActivationRoutingGroup` can be used for SOME/IP Field notifiers.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicastAndMulticast` enables the routing over IP Unicast and Multicast. The Pdus share the same `PduActivationRoutingGroup` for unicast and multicast.

⌋

[TPS_SYST_02231] `PduActivationRoutingGroup`s for methods ⌈The `PduActivationRoutingGroup` enables the routing of a group of `Pdu`s that are related with the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifier`s. The `eventGroupControlType` attribute is irrelevant for the

PduActivationRoutingGroup that is aggregated by the AbstractServiceInstance in the role methodActivationRoutingGroup.⌋

[constr_5073] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is aggregated by an EventHandler

Imposition time: CP: IT_SysDesc

⌈An EventHandler that aggregates a PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast shall be aggregated by a ProvidedServiceInstance that has a localUnicastAddress reference that points to an IP Unicast Address.⌋

[constr_5074] PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast that is aggregated by an EventHandler

Imposition time: CP: IT_SysDesc

⌈An EventHandler that aggregates a PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationMulticast or activationUnicastAndMulticast shall have an eventMulticastAddress reference that points to a “remote” IP Multicast Address. The ProvidedServiceInstance that aggregates the EventHandler shall have a localUnicastAddress reference to a “local” UDP ApplicationEndpoint.⌋

[constr_5075] Allowed references of SoConIPduIdentifiers by PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast and allowed SoConIPduIdentifier references

Imposition time: CP: IT_SysDesc

⌈A PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast is allowed to reference SoConIPduIdentifiers only in the iPduIdentifierUdp role.⌋

[TPS_SYST_02232] Events provided over UDP ⌈Pdus of an EventHandler that are provided over UDP are described by SoConIPduIdentifiers referenced by the iPduIdentifierUdp reference from the PduActivationRoutingGroup that is aggregated in the role pduActivationRoutingGroup by the EventHandler.⌋

[TPS_SYST_02233] Events provided over TCP ⌈Pdus of an EventHandler that are provided over TCP are described by SoConIPduIdentifiers referenced by the iPduIdentifierTcp reference from the PduActivationRoutingGroup that is aggregated in the role pduActivationRoutingGroup by the EventHandler.⌋

[constr_5076] PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ProvidedServiceInstance

Imposition time: CP: IT_SysDesc

⌈If the PduActivationRoutingGroup contains the iPduIdentifierTcp reference then the aggregating ProvidedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP address.⌋

[constr_5077] PduActivationRoutingGroup with iPduIdentifierUdp reference that is aggregated by a ProvidedServiceInstance

Imposition time: CP: IT_SysDesc

⌈If the PduActivationRoutingGroup contains the iPduIdentifierUdp reference then the aggregating ProvidedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP address.⌋

[constr_5078] PduTriggerings referenced by a PduActivationRoutingGroup shall be on the same VLAN as the referencing PduActivationRoutingGroup

Imposition time: CP: IT_SysDesc

⌈Each PduTriggering referenced by a PduActivationRoutingGroup via SoConIPduIdentifier shall be aggregated by the same VLAN (EthernetPhysicalChannel) to which the AbstractServiceInstance that aggregates the PduActivationRoutingGroup belongs via the localUnicastAddress.⌋

[constr_5079] Service communication is restricted to one VLAN

Imposition time: CP: IT_SysDesc

⌈All SocketAddress elements that are referenced by a AbstractServiceInstance with the localUnicastAddress and remoteUnicastAddress shall belong to the same VLAN (EthernetPhysicalChannel).⌋

[constr_5080] ApplicationEndpoints referenced by EventHandlers and by the aggregating ProvidedServiceInstance shall be in the same VLAN

Imposition time: CP: IT_SysDesc

⌈The ApplicationEndpoint that is referenced by an EventHandler in the role eventMulticastAddress shall belong to the same VLAN (EthernetPhysicalChannel) as the ApplicationEndpoint that is referenced by the localUnicastAddress reference from the ProvidedServiceInstance that aggregates the EventHandler.⌋

[constr_3456] Existence of `ProvidedServiceInstance`.`loadBalancingPriority` and `ProvidedServiceInstance`.`loadBalancingWeight`

Imposition time: CP: `IT_SysDesc`

⌈The attributes `ProvidedServiceInstance.loadBalancingPriority` and `ProvidedServiceInstance.loadBalancingWeight` shall either not exist or be defined both.⌋

[TPS_SYST_01108] `ProvidedServiceInstance` priority ⌈The `priority` in the `ProvidedServiceInstance` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership`, the `priority` that is defined at the `NetworkEndpoint` and the `priority` that is defined at the `ApplicationEndpoint`.⌋

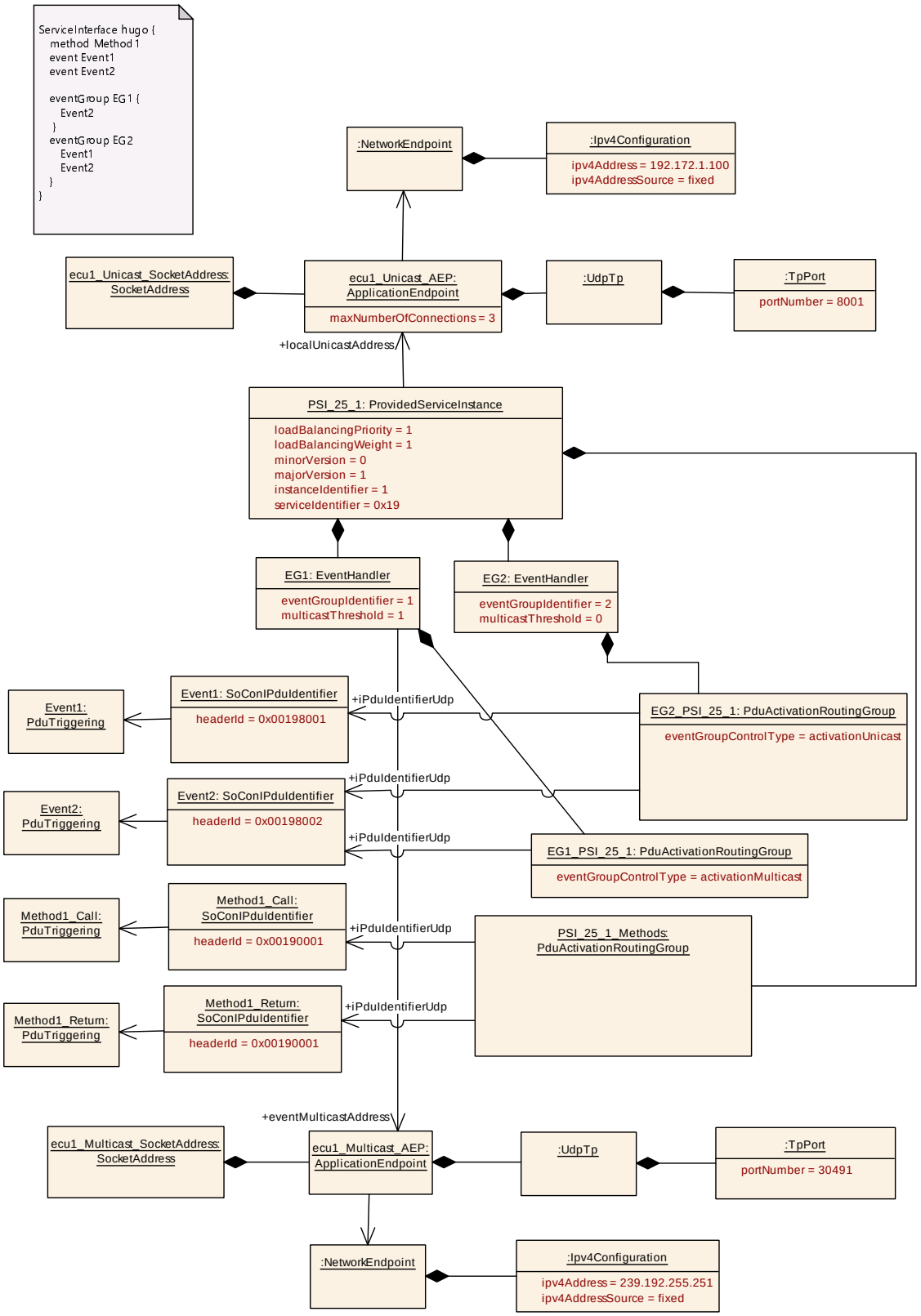

Figure 6.47: Example for the modeling of a `ProvidedServiceInstance` that is deployed on a local Unicast Endpoint

The example in Figure 6.47 shows a ProvidedServiceInstance named PSI_25_1 that is provided on Udp Port 8001 and IPv4 address 192.172.1.100. The Service that is represented by the ProvidedServiceInstance contains one method (named Method1) and two events (named Event1 and Event2). In addition the service contains two different event groups that are represented by two EventHandlers EG1 and EG2.

On VFB level the method is represented by a ClientServerOperation and each event by a VariableDataPrototype in a SenderReceiverInterface.

The ClientServerOperation is mapped by a ClientServerToSignalMapping to a callSignal and to a returnSignal that in turn are mapped via ISignals into an ISignalIPdu. Each VariableDataPrototype is mapped by the SenderReceiverToSignalMapping or by the SenderReceiverToSignalGroupMapping to one or several SystemSignals that in turn are mapped via ISignals into an ISignalIPdu.

The PduTriggerings of these ISignalIPdus are referenced by the SoConIPduIdentifier that assigns a headerId to each ISignalIPdu.

The event group EG2 contains Event1 and Event2 and the PduActivationRoutingGroup named EG2_PSI_25_1 activates the SoConIPduIdentifiers that are representing these two events for transmission over IP Unicast since the eventGroupControlType is set to activationUnicast.

The event group EG1 contains Event1 and the PduActivationRoutingGroup named EG1_PSI_25_1 activates the SoConIPduIdentifier for transmission over IP Multicast since the eventGroupControlType is set to activationMulticast. EventHandler EG1 has the multicastThreshold attribute set to 1. This means that the first client that will subscribe to this event group will be served via IP Multicast. This is the reason why the EventHandler EG1 points with the eventMulticastAddress to an IP Multicast address to which the event will be transmitted.

In the SoAd configuration such a System Description will result in a SoAdSocketConnectionGroup with a SoAdSocketLocalPort and SoAdSocketLocalAddressRef that are derived from the ApplicationEndpoint referenced in the localUnicastAddress role by the ProvidedServiceInstance.

The SoAdSocketConnectionGroup will contain three SoAdSocketConnections since the maxNumberOfConnections is set to 3 in this example. The ProvidedServiceInstance does not contain a remoteUnicastAddress reference. This means that Service Discovery is used and that the SoAdSocketRemoteIpAddress needs to be configured to ANY and the SoAdSocketRemotePort to 0 in all three SoAdSocketConnections.

In addition one multicast SoAdSocketConnection will be added to the SoAdSocketConnectionGroup since the EventHandler EG1 points with the eventMulticastAddress to a multicast address.

6.7.4.5.2 SOME/IP ConsumedServiceInstance

The ConsumedServiceInstance is used to define a SOME/IP Service consumer. Please note that in the AUTOSAR model all necessary information about the searched service is configurable on the client side. A model that contains only the ConsumedServiceInstance description is sufficient for the configuration of the EcuInstance to which the ConsumedServiceInstance is deployed. This is different to former versions of the System Template. The design criterion for the model was to disentangle the ProvidedServiceInstance and the ConsumedServiceInstance from each other.

[TPS_SYST_02234] SOME/IP Service search ⌈A defined ConsumedServiceInstance is searching for a SOME/IP Service Instance that fulfills all of the following conditions:

- Service Identifier that matches the value set in serviceIdentifier,
- Service Instance Identifier that matches the value set in instanceIdentifier
- Service major version that matches the value set in majorVersion,
- Service minor version:
  - in case versionDrivenFindBehavior = exactOrAnyMinorVersion: Service minor version that matches the value set in minorVersion or *ANY* minor version of the Service Instance in case the minorVersion is set to *ANY*
  - in case versionDrivenFindBehavior = minimumMinorVersion: Service minor version that matches at least the value set in minorVersion or is higher

⌋

[constr_5110] Search for a collection of ServiceInstances is not supported

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.instanceIdentifier is not allowed to be set to the value ANY or ALL.⌋

The reason for [constr_5110] is that the AUTOSAR SD module is only able to send wildcard finds for a Minor Version. The search for a collection of ServiceInstances is not supported by the Classic Platform Basic Software.

[constr_5081] ConsumedServiceInstance.serviceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.serviceIdentifier is mandatory.⌋

[constr_5082] ConsumedServiceInstance.majorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.majorVersion is mandatory.⌋

[constr_5083] ConsumedServiceInstance.minorVersion is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.minorVersion is mandatory.⌋

[constr_3560] minimumMinorVersion and ConsumedServiceInstance.minorVersion value

Status: DRAFT

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.minorVersion shall not have the value *ANY* if versionDrivenFindBehavior = minimumMinorVersion.⌋

[constr_5084] ConsumedServiceInstance.instanceIdentifier is mandatory

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.instanceIdentifier is mandatory.⌋

Class	**ConsumedServiceInstance**			
Note	Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowedService Provider	NetworkEndpoint	*	ref	NetworkEndpoint on which the ProvidedServiceInstance that is communicating with this ConsumedService Instance is allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=allowedServiceProvider, allowedService Provider.networkEndpoint, allowedService Provider.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
autoRequire	Boolean	0..1	attr	Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.
blocklisted Version	SomeipServiceVersion	*	aggr	Collection of blocklisted versions **Tags:** atp.Splitkey=blocklistedVersion atp.Status=draft

▽

△

Class	**ConsumedServiceInstance**			
consumedEvent Group	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedEventGroup, consumedEvent Group.shortName, consumedEventGroup.variation Point.shortLabel vh.latestBindingTime=postBuild
eventMulticast Subscription Address	ApplicationEndpoint	0..1	ref	Multicast Address that is used by the client to subscribe to the server: This enables the multicast subscription feature. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastSubscriptionAddress, event MulticastSubscriptionAddress.applicationEndpoint, event MulticastSubscriptionAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	AnyServiceInstanceId	0..1	attr	This attribute represents the ability to describe the required service instance ID.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the CSI is consumed (udp, tcp or both). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
minorVersion	AnyVersionString	0..1	attr	Minor Version of the ServiceInterface. Value can be set to a number that represents the Minor Version of the searched service or to ANY.
remoteUnicast Address	ApplicationEndpoint	0..2	ref	This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=remoteUnicastAddress, remoteUnicast Address.applicationEndpoint, remoteUnicast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdClientTimer Config	SomeipSdClientService InstanceConfig	0..1	ref	Client specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig, sdClientTimer Config.someipSdClientServiceInstanceConfig, sdClient TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/IP service ID that is searched.
versionDriven FindBehavior	ServiceVersion AcceptanceKindEnum	0..1	attr	Defines the service discovery find behavior. **Tags:** atp.Status=draft

Table 6.149: ConsumedServiceInstance

[TPS_SYST_03050] Usage of ConsumedServiceInstance.blocklistedVersion

Status: DRAFT

⌈A service connection of a ConsumedServiceInstance to a ProvidedServiceInstance is not considered for service discovery if the SomeipServiceVersion.minorVersion exists in the collection of SomeipServiceVersions aggregated at the ConsumedServiceInstance in the role blocklistedVersion.⌋

A typical scenario for using a blocklist may be: For a certain ConsumedServiceInstance a certain compatible provider service version inside a system may not work which may have been identified after the design phase. In order to keep the system running this certain provider version won't be considered in the service search if it has been blocklisted. Therefore, the ConsumedServiceInstance may connect only to ProvidedServiceInstances that fulfill the search criteria and are not blocklisted.

[constr_3559] ConsumedServiceInstance.blocklistedVersion is restricted to the usage of minorVersion

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The majorVersion attribute shall not be used in the SomeipServiceVersion that is aggregated by the ConsumedServiceInstance in the role blocklistedVersion.⌋

[TPS_SYST_02235] ConsumedServiceInstance deployment ⌈The deployment of a ConsumedServiceInstance to an EcuInstance is realized in two ways:

- if the localUnicastAddress reference is available then the referenced ApplicationEndpoint defines the TCP/UDP Port and the NetworkEndpoint that is referenced by the ApplicationEndpoint defines the IP Address on which the service consumer is located. Over the defined TCP/UDP port the service consumer sends out the method calls to a service provider that was found by the Service Discovery search. The endpoint information is also transported in the SOME/IP SubscribeEventGroup message to the server to indicate on which address the incoming events are expected. With the SocketAddress.connector reference the deployment to an EcuInstance is achieved.
- if the localUnicastAddress reference is not available (in case Events are received via IP Multicast only as described by [TPS_SYST_02302] and the aggregating ConsumedServiceInstance does not define any methods) then the ApplicationEndpoint that is referenced by the eventMulticastAddress reference from the ConsumedEventGroup shall be EcuInstance specific and the SocketAddress that contains the referenced ApplicationEndpoint is allowed to reference only a single EthernetCommunicationConnector in the multicastConnector role. With this SocketAddress.multicastConnector reference the deployment to an EcuInstance is achieved.

⌋

Please note that the Service provider and the Service consumer do not need to find each other if the ConsumedServiceInstance contains a remoteUnicastAddress reference to an ApplicationEndpoint. If this reference is set, the client will not get the server's address from the Service Offer message.

The client already knows the address of the service provider from the configuration and all necessary socket connections can be set up from this configuration information. The client still needs to subscribe to event groups that are of interest for it in this setup.

[TPS_SYST_02236] Static configuration between ConsumedServiceInstance and ProvidedServiceInstance ⌈If the ConsumedServiceInstance contains a remoteUnicastAddress reference to an ApplicationEndpoint the SoAd will setup one single SocketConnection between the localUnicastAddress and the remoteUnicastAddress.⌋

[TPS_SYST_02237] Maximal number of servers that may connect to the local client address ⌈If the ConsumedServiceInstance references an ApplicationEndpoint in the role localUnicastAddress and the remoteUnicastAddress is not defined then the attribute ApplicationEndpoint.maxNumberOfConnections defines the maximal number of remote endpoints that will be able to connect to this local client address. The SoAd will setup a SocketConnection for each potential remote endpoint. The local Address of the SocketConnection is derived from ConsumedServiceInstance.localUnicastAddress. The remote Address is set to ANY address in case that the remoteUnicastAddress is not used.⌋

[constr_5085] ConsumedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the ConsumedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

In other words the localUnicastAddress is not allowed to point to a IP Multicast address. Please note that the ConsumedServiceInstance.localUnicastAddress does not need to be set in all cases as specified in [TPS_SYST_02238].

[constr_5086] ConsumedServiceInstance.remoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈The ConsumedServiceInstance.remoteUnicastAddress shall point to an IP Unicast address.⌋

Please note that a Service may provide some portions (Events, Methods) over TCP and and other portions over UDP. This is the reason why the ConsumedServiceInstance is able to reference up to two localUnicastAddresses.

[TPS_SYST_02238] ConsumedServiceInstance.localUnicastAddress reference target ⌈The ConsumedServiceInstance is allowed to have:

- a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port,
- a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port,
- a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port and a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port.
- no configured localUnicastAddress.

⌋

The Routing Group Activation is controlled in AUTOSAR by the Service Discovery module.

[TPS_SYST_02239] Methods consumed by a ConsumedServiceInstance ⌈If the ConsumedServiceInstance finds the searched service then the PduActivationRoutingGroup that is aggregated in the role methodActivationRoutingGroup is activated. All Methods that are provided by the found ProvidedServiceInstance can by called by ConsumedServiceInstance.⌋

Please note that according to [TPS_SYST_02151] the relationship between the call and return is achieved by means of meta data items attached to the Pdus by the Socket Adapter.

[TPS_SYST_02240] Service methods consumed over UDP ⌈Method Pdus (Call and Return ISignalIPdus) of a ConsumedServiceInstance that are accessed over UDP are described by SoConIPduIdentifiers referenced by the iPduIdentifierUdp reference from the PduActivationRoutingGroup that is aggregated in the role methodActivationRoutingGroup by the ConsumedServiceInstance.⌋

[TPS_SYST_02241] Service methods consumed over TCP ⌈Method Pdus (Call and Return ISignalIPdus) of a ConsumedServiceInstance that are accessed over TCP are described by SoConIPduIdentifiers referenced by the iPduIdentifierTcp reference from the PduActivationRoutingGroup that is aggregated in the role methodActivationRoutingGroup by the ConsumedServiceInstance.⌋

[TPS_SYST_02242] Subscription to a SOME/IP Event group ⌈A ConsumedServiceInstance subscribes to a published event group via a ConsumedEventGroup that is aggregated in the role consumedEventGroup. The eventGroupIdentifier identifies the SOME/IP Event Group in the context of the ConsumedServiceInstance to which the consumer subscribes.⌋

Class	**ConsumedEventGroup**			
Note	This element represents an event-group to which the service consumer wants to subscribe.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ConsumedServiceInstance.consumedEventGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
autoRequire	Boolean	0..1	attr	Defines that this ConsumedEventGroup shall be requested (subscribed) as soon as the corresponding ConsumedServiceInstance is requested. This could be at ECU start, if ConsumedServiceInstance.autoRequire is set to TRUE or as soon as the ConsumedServiceInstance is requested by the application, if ConsumedService Instance.autoRequire is set to FALSE.
eventGroup Identifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.
eventMulticast Address	ApplicationEndpoint	*	ref	This reference defines the multicast address or a multicast address resource where the events of the event group are received. If the multicast address is determined via configuration and not at runtime via service discovery this reference points to the multicast address over which the events will be received. If the multicast address is determined at runtime via service discovery this reference shall be used to define the necessary local multicast address resources, i.e. RAM space in the TcpIp module in which the multicast address is stored at runtime. Please note that in this case the referenced address may be defined as ANY UDP port and ANY IP address since the multicast address will be received at runtime. If several multicast addresses are considered to be used the ConsumedEventGroup shall point to different ApplicationEndpoint objects to reserve the necessary resources in the configuration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=eventMulticastAddress, eventMulticast Address.applicationEndpoint, eventMulticast Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events. **Tags:** atp.Splitkey=pduActivationRoutingGroup, pdu ActivationRoutingGroup.shortName
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
sdClientTimer Config	SomeipSdClientEvent GroupTimingConfig	0..1	ref	Client Timing configuration settings that are EventGroup specific. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig, sdClientTimer Config.someipSdClientEventGroupTimingConfig, sdClient TimerConfig.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.150: ConsumedEventGroup

[TPS_SYST_02243] Reception of events over UDP/TCP Port in case of Service Discovery ⌈The events of an event group that are described by an `ConsumedEvent-Group` are received either:

- via IP Unicast on the UDP/TCP Port that is defined by the ApplicationEndpoint that is referenced by the ConsumedServiceInstance in the role localUnicastAddress or
- via IP Multicast:
  - in case of dynamic SD configuration the Multicast IP Address and UDP Port to which the events are transmitted will be provided by the server at runtime in the SOME/IP SubscribeEventGroupAck message,
  - in case of static configuration the Multicast IP Address and UDP Port to which the events are transmitted shall be configured by ConsumedEventGroup.eventMulticastAddress.

⌋

Please note that the ConsumedEventGroup.eventMulticastAddress shall also be used in case of a dynamic SD Configuration where the client learns the multicast address at runtime. In this case the ApplicationEndpoint and the corresponding NetworkEndpoint that are referenced by eventMulticastAddress define a resource in the TcpIp module where the multicast address that will be determined at runtime will be stored. Since the referenced ApplicationEndpoint and the corresponding NetworkEndpoint are placeholders the configured TP Port and IP address shall be set to ANY.

There are different scenarios that are considered:

- The same multicast address is used by all potential servers. On the client a resource for this single multicast address needs to be reserved. All ConsumedEventGroups need to reference the same placeholder ApplicationEndpoint with eventMulticastAddress. In the Ecu Configuration one TcpIpLocalAddr container is created from the placeholder ApplicationEndpoint. In addition one SoAdSocketConnectionGroup is created that points with the SoAdSocketLocalAddressRef to the TcpIpLocalAddr container. All Pdus received over multicast are assigned to this SoAdSocketConnectionGroup.
- The server is using different multicast addresses, e.g. Multicast Address A for EventHandler A and Multicast Address B for EventHandler B. On the client side resources for all used multicast addresses need to be reserved. In the example the ConsumedEventGroup A needs to reference one placeholder ApplicationEndpoint with eventMulticastAddress. The ConsumedEventGroup B needs to reference a second placeholder ApplicationEndpoint with eventMulticastAddress. In the Ecu Configuration a TcpIpLocalAddr container is created for each used placeholder. In addition one SoAdSocketConnectionGroup per placeholder is created. In the example the Pdus of ConsumedEventGroup A are assigned to one SoAdSocketConnectionGroup and Pdus of ConsumedEventGroup B are assigned to the second SoAdSocketConnectionGroup. With this approach it is possible to define at

configuration time which Pdus share the same multicast address, and which ones are going over different multicast addresses.

If more than one multicast address is used it is not known at configuration time which server will publish which address at which point in time. The configuration needs to be prepared for all possible scenarios. Please be aware that this comes with a high resource consumption.

[constr_3262] `ConsumedEventGroup.eventGroupIdentifier` is mandatory

Imposition time: CP: `IT_SysDesc`

⌈The `ConsumedEventGroup.eventGroupIdentifier` is mandatory.⌋

[constr_3457] Uniqueness of `ConsumedEventGroup.eventGroupIdentifier` in the scope of a `ConsumedServiceInstance`

Imposition time: CP: `IT_SysDesc`

⌈Each `ConsumedEventGroup` that is aggregated by a `ConsumedServiceInstance` shall have a unique `eventGroupIdentifier` value in the scope of the aggregating `ConsumedServiceInstance`.⌋

[TPS_SYST_02245] Event groups consumed by a `ConsumedServiceInstance`
⌈If the `ConsumedServiceInstance` subscribes to an event group then the `PduActivationRoutingGroup` that is aggregated in the role `pduActivationRoutingGroup` by an `ConsumedEventGroup` is activated.⌋

[TPS_SYST_02246] `PduActivationRoutingGroup`s for `ConsumedEventGroup`s
⌈The `PduActivationRoutingGroup` that is aggregated by the `ConsumedEventGroup` in the role `pduActivationRoutingGroup` enables the routing of a group of `Pdu`s that are related to the `PduActivationRoutingGroup` via the referenced `SoConIPduIdentifier`s.

- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` enables the receiving of events over IP Unicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` enables the receiving of events over IP Multicast.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `triggerUnicast` enables the receiving of initial events that are sent out by the server immediately after the client got subscribed. This `PduActivationRoutingGroup` can used for SOME/IP Field notifiers.
- The `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationAndTriggerUnicast` enables the routing over IP Unicast and makes sure that initial events are sent out

by the server immediately after the client got subscribed. This PduActivationRoutingGroup can be used for SOME/IP Field notifiers.

- The PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicastAndMulticast enables the receiving over IP Unicast and Multicast. The Pdus share the same PduActivationRoutingGroup for unicast and multicast.

⌋

[constr_5087] PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is referenced by a ConsumedEventGroup

Imposition time: CP: IT_SysDesc

⌈A ConsumedEventGroup that aggregates a PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast shall be aggregated by a ConsumedServiceInstance that has a localUnicastAddress reference that points to an IP Unicast Address.⌋

[constr_9402] PduActivationRoutingGroup with PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast that is aggregated by a ConsumedEventGroup

Imposition time: CP: IT_SysDesc

⌈If a ConsumedEventGroup aggregates at least one PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast, then the aggregating ConsumedServiceInstance shall have a localUnicastAddress reference to a "local" UDP ApplicationEndpoint.⌋

Note that in the context of [constr_9402] also [constr_5075] applies as well.

[TPS_SYST_02247] Events consumed over UDP ⌈Pdus of an ConsumedEventGroup that are consumed over UDP are described by SoConIPduIdentifiers referenced by the iPduIdentifierUdp reference from the PduActivationRoutingGroup that is aggregated in the role pduActivationRoutingGroup by the ConsumedEventGroup.⌋

[TPS_SYST_02248] Events consumed over TCP ⌈Pdus of an ConsumedEventGroup that are consumed over TCP are described by SoConIPduIdentifiers referenced by the iPduIdentifierTcp reference from the PduActivationRoutingGroup that is aggregated in the role pduActivationRoutingGroup by the ConsumedEventGroup.⌋

[constr_5088] PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ConsumedServiceInstance

Imposition time: CP: IT_SysDesc

⌈If the PduActivationRoutingGroup contains the iPduIdentifierTcp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP address.⌋

[constr_5089] PduActivationRoutingGroup with iPduIdentifierUdp reference that is aggregated by a ConsumedServiceInstance

Imposition time: CP: IT_SysDesc

⌈If the PduActivationRoutingGroup contains the iPduIdentifierUdp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP address.⌋

[TPS_SYST_02014] ConsumedEventGroup priority ⌈The priority in the ConsumedEventGroup shall be used as Ethernet Header information together with the vlanIdentifier. If defined the priority overwrites the defaultPriority that is defined in the VlanMembership, the priority that is defined at the NetworkEndpoint and the priority that is defined at the ApplicationEndpoint.⌋

[constr_5090] ApplicationEndpoints referenced by ConsumedEventGroups and by the aggregating ConsumedServiceInstance shall be in the same VLAN

Imposition time: CP: IT_SysDesc

⌈The ApplicationEndpoint that is referenced by an ConsumedEventGroup in the role eventMulticastAddress shall belong to the same VLAN (EthernetPhysicalChannel) as the ApplicationEndpoint that is referenced by the localUnicastAddress reference from the ConsumedServiceInstance that aggregates the ConsumedEventGroup.⌋

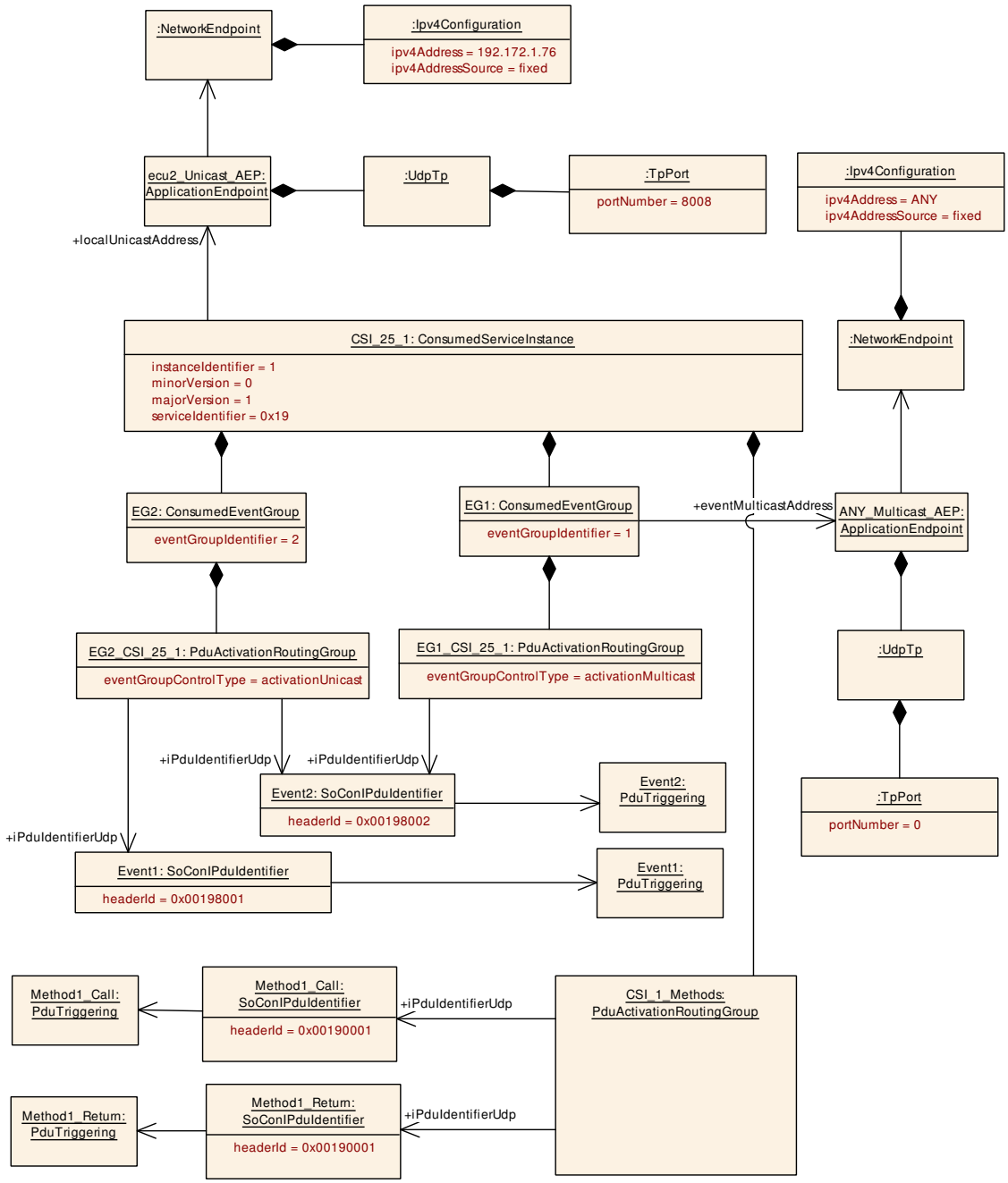

Figure 6.48: Example for the modeling of a `ConsumedServiceInstance` that is deployed on a local Unicast Endpoint

The example in Figure 6.48 shows a `ConsumedServiceInstance` named `CSI_25_1` that is deployed on Udp Port 8008 and IPv4 address 192.172.1.76. The Service that is represented by the `ConsumedServiceInstance` contains one method (named `Method1`) and two events (named `Event1` and `Event2`). In addition the service contains two different event groups that are represented by two `ConsumedEventGroup`s `EG1` and `EG2`.

The PduTriggerings for the ISignalIPdus that transport the events, the method call and the method response are referenced by the SoConIPduIdentifier that assigns a headerId to each ISignalIPdu.

The event group EG2 contains Event1 and Event2 and the PduActivationRoutingGroup named EG2_CSI_25_1 activates the SoConIPduIdentifiers that are representing these two events for transmission over IP Unicast since the eventGroupControlType is set to activationUnicast.

The event group EG1 contains Event1 and the PduActivationRoutingGroup named EG1_CSI_25_1 activates the SoConIPduIdentifier for transmission over IP Multicast since the eventGroupControlType is set to activationMulticast.

In the SoAd configuration such a System Description will result in a SoAdSocketConnectionGroup with a SoAdSocketLocalPort and SoAdSocketLocalAddressRef that are derived from the ApplicationEndpoint referenced in the localUnicastAddress role by the ConsumedServiceInstance.

The SoAdSocketConnectionGroup will contain one single SoAdSocketConnection since the maxNumberOfConnections is not set in this example. The ProvidedServiceInstance does not contain a remoteUnicastAddress reference. This means that Service Discovery is used and that the SoAdSocketRemoteIpAddress needs to be configured to ANY and the SoAdSocketRemotePort to 0 in the SoAdSocketConnection.

Since only a single multicast placeholder ApplicationEndpoint is referenced with the eventMulticastAddress by all existing ConsumedEventGroups one multicast SoAdSocketConnectionGroup needs to be created with two SoAdSocketConnections.

6.7.4.5.2.1 ConsumedServiceInstance with multicast only reception

A ProvidedServiceInstance may be configured to directly use multicast for the transport of the Events of an EventGroup (EventHandler.multicastThreshold = 1). In this case it is not required for a Client to have a unicast socket prepared if the server will always use the multicast transport.

In order to take benefit of a “multicast only event transport” on Client side, the Client needs the system knowledge that a specific EventGroup will be transported by the Server using multicast only. Such an approach weakens the service discovery paradigm / disentangle approach, where Client and Server do not need to know the transport details. However, if there is the knowledge available, the Clients can be configured for “multicast only event transport”.

Note that a Server that is not configured to provide “only multicast event transport” for an EventGroup will reject a subscription with no unicast option configured, during runtime.

While the configuration of "multicast only event transport" on the ProvidedServiceInstance / EventHandler is obvious, the setup on the ConsumedServiceInstance has several implications.

The indication that a ConsumedEventGroup shall be received using multicast only transport is given by having only ConsumedEventGroup.pduActivationRoutingGroup aggregated where PduActivationRoutingGroup.eventGroupControlType is only set to activationMulticast.

[TPS_SYST_02302] Definition of multicast only reception of an EventGroup ⌈If a ConsumedEventGroup has aggregated one or several PduActivationRoutingGroups in the role pduActivationRoutingGroup and all of these PduActivationRoutingGroups have PduActivationRoutingGroup.eventGroupControlType set to activationMulticast then this ConsumedEventGroup is defined to only receive Events via multicast.⌋

[TPS_SYST_02244] ConsumedServiceInstance without a defined localUnicastAddress ⌈If a ConsumedServiceInstance does not have a defined localUnicastAddress then the events will be received over IP Multicast only.⌋

If the ConsumedServiceInstance receives events over IP Multicast only and does not define any methods (i.e. the ConsumedServiceInstance does not aggregate any PduActivationRoutingGroups in the role methodActivationRoutingGroup) that request the unicast configuration then the ConsumedServiceInstance.localUnicastAddress reference can be skipped as described by [TPS_SYST_02244]. In this case the SOME/IP message SubscribeEventGroup will not contain any endpoint options. But since the ConsumedServiceInstance is created for a specific EcuInstance and the localUnicastAddress is missing the connection of the ConsumedServiceInstance to a specific EcuInstance needs to be established in a different way. In this case the ApplicationEndpoint that is referenced by the eventMulticastAddress reference from the ConsumedEventGroup shall be EcuInstance specific. In other words the SocketAddress that contains the referenced ApplicationEndpoint shall reference only a single EthernetCommunicationConnector in the multicastConnector role.

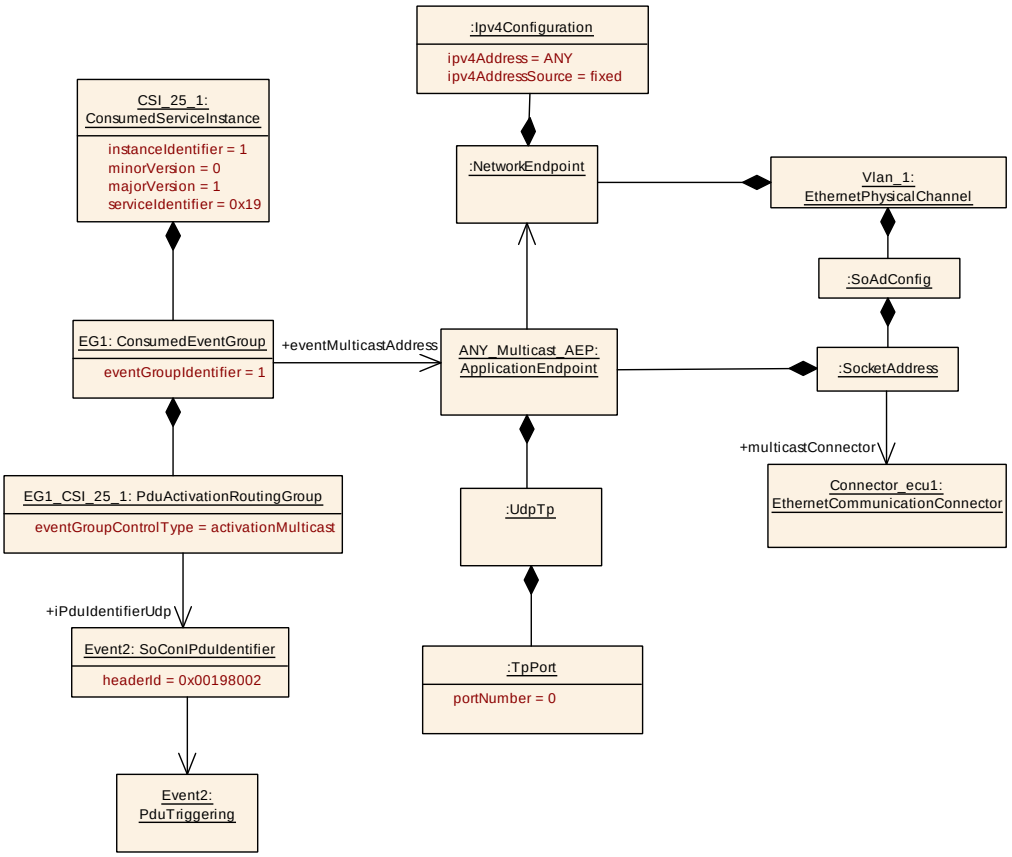

Figure 6.49: Example for the modeling of a `ConsumedServiceInstance` that is receiving Events over multicast only

6.7.4.5.3 SOME/IP Service Discovery Server Configuration

For every `ProvidedServiceInstance` on a Server different phases are existing where a suitable Service Discovery Message sending behavior is configurable:

- Down
- Available
  - Initial Wait Phase
  - Repetition Phase
  - Main Phase

[TPS_SYST_02249] Service Discovery Message sending behavior on `ProvidedServiceInstance` ⌈The Service Discovery Message sending behavior on a `ProvidedServiceInstance` is configurable with the `SomeipSdServerServiceInstanceConfig` that is referenced in the role `sdServerTimerConfig`.⌋

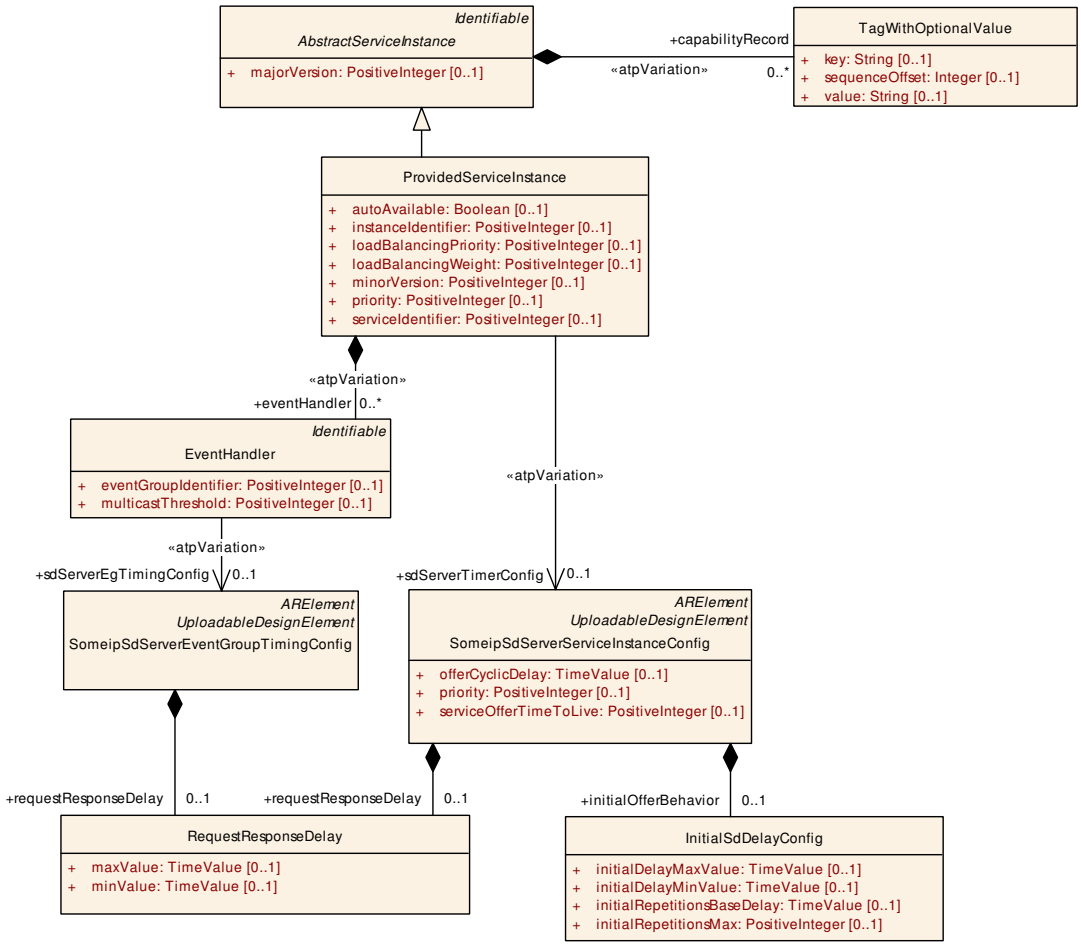

Figure 6.50: Model of SD Server Timing

Class	**SomeipSdServerServiceInstanceConfig**			
Note	Server specific settings that are relevant for the configuration of SOME/IP Service-Discovery. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialOffer Behavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server. **Tags:** atp.Splitkey=initialOfferBehavior
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds) and greater then 0.
priority	PositiveInteger	0..1	attr	This attribute defines the VLAN frame priority for Service Discovery messages that result from ProvidedSomeip ServiceInstances that are referencing the SomeipSd ServerServiceInstanceConfig (OfferService, StopOffer Service, SubscribeEventGroupAck). Values from 0 (best effort) to 7 (highest) are allowed.

△

Class	SomeipSdServerServiceInstanceConfig			
requestResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds. The Service Discovery shall delay answers to entries that were transported in a multicast SOME/IP-SD message (e.g. FindService). **Tags:** atp.Splitkey=requestResponseDelay
serviceOfferTimeToLive	PositiveInteger	0..1	attr	Defines the time in seconds the service offer is valid.

Table 6.151: SomeipSdServerServiceInstanceConfig

[constr_5382] Relation between the value of attributes offerCyclicDelay and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈In the context of any given SomeipSdServerServiceInstanceConfig, if the value of attribute offerCyclicDelay exists, it shall be less or equal to the value of attribute serviceOfferTimeToLive.⌋

[constr_5383] Relation between the value of attributes initialRepetitionsBaseDelay and initialRepetitionsMax and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈In the context of any given SomeipSdServerServiceInstanceConfig, if the value of attribute initialRepetitionsMax in initialOfferBehavior is greater than zero, the value of attribute serviceOfferTimeToLive shall be greater or equal to initialRepetitionsBaseDelay * 2^initialRepetitionsMax.⌋

Class	InitialSdDelayConfig			
Note	This element is used to configure the offer behavior of the server and the find behavior on the client.			
Base	*ARObject*			
Aggregated by	SomeipSdClientServiceInstanceConfig.initialFindBehavior, SomeipSdServerServiceInstanceConfig.initialOfferBehavior			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialDelayMaxValue	TimeValue	0..1	attr	Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).
initialDelayMinValue	TimeValue	0..1	attr	Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig).
initialRepetitionsBaseDelay	TimeValue	0..1	attr	The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay.
initialRepetitionsMax	PositiveInteger	0..1	attr	Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).

Table 6.152: InitialSdDelayConfig

Class	RequestResponseDelay			
Note	Time to wait before answering the query.			
Base	*ARObject*			
Aggregated by	SomeipSdClientEventGroupTimingConfig.requestResponseDelay, SomeipSdServerEventGroupTimingConfig.requestResponseDelay, SomeipSdServerServiceInstanceConfig.requestResponseDelay			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxValue	TimeValue	0..1	attr	Maximum allowable response delay to entries received by multicast in seconds.
minValue	TimeValue	0..1	attr	Minimum allowable response delay to entries received by multicast in seconds.

Table 6.153: RequestResponseDelay

[TPS_SYST_02174] Initial Wait Phase configuration for a ProvidedServiceInstance ⌈The Initial Wait Phase for a ProvidedServiceInstance is configured with the initialOfferBehavior and the two attributes initialDelayMinValue and initialDelayMaxValue.⌋

When a calculated random timer based on these min and max values expires, the first OfferService message will be sent out.

[constr_9215] Existence of InitialSdDelayConfig.initialDelayMaxValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdServerServiceInstanceConfig in the role initialOfferBehavior, the attribute initialDelayMaxValue shall exist.⌋

[constr_9216] Existence of InitialSdDelayConfig.initialDelayMinValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each InitialSdDelayConfig that is aggregated by a SomeipSdServerServiceInstanceConfig in the role initialOfferBehavior, the attribute initialDelayMinValue shall exist.⌋

[TPS_SYST_02258] Shared random timer for ProvidedServiceInstance service discovery ⌈If several ProvidedServiceInstances reference the same SomeipSdServerServiceInstanceConfig in the role of sdServerTimerConfig and if it is ensured that all ProvidedServiceInstances are requested / released in the same point in time (e.g. assigned to the same ConsumedProvidedServiceInstanceGroup and referenced only by one ConsumedProvidedServiceInstanceGroup, or all referencing ProvidedServiceInstances has set autoAvailable to TRUE and no runtime changes to the availability is performed), then the timing behavior is shared between the referencing ProvidedServiceInstances. Thus, if the calculated random timer based on the min and max values of the InitialSdDelayConfig expires, then OfferService entries of all referencing ProvidedServiceInstances shall be sent out.⌋

[TPS_SYST_02259] Shared random timer for `ProvidedServiceInstance` service discovery ⌈If several `ProvidedServiceInstance`s reference the same `SomeipSdServerServiceInstanceConfig` in the role of `sdServerTimerConfig` and if it cannot be ensured that all `ProvidedServiceInstance`s are requested / released in the same point in time, then the timing behavior is handled per `ProvidedServiceInstance`. The timing behavior shall be derived from the referenced `SomeipSdServerServiceInstanceConfig` for each referencing `ProvidedServiceInstance`.⌋

Note: [TPS_SYST_02259] supports an efficient modeling if timing behaviors are equal for the referencing `ProvidedServiceInstance`s.

[TPS_SYST_02260] Individual random timer for `ProvidedServiceInstance` service discovery ⌈If `ProvidedServiceInstance`s reference their own `SomeipSdServerServiceInstanceConfig` in the role of `sdServerTimerConfig`, then the timing behavior is handled per `ProvidedServiceInstance`. Thus, every `ProvidedServiceInstance` has its own calculated random timer based on the min and max value of the `InitialSdDelayConfig`. If the calculated random timer expires, then a OfferService entry of the referencing `ProvidedServiceInstance` shall be sent out.⌋

Note: [TPS_SYST_02260] enables a scattering of OfferService within the configured min and max value of the `InitialSdDelayConfig` of different `ProvidedServiceInstance`s.

[TPS_SYST_02175] Repetition Wait Phase configuration for a `ProvidedServiceInstance` ⌈The Repetition Wait Phase for a `ProvidedServiceInstance` is configured with the `initialOfferBehavior` and the two attributes `initialRepetitionsMax` and `initialRepetitionsBaseDelay`.⌋

If the Repetition Phase is entered the Service Discovery waits for the `initialRepetitionsBaseDelay` and transmits an `OfferService` entry. If the amount of sent `OfferService` entries reaches `initialRepetitionsMax` the Main Phase will be entered.

If `initialRepetitionsMax` is configured to 0 the Repetition Phase will be skipped and the Main Phase will be entered.

[TPS_SYST_02176] Main Phase configuration for a `ProvidedServiceInstance` ⌈The Main Phase for a `ProvidedServiceInstance` is configured with the `offerCyclicDelay` attribute of `SomeipSdServerServiceInstanceConfig`.⌋

The `OfferService` entry will be sent cyclically with an interval that is defined by the value of attribute `offerCyclicDelay`.

[TPS_SYST_02177] TTL for Offer Service Entries ⌈The lifetime of a `ProvidedServiceInstance` is configurable with the `serviceOfferTimeToLive` attribute of `SomeipSdServerServiceInstanceConfig`.

If the time that is configured by serviceOfferTimeToLive expires the ProvidedServiceInstance will no longer be offered.

If the offerCyclicDelay attribute of SomeipSdServerServiceInstanceConfig is not configured, or the configured value of offerCyclicDelay is 0, then the configured value of serviceOfferTimeToLive shall be ignored and the infinite value (0xFFFF) shall be used instead.⌋

[constr_9217] Existence of SomeipSdServerServiceInstanceConfig.serviceOfferTimeToLive

Imposition time: CP: IT_SysDesc

⌈For each SomeipSdServerServiceInstanceConfig, the attribute serviceOfferTimeToLive shall exist.⌋

[TPS_SYST_02178] Servers RequestResponseDelay for received FindService entries ⌈The Server will delay the OfferService answer to a received multicast FindService entry by the configured SomeipSdServerServiceInstanceConfig.requestResponseDelay.

The actual delay will be randomly chosen between the maxValue and minValue.⌋

[constr_9218] Existence of RequestResponseDelay.minValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdServerServiceInstanceConfig in the role requestResponseDelay, the attribute minValue shall exist.⌋

[constr_9219] Existence of RequestResponseDelay.maxValue aggregated by SomeipSdServerServiceInstanceConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdServerServiceInstanceConfig in the role requestResponseDelay, the attribute maxValue shall exist.⌋

The ProvidedServiceInstance aggregates an EventHandler in the role eventHandler that allows to define service instance specific configuration settings for a SOME/IP EventGroup. The EventGroup specific timing settings are configured in the SomeipSdServerEventGroupTimingConfig that is referenced by the EventHandler in the role sdServerEgTimingConfig.

Class	SomeipSdServerEventGroupTimingConfig			
Note	EventGroup specific timing configuration settings. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request ResponseDelay	RequestResponseDelay	0..1	aggr	The Service Discovery shall delay answers to unicast messages triggered by multicast messages (e.g. Subscribe Eventgroup after Offer Service). **Tags:** atp.Splitkey=requestResponseDelay

Table 6.154: SomeipSdServerEventGroupTimingConfig

[TPS_SYST_02182] Servers `RequestResponseDelay` for received `SubscribeEventGroup` entries ⌈The Server will delay the `SubscribeEventGroupAck` answer to a received `SubscribeEventGroup` message that was triggered by a multicast `ServiceOffer` by the configured `SomeipSdServerEventGroupTimingConfig.requestResponseDelay` that is referenced by the `EventHandler` in the role `sdServerEgTimingConfig`.

The actual delay will be randomly chosen between the `maxValue` and `minValue`.⌋

[constr_9220] Existence of `RequestResponseDelay.minValue` aggregated by `SomeipSdServerEventGroupTimingConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `RequestResponseDelay` that is aggregated by a `SomeipSdServerEventGroupTimingConfig` in the role `requestResponseDelay`, the attribute `minValue` shall exist.⌋

[constr_9221] Existence of `RequestResponseDelay.maxValue` aggregated by `SomeipSdServerEventGroupTimingConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `RequestResponseDelay` that is aggregated by a `SomeipSdServerEventGroupTimingConfig` in the role `requestResponseDelay`, the attribute `maxValue` shall exist.⌋

6.7.4.5.4 SOME/IP Service Discovery Client Configuration

For every `ConsumedServiceInstance` on a Client different phases are existing:

- Down
- Requested
  - Initial Wait Phase
  - Repetition Phase

– Main Phase

[TPS_SYST_02250] Service Discovery Message sending behavior on ConsumedServiceInstance ⌈The Service Discovery Message sending behavior on a ConsumedServiceInstance is configurable with the SomeipSdClientServiceInstanceConfig that is referenced in the role sdClientTimerConfig.⌋

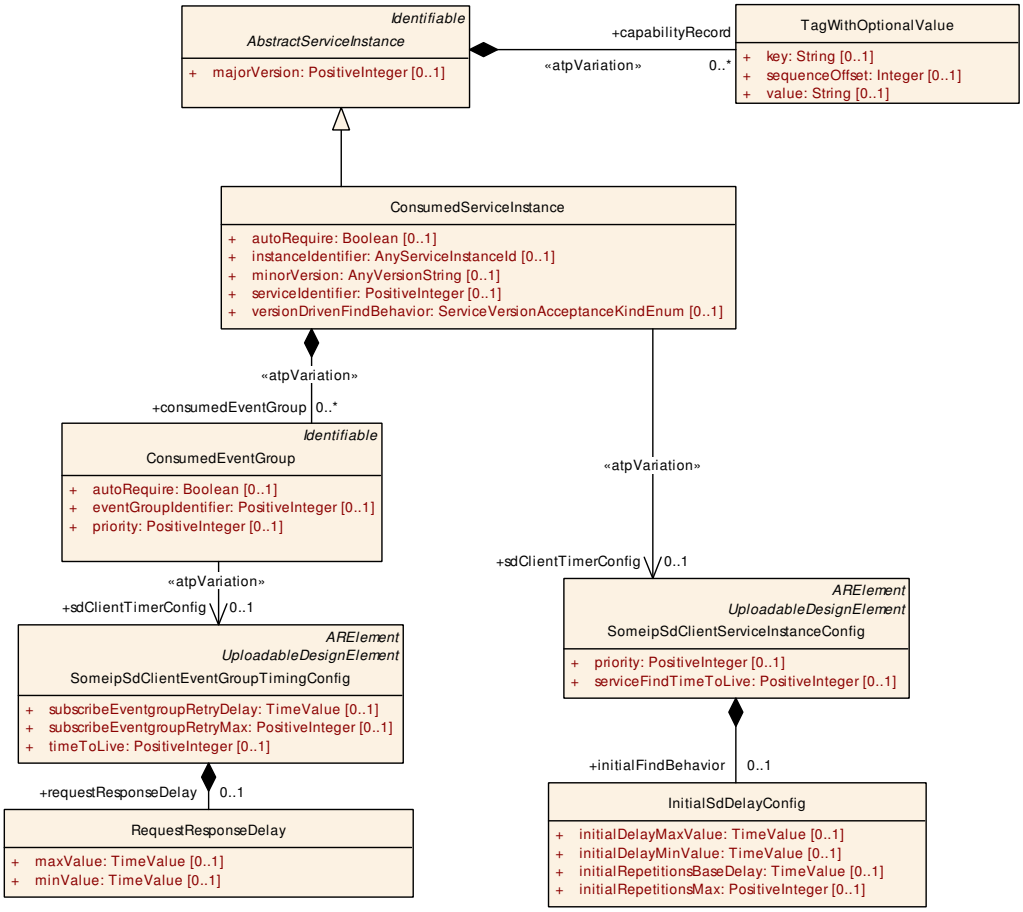

Figure 6.51: Model of SD Client Timing

[TPS_SYST_02183] Initial Wait Phase configuration for a ConsumedServiceInstance ⌈The Initial Wait Phase for a ConsumedServiceInstance is configured with the initialFindBehavior and the two attributes initialDelayMinValue and initialDelayMaxValue.⌋

If a calculated random timer based on these min and max values expires the first FindService entry will be sent out. When the calculated random timer expires and no OfferService is received the Repetition Phase will be entered.

[constr_9210] Existence of `InitialSdDelayConfig.initialDelayMaxValue` aggregated by `SomeipSdClientServiceInstanceConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `InitialSdDelayConfig` that is aggregated by a `SomeipSdClientServiceInstanceConfig` in the role `initialFindBehavior`, the attribute `initialDelayMaxValue` shall exist.⌋

[constr_9211] Existence of `InitialSdDelayConfig.initialDelayMinValue` aggregated by `SomeipSdClientServiceInstanceConfig`

Imposition time: CP: `IT_SysDesc`

⌈For each `InitialSdDelayConfig` that is aggregated by a `SomeipSdClientServiceInstanceConfig` in the role `initialFindBehavior`, the attribute `initialDelayMinValue` shall exist.⌋

[TPS_SYST_02261] Shared random timer for `ConsumedServiceInstance` service discovery ⌈If several `ConsumedServiceInstance`s reference the same `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig` and if it is ensured that all `ConsumedServiceInstance`s are requested / released in the same point in time (e.g. assigned to the same `ConsumedProvidedServiceInstanceGroup` and referenced only by one `ConsumedProvidedServiceInstanceGroup`, or all referencing `ConsumedServiceInstance` has set `autoRequire` to TRUE and no runtime changes to the require state is performed), then the timing behavior is shared between the referencing `ConsumedServiceInstance`s. Thus, if the calculated random timer based on the min and max values of the `InitialSdDelayConfig` expires, then FindService entries of all referencing `ConsumedServiceInstance`s shall be sent out.⌋

[TPS_SYST_02262] Shared random timer for `ConsumedServiceInstance` service discovery ⌈If several `ConsumedServiceInstance`s reference the same `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig` and if it cannot be ensured that all `ConsumedServiceInstance`s are requested / released in the same point in time, then the timing behavior is handled per `ConsumedServiceInstance`. The timing behavior shall be derived from the referenced `SomeipSdClientServiceInstanceConfig` for each referencing `ConsumedServiceInstance`.⌋

Note: [TPS_SYST_02262] support an efficient modeling if timing behaviors are equal for the referencing `ConsumedServiceInstance`s.

[TPS_SYST_02263] Individual random timer for `ConsumedServiceInstance` service discovery ⌈If `ConsumedServiceInstance`s reference their own `SomeipSdClientServiceInstanceConfig` in the role of `sdClientTimerConfig`, then the timing behavior is handled per `ConsumedServiceInstance`. Thus, every `ConsumedServiceInstance` has its own calculated random timer based on the min and max value of the `InitialSdDelayConfig`. Thus, if the calculated random timer

expires, then a FindService entry of the referencing ConsumedServiceInstance shall be sent out.⌋

Note: [TPS_SYST_02263] enables a scattering of FindServices within the configured min and max value of the InitialSdDelayConfig, respectively of different ConsumedServiceInstances.

[TPS_SYST_02184] Repetition Wait Phase configuration for a ConsumedServiceInstance ⌈The Repetition Wait Phase for a ConsumedServiceInstance is configured with the initialFindBehavior and the two attributes initialRepetitionsMax and initialRepetitionsBaseDelay.⌋

If the Repetition Phase is entered, the Service Discovery waits the initialRepetitionsBaseDelay and sends a FindService entry.

If the amount of sent FindService entries reaches initialRepetitionsMax and no OfferService is received the Main Phase will be entered. In the Main Phase no further FindService entries are sent by the client.

[TPS_SYST_02185] TTL for Find Service Entries ⌈The lifetime of a ConsumedServiceInstance is configurable with the serviceFindTimeToLive attribute of SomeipSdClientServiceInstanceConfig. Note! The TTL value for FindService entries shall be ignored by the server service, and the configuration is only kept for backward compatibility.⌋

The ConsumedServiceInstance aggregates a ConsumedEventGroup in the role consumedEventGroup that allows to define service instance specific configuration settings for a SOME/IP EventGroup. The EventGroup specific timing settings are configured in the SomeipSdClientEventGroupTimingConfig that is referenced by the ConsumedEventGroup in the role sdClientTimerConfig.

Class	**SomeipSdClientEventGroupTimingConfig**			
Note	This meta-class is used to specify configuration related to service discovery in the context of an event group on SOME/IP. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
request ResponseDelay	RequestResponseDelay	0..1	aggr	The Service Discovery shall delay answers to unicast messages triggered by multicast messages (e.g. Subscribe Eventgroup after Offer Service). **Tags:** atp.Splitkey=requestResponseDelay
subscribe Eventgroup RetryDelay	TimeValue	0..1	attr	This attribute defines the interval in seconds to re-trigger a subscription to a Eventgroup, if a retry to subscribe to a Eventgroup is configured (subscribeEventgroupRetryMax > 0).

▽

△

Class	SomeipSdClientEventGroupTimingConfig			
subscribe Eventgroup RetryMax	PositiveInteger	0..1	attr	This attribute define the maximum counts of retries to subscribe to an Eventgroup. If the value is set to 0 no retry shall be done. If the value is set to 255 the retry shall be done as along as the Eventgroup is requested and no SubscribeEventGroupAck was received.
timeToLive	PositiveInteger	0..1	attr	Defines the time in seconds the subscription of this event is expected by the client. this value is sent from the client to the server in the SD-subscribeEvent message.

Table 6.155: SomeipSdClientEventGroupTimingConfig

[TPS_SYST_02187] **SomeipSdClientEventGroupTimingConfig.timeToLive for SubscribeEventGroup Entries** ⌈The lifetime of an event subscription is configurable with the timeToLive attribute of SomeipSdClientEventGroupTimingConfig that is aggregated by an ConsumedEventGroup in the role sdClientTimerConfig.

If the time that is configured by timeToLive expires the event subscription is canceled.⌋

[constr_9212] Existence of SomeipSdClientEventGroupTimingConfig.timeToLive

Imposition time: CP: IT_SysDesc

⌈For each SomeipSdClientEventGroupTimingConfig, the attribute timeToLive shall exist.⌋

It is possible to define a retry for subscription to a ConsumedEventGroup. The retry is optional and used by a ConsumedServiceInstance. It could be used to speed up the recovery if a SOME/IP-SD message is lost (e.g. SubscribeEventGroupAck) and the interval between cyclic offers (SomeipSdServerServiceInstanceConfig.offerCyclicDelay) of the corresponding ProvidedServiceInstance are to large to get a fast recovery, or to speed up the subscription to a ConsumedEventGroup if an ConsumedEventGroup is requested somewhere between two cyclic offers.

The retry is configurable within SomeipSdServerServiceInstanceConfig by setting the subscribeEventgroupRetryMax value greater than 0 to define how often a retry shall be triggered and subscribeEventgroupRetryDelay to define the timing interval the retries shall be triggered. If subscribeEventgroupRetryMax is set to 255, the retry to subscribe to a ConsumedEventGroup shall be done as long as the ConsumedEventGroup is requested and no acknowledgment (SubscribeEventGroupAck) from the ProvidedServiceInstance was received.

[constr_5095] Relationship between the timing behavior of the ConsumedEventGroup retry and the timing behavior of an Offer message

Imposition time: CP: IT_SysDesc

⌈The timing behavior for a retry to a ConsumedEventGroup (subscribeEventgroupRetryMax, subscribeEventgroupRetryDelay) shall not overlap to the timing behavior (SomeipSdServerServiceInstanceConfig.offerCyclicDelay) of the corresponding ProvidedServiceInstance.⌋

[constr_5096] ConsumedEventGroup with value subscribeEventgroupRetryMax set to 255

Imposition time: CP: IT_SysDesc

⌈Retry to a ConsumedEventGroup with value subscribeEventgroupRetryMax set to 255 is only allowed if the SomeipSdServerServiceInstanceConfig.offerCyclicDelay is set 0 and serviceOfferTimeToLive is set to 0xffffff of the corresponding ProvidedServiceInstance.⌋

[TPS_SYST_02188] Clients RequestResponseDelay for received ServiceOffer entries ⌈The Client will delay the SubscribeEventGroup answer to a received ServiceOffer message by the configured SomeipSdClientEventGroupTimingConfig.requestResponseDelay that is aggregated by the ConsumedEventGroup in the role sdClientTimerConfig.

The actual delay will be randomly chosen between the maxValue and minValue.⌋

[constr_9213] Existence of RequestResponseDelay.minValue aggregated by SomeipSdClientEventGroupTimingConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdClientEventGroupTimingConfig in the role requestResponseDelay, the attribute minValue shall exist.⌋

[constr_9214] Existence of RequestResponseDelay.maxValue aggregated by SomeipSdClientEventGroupTimingConfig

Imposition time: CP: IT_SysDesc

⌈For each RequestResponseDelay that is aggregated by a SomeipSdClientEventGroupTimingConfig in the role requestResponseDelay, the attribute maxValue shall exist.⌋

6.7.4.5.5 Group of ConsumedServiceInstances and ProvidedServiceInstances

The AUTOSAR ServiceDiscovery provide a mechanism of starting/stopping ConsumedProvidedServiceInstanceGroups. Therefore, several ConsumedServiceInstances and ProvidedServiceInstances, respectively, could by enclosed to a ConsumedProvidedServiceInstanceGroup.

Class	**ConsumedProvidedServiceInstanceGroup**			
Note	The AUTOSAR ServiceDiscovery is able to start and to stop ClientServices and Server Services,respectively, at runtime. A SdServiceGroup contains several ClientServices and Server Services, respectively. **Tags:** atp.recommendedPackage=ConsumedProvidedServiceInstanceGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
consumed ServiceInstance	ConsumedService Instance	*	ref	This reference assigns a set of ProvidedServiceInstances to the ConsumedProvidedServiceInstanceGroup. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedServiceInstance, consumed ServiceInstance.consumedServiceInstance, consumed ServiceInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild
providedService Instance	ProvidedService Instance	*	ref	This reference assigns a set of ConsumedService Instances to the ConsumedProvidedServiceInstance Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedServiceInstance, providedService Instance.providedServiceInstance, providedService Instance.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.156: ConsumedProvidedServiceInstanceGroup

The `ConsumedProvidedServiceInstanceGroup` is mapped to a `SdService-Group` within the ECUC configuration. The `SdServiceGroup`s could be accessed via dedicated APIs within the AUTOSAR ServiceDiscovery (Sd) module to start and stop the requested `SdServiceGroup`. For example, the `SdServiceGroup`s could be switched according the requested partial network similar to `ComISignalIPduGroup` and therefore the `PncMapping` refers the `ConsumedProvidedServiceInstance-Group` in the role `pncConsumedProvidedServiceInstanceGroup`.

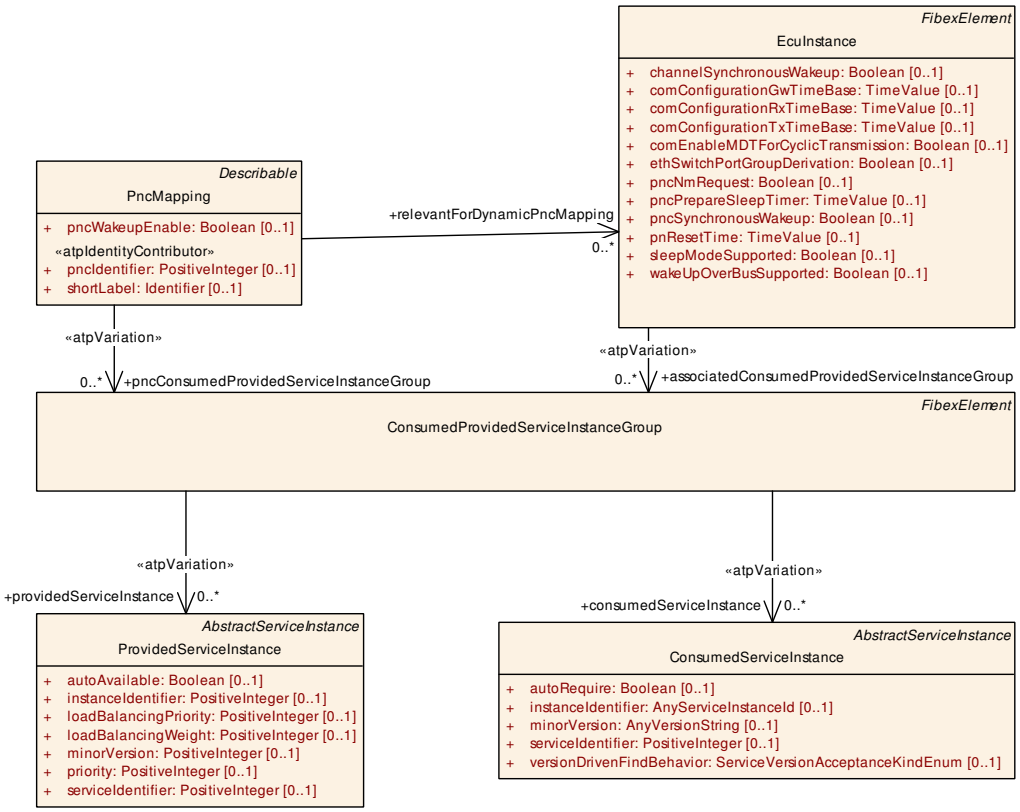

Figure 6.52: ConsumedProvidedServiceInstanceGroup

Starting a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ConsumedServiceInstance`s to sent out *FindService* entries on the network and stopping a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ConsumedServiceInstance`s to unsubscribe from the subscribed Event Groups and sent out an unsubscribe entry on the network.

Starting a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ProvidedServiceInstance`s to sent out *OfferService* entries on the network and stopping a `ConsumedProvidedServiceInstanceGroup` would trigger the referencing `ProvidedServiceInstance`s to sent out a *StopOffer* entries on the network.

6.7.4.6 DDS Specific Description

In order to simplify the DDS configuration at EcuC level, the DDS-related model elements have been integrated into this document:

the DDS configuration at SystemTemplate level shall be used the derive the EcuC configuration of the DDS Entities, e.g. DDS DomainParticipants, Publishers, Subscribers, DataWriters, DataReaders.

DDS-specific model elements can be detailed both for **DDS signal-based communications** (a single `ISignal`, with `dataTypePolicy` set to `ddsSignal`, used to

carry one DDS data, by means of DdsCpISignalToDdsTopicMapping) and for **DDS service-oriented communications** (by means of DdsCpServiceInstances, for details, refer to section 6.7.4.6.16).

Those model elements are grouped together by the DdsCpConfig, then they are linked to a ISignal or to a DdsCpServiceInstance by a reference to a DdsCp-Topic.

The DDS supports both unicast and multicast traffic in the data-centric communication paradigm and in the service-oriented communications paradigm. This support is extended to:

- static networks, in which all endpoints are known at configuration time
- dynamic networks, in which standard OMG discovery protocols ([31, OMG DDS-RTPS specification]) are used at run-time for endpoints discovery.

At system level, user traffic support is currently limited to the unicast communication paradigm.

6.7.4.6.1 DDS Common Parameters

The DdsCpConfig class is a class used to group all the DDS relevant parameters.

At top level, a DDS Domain, characterized by its ID, is defined. A DDS Domain is used to isolate applications to each others: only entities belonging to the same DDS Domain can communicate with each other.

DDS Topics are part of a DDS Domain: each DDS Topic has visibility only inside a specific Domain, so Topics can be defined as part of a Domain.

A DDS Partition is used to logically group several Topics: it is possible to assign each Topic to one or more partitions (used to logically sub-partition a Domain). A Topic is characterized by a set of QoS policies.

The sections below give some details of the DdsCpConfig (and its sub-containers) architecture and the DdsCpQosProfile container.

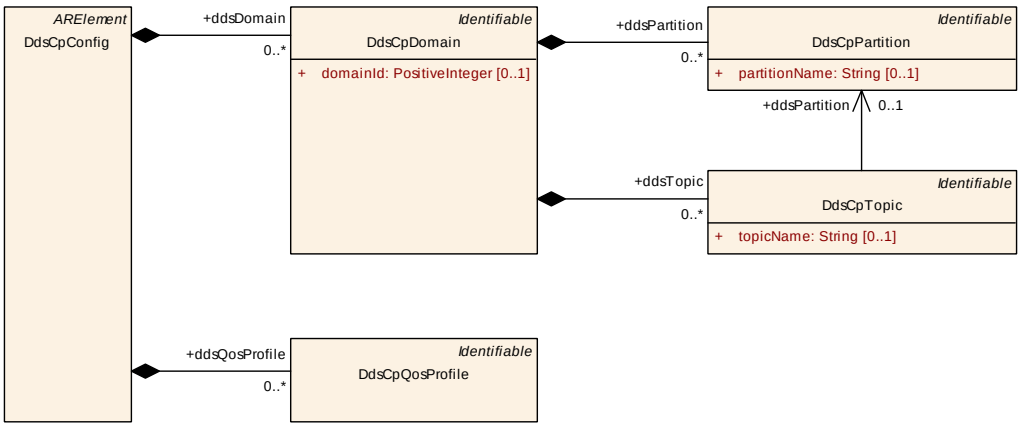

Figure 6.53: Dds Common parameters

Class	DdsCpConfig			
Note	Collection of DDS definitions. **Tags:** atp.Status=candidate atp.recommendedPackage=DdsCpConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ddsDomain	DdsCpDomain	*	aggr	Collection of DDS Domain definitions. **Tags:** atp.Splitkey=ddsDomain, ddsDomain.shortName atp.Status=candidate
ddsQosProfile	DdsCpQosProfile	*	aggr	Collection of DDS QOS Profiles. **Tags:** atp.Splitkey=ddsQosProfile, ddsQosProfile.shortName atp.Status=candidate

Table 6.157: DdsCpConfig

[TPS_SYST_03086] DdsDomain ⌈The `DdsCpDomain` container is used to model a DDS Domain.⌋

For details about DDS Domain, refer to the chapter "Domain Module" of the DDS OMG specification ([32]).

Class	DdsCpDomain			
Note	Definition of a DDS Domain. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConfig.ddsDomain			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ddsPartition	DdsCpPartition	*	aggr	Collection of DDS Partition definitions. **Tags:** atp.Splitkey=ddsPartition, ddsPartition.shortName atp.Status=candidate
ddsTopic	DdsCpTopic	*	aggr	Collection of DDS Topics. **Tags:** atp.Splitkey=ddsTopic, ddsTopic.shortName atp.Status=candidate
domainId	PositiveInteger	0..1	attr	Definition of the DDS Domain Id. **Tags:** atp.Status=candidate

Table 6.158: DdsCpDomain

[TPS_SYST_03087] DdsTopic ⌈The `DdsCpTopic` container is used to model a DDS Topic.⌋

[constr_3826] `DdsCpTopic.ddsPartition` reference restriction

Imposition time: CP: `IT_EcuExt`

⌈The `DdsCpTopic` that is aggregated by a `DdsCpDomain` in the role `ddsTopic` is only allowed to reference a `DdsCpPartition` in the role `ddsPartition` that is aggregated by the same `DdsCpDomain`.⌋

For details about DDS Topic, refer to the chapter "Topic Definition Module" of the DDS OMG specification ([32]).

Class	**DdsCpTopic**			
Note	Definition of a DDS Topic. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpDomain.ddsTopic			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsPartition	DdsCpPartition	0..1	ref	Reference to the DDS Partition this topic is communicated. **Tags:** atp.Splitkey=ddsPartition atp.Status=candidate
topicName	String	0..1	attr	Definition of the DDS Topic Name. **Tags:** atp.Status=candidate

Table 6.159: DdsCpTopic

[TPS_SYST_03088] DdsPartition ⌈The `DdsCpPartition` container is used to model a DDS Partition. Note that a DDS Partition is defined by using PARTITION QoS.⌋

For details about DDS Partition, refer to the chapter "PARTITION" of the DDS OMG specification ([32]).

Class	**DdsCpPartition**			
Note	Definition of a DDS Partition. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpDomain.ddsPartition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
partitionName	String	0..1	attr	Definition of the DDS Partition Name. '*' may be used to define the default partition. **Tags:** atp.Status=candidate

Table 6.160: DdsCpPartition

[TPS_SYST_03089] DdsQoSProfile ⌈The `DdsCpQosProfile` container is used to model a DDS QoS Profile. It shall be defined in such a way to guarantee compliance to the OMG standard (DDS Consolidated XML Syntax).⌋

For details about DDS QoS policies, refer to the chapter "Supported QoS" of the DDS OMG specification ([32]).

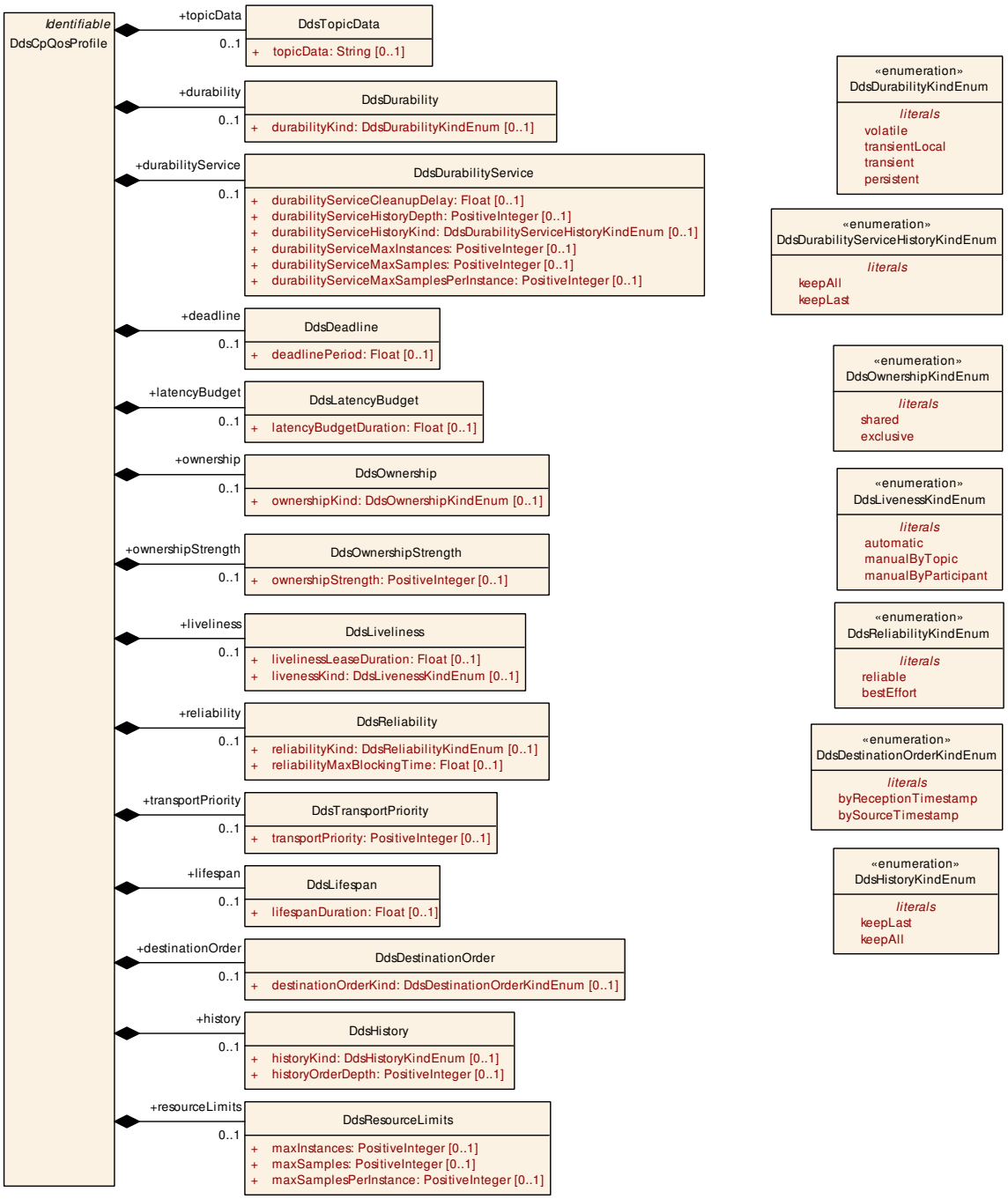

Figure 6.54: Dds QoS Profile

Class	**DdsCpQosProfile**			
Note	Definition of a DDS QOS Profile. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConfig.ddsQosProfile			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DdsCpQosProfile**			
deadline	DdsDeadline	0..1	aggr	Defines the DDS DEADLINE QoS policy. **Tags:** atp.Splitkey=deadline atp.Status=candidate
destination Order	DdsDestinationOrder	0..1	aggr	Defines the DDS DESTINATION_ORDER QoS policy. **Tags:** atp.Splitkey=destinationOrder
durability	DdsDurability	0..1	aggr	Defines the DDS DURABILITY QoS policy. **Tags:** atp.Splitkey=durability atp.Status=candidate
durability Service	DdsDurabilityService	0..1	aggr	Defines the DDS DURABILITY_SERVICE QoS policy. **Tags:** atp.Splitkey=durabilityService atp.Status=candidate
history	DdsHistory	0..1	aggr	Defines the DDS HISTORY QoS policy. **Tags:** atp.Splitkey=history
latencyBudget	DdsLatencyBudget	0..1	aggr	Defines the DDS LATENCY_BUDGET QoS policy. **Tags:** atp.Splitkey=latencyBudget atp.Status=candidate
lifespan	DdsLifespan	0..1	aggr	Defines the DDS LIFESPAN QoS policy. **Tags:** atp.Splitkey=lifespan
liveliness	DdsLiveliness	0..1	aggr	Defines the DDS LIVELINESS QoS policy. **Tags:** atp.Splitkey=liveliness atp.Status=candidate
ownership	DdsOwnership	0..1	aggr	Defines the DDS OWNERSHIP QoS policy. **Tags:** atp.Splitkey=ownership atp.Status=candidate
ownership Strength	DdsOwnershipStrength	0..1	aggr	Defines the DDS OWNERSHIP_STRENGTH QoS policy. **Tags:** atp.Splitkey=ownershipStrength atp.Status=candidate
reliability	DdsReliability	0..1	aggr	Defines the DDS RELIABILITY QoS policy. **Tags:** atp.Splitkey=reliability
resourceLimits	DdsResourceLimits	0..1	aggr	Defines the DDS RESOURCE_LIMITS QoS policy. **Tags:** atp.Splitkey=resourceLimits
topicData	DdsTopicData	0..1	aggr	Defines the DDS TOPIC_DATA QoS policy. **Tags:** atp.Splitkey=topicData
transportPriority	DdsTransportPriority	0..1	aggr	Defines the DDS TRANSPORT_PRIORITY QoS policy. **Tags:** atp.Splitkey=transportPriority

Table 6.161: DdsCpQosProfile

6.7.4.6.2 DDS QoS Topic Data

Class	**DdsTopicData**			
Note	Describes the DDS TOPIC_DATA QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.topicData			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
topicData	String	0..1	attr	See "TOPIC_DATA" chapter in DDS. **Tags:** atp.Status=candidate

Table 6.162: DdsTopicData

6.7.4.6.3 DDS QoS Durability

Class	**DdsDurability**			
Note	Describes the DDS DURABILITY QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.durability			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
durabilityKind	DdsDurabilityKindEnum	0..1	attr	See "DURABILITY" chapter in DDS. **Tags:** atp.Status=candidate

Table 6.163: DdsDurability

Enumeration	**DdsDurabilityKindEnum**
Note	Defines the DDS DURABILITY kind. **Tags:** atp.Status=candidate
Aggregated by	DdsDurability.durabilityKind
Literal	***Description***
persistent	See "DURABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=3 atp.Status=candidate
transient	See "DURABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=2 atp.Status=candidate
transientLocal	See "DURABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate



△

Enumeration	**DdsDurabilityKindEnum**
volatile	See "DURABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 6.164: DdsDurabilityKindEnum

6.7.4.6.4 DDS QoS Durability Service

Class	**DdsDurabilityService**			
Note	Describes the DDS DURABILITY_SERVICE QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.durabilityService			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
durability ServiceCleanup Delay	Float	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds. **Tags:** atp.Status=candidate
durability ServiceHistory Depth	PositiveInteger	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. **Tags:** atp.Status=candidate
durability ServiceHistory Kind	DdsDurabilityService HistoryKindEnum	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. **Tags:** atp.Status=candidate
durability ServiceMax Instances	PositiveInteger	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. **Tags:** atp.Status=candidate
durability ServiceMax Samples	PositiveInteger	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. **Tags:** atp.Status=candidate
durability ServiceMax SamplesPer Instance	PositiveInteger	0..1	attr	See "DURABILITY_SERVICE" chapter in DDS. **Tags:** atp.Status=candidate

Table 6.165: DdsDurabilityService

Enumeration	**DdsDurabilityServiceHistoryKindEnum**
Note	Defines the DDS DURABILITY_SERVICE HISTORY kind. **Tags:** atp.Status=candidate
Aggregated by	DdsDurabilityService.durabilityServiceHistoryKind
Literal	*Description*
keepAll	See "DURABILITY_SERVICE" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

▽

△

Enumeration	**DdsDurabilityServiceHistoryKindEnum**
keepLast	See "DURABILITY_SERVICE" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate

Table 6.166: DdsDurabilityServiceHistoryKindEnum

6.7.4.6.5 DDS QoS Deadline

Class	**DdsDeadline**			
Note	Describes the DDS DEADLINE QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.deadline			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
deadlinePeriod	Float	0..1	attr	See "DEADLINE" chapter of DDS. Time given in seconds. **Tags:** atp.Status=candidate

Table 6.167: DdsDeadline

6.7.4.6.6 DDS QoS Latency Budget

Class	**DdsLatencyBudget**			
Note	Describes the DDS LATENCY_BUDGET QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.latencyBudget			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
latencyBudget Duration	Float	0..1	attr	See "LATENCY_BUDGET" chapter of DDS. Time given in seconds. **Tags:** atp.Status=candidate

Table 6.168: DdsLatencyBudget

6.7.4.6.7 DDS QoS Ownership

Class	**DdsOwnership**			
Note	Describes the DDS OWNERSHIP QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.ownership			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ownershipKind	DdsOwnershipKind Enum	0..1	attr	See "OWNERSHIP" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.169: DdsOwnership

Enumeration	**DdsOwnershipKindEnum**
Note	Defines the DDS OWNERSHIP kind. **Tags:** atp.Status=candidate
Aggregated by	DdsOwnership.ownershipKind
Literal	***Description***
exclusive	See "OWNERSHIP" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate
shared	See "OWNERSHIP" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 6.170: DdsOwnershipKindEnum

6.7.4.6.8 DDS QoS Ownership Strength

Class	**DdsOwnershipStrength**			
Note	Describes the DDS OWNERSHIP_STRENGTH QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.ownershipStrength			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ownership Strength	PositiveInteger	0..1	attr	See "OWNERSHIP_STRENGTH" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.171: DdsOwnershipStrength

6.7.4.6.9 DDS QoS Liveliness

Class	**DdsLiveliness**			
Note	Describes the DDS LIVELINESS QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.liveliness			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
livelinessLease Duration	Float	0..1	attr	See "LIVELINESS" chapter of DDS. Time given in seconds. **Tags:** atp.Status=candidate
livenessKind	DdsLivenessKindEnum	0..1	attr	See "LIVELINESS" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.172: DdsLiveliness

Enumeration	**DdsLivenessKindEnum**
Note	Defines the DDS LIVELINESS kind. **Tags:** atp.Status=candidate
Aggregated by	DdsLiveliness.livenessKind
Literal	***Description***
automatic	See "LIVELINESS" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate
manualBy Participant	See "LIVELINESS" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=2 atp.Status=candidate
manualByTopic	See "LIVELINESS" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate

Table 6.173: DdsLivenessKindEnum

6.7.4.6.10 DDS QoS Reliability

Class	**DdsReliability**			
Note	Describes the DDS RELIABILITY QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.reliability			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
reliabilityKind	DdsReliabilityKindEnum	0..1	attr	See "RELIABILITY" chapter of DDS. **Tags:** atp.Status=candidate

▽

△

Class	**DdsReliability**			
reliabilityMax BlockingTime	Float	0..1	attr	See "RELIABILITY" chapter of DDS. Time given in seconds. **Tags:** atp.Status=candidate

Table 6.174: DdsReliability

Enumeration	**DdsReliabilityKindEnum**
Note	See "RELIABILITY" chapter of DDS. **Tags:** atp.Status=candidate
Aggregated by	DdsReliability.reliabilityKind
Literal	***Description***
bestEffort	See "RELIABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate
reliable	See "RELIABILITY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 6.175: DdsReliabilityKindEnum

6.7.4.6.11 DDS QoS Transport Priority

Class	**DdsTransportPriority**			
Note	Describes the DDS TRANSPORT_PRIORITY QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.transportPriority			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
transportPriority	PositiveInteger	0..1	attr	See "TRANSPORT_PRIORITY" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.176: DdsTransportPriority

6.7.4.6.12 DDS QoS Lifespan

Class	**DdsLifespan**
Note	Describes the DDS LIFESPAN QoS policy. **Tags:** atp.Status=candidate
Base	*ARObject*
Aggregated by	DdsCpQosProfile.lifespan

▽

△

Class	**DdsLifespan**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
lifespanDuration	Float	0..1	attr	See "LIFESPAN" chapter of DDS. Time given in seconds. **Tags:** atp.Status=candidate

Table 6.177: DdsLifespan

6.7.4.6.13 DDS QoS Destination Order

Class	**DdsDestinationOrder**			
Note	Describes the DDS DESTINATION_ORDER QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.destinationOrder			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destination OrderKind	DdsDestinationOrder KindEnum	0..1	attr	See "DESTINATION_ORDER" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.178: DdsDestinationOrder

Enumeration	**DdsDestinationOrderKindEnum**
Note	Defines the DDS DESTINATION_ORDER kind. **Tags:** atp.Status=candidate
Aggregated by	DdsDestinationOrder.destinationOrderKind
Literal	***Description***
byReception Timestamp	See "DESTINATION_ORDER" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate
bySource Timestamp	See "DESTINATION_ORDER" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate

Table 6.179: DdsDestinationOrderKindEnum

6.7.4.6.14 DDS QoS History

Class	**DdsHistory**
Note	Describes the DDS HISTORY QoS policy. **Tags:** atp.Status=candidate
Base	*ARObject*

▽

△

Class	**DdsHistory**			
Aggregated by	DdsCpQosProfile.history			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
historyKind	DdsHistoryKindEnum	0..1	attr	See "HISTORY" chapter of DDS. **Tags:** atp.Status=candidate
historyOrder Depth	PositiveInteger	0..1	attr	See "HISTORY" chapter of DDS. **Tags:** atp.Status=candidate

Table 6.180: DdsHistory

Enumeration	**DdsHistoryKindEnum**
Note	Defines the DDS HISTORY kind. **Tags:** atp.Status=candidate
Aggregated by	DdsHistory.historyKind
Literal	*Description*
keepAll	See "HISTORY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate
keepLast	See "HISTORY" chapter of DDS. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 6.181: DdsHistoryKindEnum

6.7.4.6.15 DDS QoS Resource Limits

Class	**DdsResourceLimits**			
Note	Describes the DDS RESOURCE_LIMITS QoS policy. **Tags:** atp.Status=candidate			
Base	*ARObject*			
Aggregated by	DdsCpQosProfile.resourceLimits			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
maxInstances	PositiveInteger	0..1	attr	See "RESOURCE_LIMITS" chapter of DDS.
maxSamples	PositiveInteger	0..1	attr	See "RESOURCE_LIMITS" chapter of DDS. **Tags:** atp.Status=candidate
maxSamples PerInstance	PositiveInteger	0..1	attr	See "RESOURCE_LIMITS" chapter of DDS.

Table 6.182: DdsResourceLimits

6.7.4.6.16 DDS Service Instances

The `DdsCpServiceInstance` class is an abstract class that derives from the `AbstractServiceInstance`. It is used to define a DDS service instance and the

DDS Topics associated to the service itself. This class combines attributes for both interface and instance parameters, and it is used for AP interoperability.

From this class, the DdsCpConsumedServiceInstance and DdsCpProvidedServiceInstance classes are derived.

A DdsCpServiceInstance can contain any combination of DdsServiceInstanceEventCps and DdsServiceInstanceOperationCps, through the inherited classes DdsCpProvidedServiceInstance and DdsCpConsumedServiceInstance. For DDS, the exchange information unit is the Dds Topic. In this case, the definition of DdsCpProvidedServiceInstance and DdsCpConsumedServiceInstance is always required, even for fully static configuration: it is necessary to link ServiceInstance objects to DDS Topics, and not only to assign a ServiceId to be used during Service discovery.

Into a DdsCpServiceInstance there are different instances of DdsCpTopic to be mapped to AUTOSAR CP specific objects:

- One pair of DdsCpTopics (one for requests and one for replies) for all the Methods into the DdsCpServiceInstance: DdsCpServiceInstance.ddsMethodRequestTopic and DdsCpServiceInstance.ddsMethodReplyTopic

  All **method requests** for a single DdsCpServiceInstance are represented as a single topic that leverages DDS discriminated union types to pack method request identification (source, target, specific method ID within the service interface), along with the input parameter data.

  Similarly, all **method replies** for a single DdsCpServiceInstance are represented as a single topic (differently named from the one dedicated to requests) that leverages DDS discriminated union types to pack method reply identification (source, target, specific method ID within the service interface), along with the output parameter / error data.

- One pair of DdsCpTopics (one for requests and one for replies) for all the Fields into the DdsCpServiceInstance: DdsCpServiceInstance.ddsFieldRequestTopic and DdsCpServiceInstance.ddsFieldReplyTopic

  All **field get or set requests** for a single DdsCpServiceInstance are represented as a single topic that leverages DDS discriminated union types to pack field get/set identification (source, target, specific method ID within the service interface), along with the input parameter data.

  Similarly **all field get or set replies** for a single DdsCpServiceInstance are represented as a single topic (differently named from the one dedicated to requests) that leverages DDS discriminated union types to pack field get/set reply identification (source, target, specific method ID within the service interface), along with the output parameter / error data.

  Note: Field Notifiers behave exactly like DDS Events. They share the same topic naming conventions, type and DDS entity functionality.

- One DdsCpTopic for each DdsServiceInstanceEventCp of the DdsCp-ServiceInstance. Each **event** and each **field notifier** is represented by its own topic.

[constr_3739] Value of ISignal.dataTypePolicy for all ISignals associated with a DdsCpServiceInstance

Imposition time: CP: IT_EcuExt

⌈For all ISignals that are referenced by an ISignalToIPduMapping where the enclosing ISignalIPdu is only referenced by PduTriggerings that are in turn referenced in one of the roles:

- DdsServiceInstanceEventCp.eventTriggering
- DdsServiceInstanceOperationCp.operationRequestTriggering
- DdsServiceInstanceOperationCp.operationResponseTriggering

The value of attribute ISignal.dataTypePolicy shall be set to ddsService.⌋

In the below picture the overall architecture of DdsCpServiceInstance class is depicted.

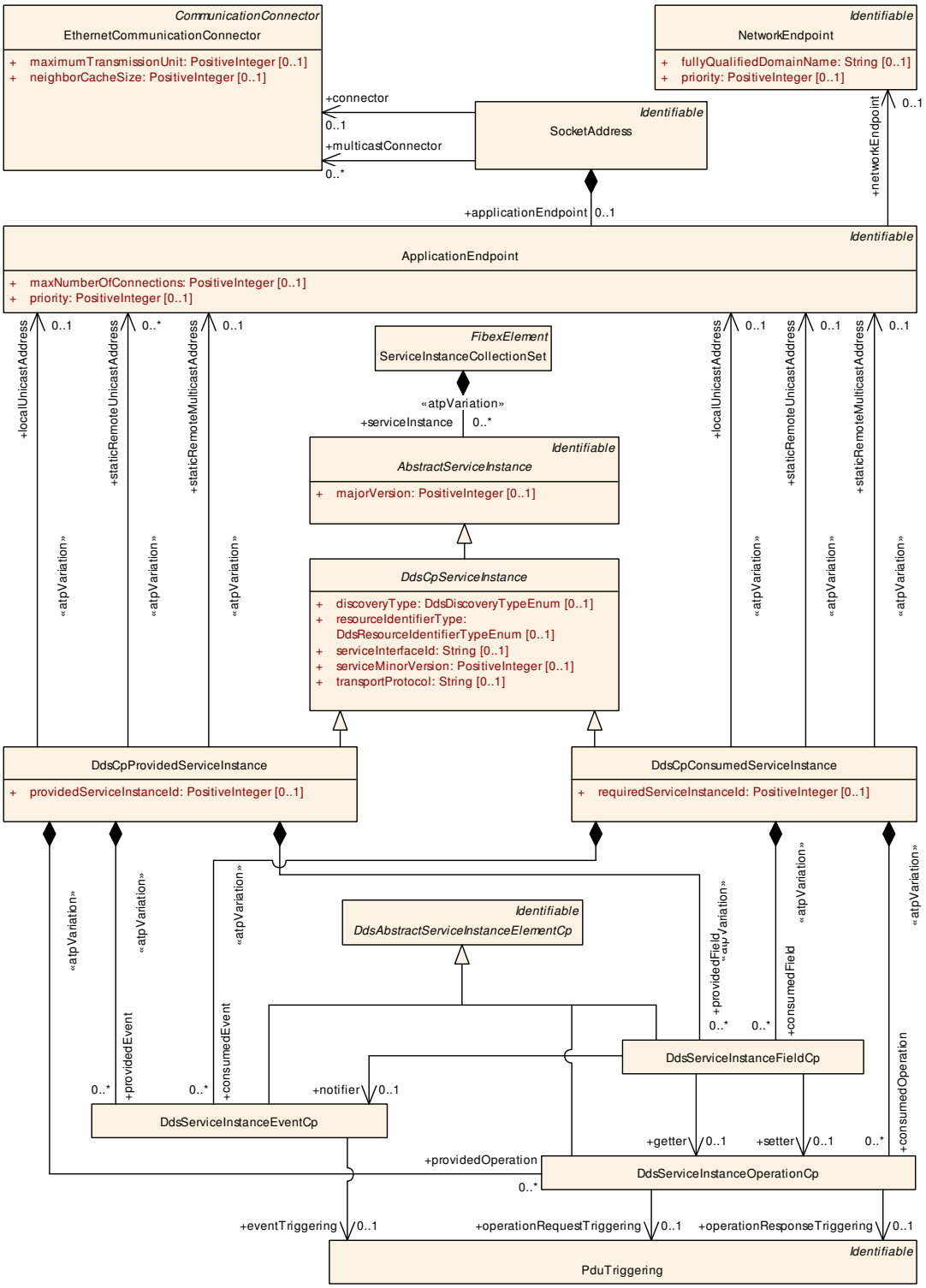

Figure 6.55: Dds Service Instance

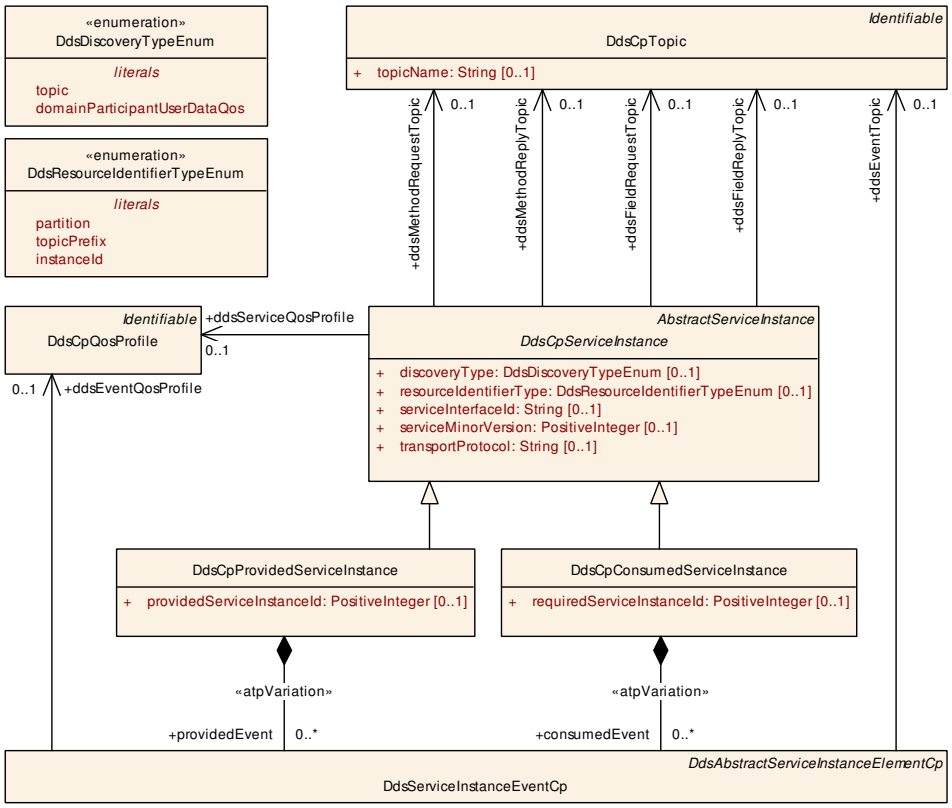

Figure 6.56: Dds Service Instance Topics

[TPS_SYST_03090] DDS Methods Request/Reply Topic ⌈The DdsCpServiceInstance.ddsMethodRequestTopic and DdsCpServiceInstance.ddsMethodReplyTopic references are used to specify the DdsCpTopic to be used respectively for Method requests and Method replies.⌋

[constr_3799] DDS Fire & Forget Method definition

Imposition time: CP: IT_EcuExt

⌈If a DdsServiceInstanceOperationCp represents a fire & forget method, then only the reference DdsServiceInstanceOperationCp.operationRequestTriggering shall exist and the reference in the role DdsServiceInstanceOperationCp.operationResponseTriggering shall not exist.⌋

[TPS_SYST_03091] DDS Fields Request/Reply Topic ⌈The DdsCpServiceInstance.ddsFieldRequestTopic and DdsCpServiceInstance.ddsFieldReplyTopic references are used to specify the DdsCpTopic to be used respectively for Fields requests and Fields replies.⌋

The DdsServiceInstanceEventCp class is used to define specific DDS parameters for events.

[constr_3740] Existence of DdsServiceInstanceEventCp.ddsEventTopic

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceEventCp, the reference in the role ddsEventTopic shall exist.⌋

[TPS_SYST_03093] DDS Operation ⌈The DdsServiceInstanceOperationCp class is used to define specific DDS parameters for Methods.⌋

[TPS_SYST_03094] DdsService QoS Profile ⌈The DdsCpQosProfile referenced in the role DdsCpServiceInstance.ddsServiceQosProfile shall be used to identify the QoS policy configuration that applies to the entire DdsCpServiceInstance.⌋

The DDS standard [32] defines QoS policies and their attributes, but not default values for the later: they are left to the final users. Besides, it is possible to have QoS profiles which are named sets of cohesive QoS policy attributes. In AUTOSAR, a DdsCpServiceInstance needs to define, at least, a QoS profile to apply to all the DDS entities that eventually represent the service events, methods and fields: DomainParticipant, Publisher, Subscriber, Topics, DataWriters and DataReaders. Because methods and field methods all operate under a limited set of 4 topics (as depicted in Figure 6.56), it is generally agreed to have these 4 inherit their QoS policy values from the Service Instance QoS profile

[constr_3735] Existence of DdsCpServiceInstance.ddsServiceQosProfile

Imposition time: CP: IT_EcuExt

⌈For each DdsCpServiceInstance, the reference in the role ddsServiceQosProfile shall exist.

Note: This profile applies to the all 4 topics used for methods and fields defined into the DdsCpServiceInstance (e.g ddsMethodRequestTopic, ddsMethodReplyTopic), ddsFieldRequestTopic, ddsFieldReplyTopic).⌋

Because events have their own per-instance topic, resulting in their own DDS DataReader or DataWriter entities they are allowed to get their own QoS policy configuration, independent from the Service Instance one. If DdsServiceInstanceEventCp does not define its own DdsCpQosProfile, that one of the parent DdsCpServiceInstance is used.

[TPS_SYST_03095] DDS Event QoSProfile ⌈If present, the DdsCpQosProfile referenced in the role DdsServiceInstanceEventCp.ddsEventQosProfile shall be used to identify the QoS policy configuration that applies to the single DdsServiceInstanceEventCp. If not present, the DdsCpServiceInstance.ddsServiceQosProfile shall be used.⌋

Class	**DdsCpServiceInstance** (abstract)			
Note	Provided and Consumed Dds Service Instances that are available at the ApplicationEndpoint. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DdsCpConsumedServiceInstance, DdsCpProvidedServiceInstance			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ddsFieldReply Topic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of field setters. **Tags:** atp.Splitkey=ddsFieldReplyTopic atp.Status=candidate
ddsField RequestTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of field getters. **Tags:** atp.Splitkey=ddsFieldRequestTopic atp.Status=candidate
ddsMethod ReplyTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of method replies. **Tags:** atp.Splitkey=ddsMethodReplyTopic atp.Status=candidate
ddsMethod RequestTopic	DdsCpTopic	0..1	ref	Reference to the DdsTopic used as fragment for the topic name of method requests. **Tags:** atp.Splitkey=ddsMethodRequestTopic atp.Status=candidate
ddsServiceQos Profile	DdsCpQosProfile	0..1	ref	Reference to the QOS Profile used for the service. **Tags:** atp.Splitkey=ddsServiceQosProfile atp.Status=candidate
discoveryType	DdsDiscoveryType Enum	0..1	attr	Defines the discovery scheme for this DDS Service Instance.
resource IdentifierType	DdsResourceIdentifier TypeEnum	0..1	attr	Defines the in-band instance identification fields used to discriminate samples related to specific Service Instances sharing the same DDS Topics. **Tags:** atp.Status=candidate
serviceInterface Id	String	0..1	attr	Unique Identifier that identifies the ServiceInterface in DDS. This Identifier is encoded in the USER_DATA QoS of the DomainParticipant associated with the Service Instance and its value is propagated by DDS Discovery messages. **Tags:** atp.Status=candidate
serviceMinor Version	PositiveInteger	0..1	attr	Minor Version of the Service defined by this DdsCp ServiceInstance. **Tags:** atp.Status=candidate
transport Protocol	String	0..1	attr	This attribute defines which Transport Layer Protocol(s) this Instance is intended to use. **Tags:** atp.Status=candidate

Table 6.183: DdsCpServiceInstance

Enumeration	**DdsDiscoveryTypeEnum**
Note	Supported discovery schemes for DDS Service Instances. **Tags:** atp.Status=candidate
Aggregated by	*DdsCpServiceInstance*.discoveryType
Literal	***Description***
domainParticipant UserDataQos	The USER_DATA QoS policy is used to advertise and discover available Service Instances hosted by each Domain Participant [33]. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate
topic	A purpose-specific Topic is used to convey availability of Service Instances and how to bind against them [33]. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate

Table 6.184: DdsDiscoveryTypeEnum

Enumeration	**DdsResourceIdentifierTypeEnum**
Note	In-band instance identification fields are used to discriminate samples related to specific Service Instances sharing the same DDS Topics. **Tags:** atp.Status=candidate
Aggregated by	*DdsCpServiceInstance*.resourceIdentifierType
Literal	***Description***
instanceId	In-band instance identification fields are used to discriminate samples related to specific Service Instances sharing the same DDS Topics [33]. **Tags:** atp.EnumerationLiteralIndex=2 atp.Status=candidate
partition	The DDS PARTITION QoS policy is used to isolate DDS Topics related to specific Service Instances [33]. **Tags:** atp.EnumerationLiteralIndex=0 atp.Status=candidate
topicPrefix	Unique prefixes are assigned to DDS Topics related to specific Service Instances [33]. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=candidate

Table 6.185: DdsResourceIdentifierTypeEnum

Class	**DdsCpProvidedServiceInstance**			
Note	This meta-class represents the ability to describe the existence and configuration of a provided service instance in a concrete implementation on top of DDS. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *DdsCpServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DdsCpProvidedServiceInstance**			
localUnicast Address	ApplicationEndpoint	0..1	ref	The local address over which the Service is provided. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.namePlural=LOCAL-UNICAST-ADDRESSES
providedEvent	DdsServiceInstance EventCp	*	aggr	Collection of provided events. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedEvent, providedEvent.shortName, providedEvent.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
providedField	DdsServiceInstance FieldCp	*	aggr	Collection of provided fields. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedField, providedField.shortName, providedField.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
provided Operation	DdsServiceInstance OperationCp	*	aggr	Collection of provided operations. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=providedOperation, providedOperation.short Name, providedOperation.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
providedService InstanceId	PositiveInteger	0..1	attr	This attribute describes the provided service instance ID. **Tags:** atp.Status=candidate
staticRemote Multicast Address	ApplicationEndpoint	0..1	ref	This reference defines the remote multicast address of Service consumers. This reference shall ONLY be used if the remote multicast address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteMulticastAddress, staticRemote MulticastAddress.applicationEndpoint, staticRemote MulticastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-MULTICAST-ADDRESSES

▽

△

Class	**DdsCpProvidedServiceInstance**			
staticRemote UnicastAddress	ApplicationEndpoint	*	ref	This reference defines the remote unicast addresses of Service consumers. This reference shall ONLY be used if the remote unicast address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteUnicastAddress, staticRemote UnicastAddress.applicationEndpoint, staticRemote UnicastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-UNICAST-ADDRESSES

Table 6.186: DdsCpProvidedServiceInstance

Class	**DdsCpConsumedServiceInstance**			
Note	This meta-class represents the ability to describe the existence and configuration of a consumed (required) service instance in a concrete implementation on top of DDS. **Tags:** atp.Status=candidate			
Base	*ARObject*, *AbstractServiceInstance*, *DdsCpServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
consumedEvent	DdsServiceInstance EventCp	*	aggr	Collection of consumed events. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedEvent, consumedEvent.short Name, consumedEvent.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
consumedField	DdsServiceInstance FieldCp	*	aggr	Collection of consumed fields. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedField, consumedField.shortName, consumedField.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
consumed Operation	DdsServiceInstance OperationCp	*	aggr	Collection of consumed operations. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consumedOperation, consumed Operation.shortName, consumedOperation.variation Point.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild

▽

△

Class	**DdsCpConsumedServiceInstance**			
localUnicast Address	ApplicationEndpoint	0..1	ref	The local address over which the Service is consumed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=localUnicastAddress, localUnicast Address.applicationEndpoint, localUnicast Address.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.namePlural=LOCAL-UNICAST-ADDRESSES
requiredService InstanceId	PositiveInteger	0..1	attr	This attribute describes the required service instance ID. **Tags:** atp.Status=candidate
staticRemote Multicast Address	ApplicationEndpoint	0..1	ref	This reference defines the remote multicast address of the Service provider. This reference shall ONLY be used if the remote multicast address of the server is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteMulticastAddress, staticRemote MulticastAddress.applicationEndpoint, staticRemote MulticastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-MULTICAST-ADDRESSES
staticRemote UnicastAddress	ApplicationEndpoint	0..1	ref	This reference defines the remote unicast address of the Service provider. This reference shall ONLY be used if the remote unicast address of the server is determined from the configuration and not at runtime. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticRemoteUnicastAddress, staticRemote UnicastAddress.applicationEndpoint, staticRemote UnicastAddress.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime xml.name Plural=STATIC-REMOTE-UNICAST-ADDRESSES

Table 6.187: DdsCpConsumedServiceInstance

Class	***DdsAbstractServiceInstanceElementCp*** (abstract)			
Note	This element represents an abstraction for DDS Events, Methods, and Fields. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DdsServiceInstanceEventCp, DdsServiceInstanceFieldCp, DdsServiceInstanceOperationCp			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.188: DdsAbstractServiceInstanceElementCp

Class	DdsServiceInstanceEventCp			
Note	This element represents an event as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedEvent, DdsCpProvidedServiceInstance.providedEvent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ddsEventQos Profile	DdsCpQosProfile	0..1	ref	Reference to the QOS Profile used for this Event. **Tags:** atp.Splitkey=ddsEventQosProfile atp.Status=candidate
ddsEventTopic	DdsCpTopic	0..1	ref	Reference to the DDS Topic used for this Event. **Tags:** atp.Splitkey=ddsEventTopic atp.Status=candidate
eventTriggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsEvent message. **Tags:** atp.Splitkey=eventTriggering atp.Status=candidate

Table 6.189: DdsServiceInstanceEventCp

Class	DdsServiceInstanceOperationCp			
Note	This element represents an operation as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedOperation, DdsCpProvidedServiceInstance.provided Operation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation Request Triggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsOperation request message. **Tags:** atp.Splitkey=operationRequestTriggering atp.Status=candidate
operation Response Triggering	PduTriggering	0..1	ref	Reference to the PduTriggering used for the upper layer transport of this DdsOperation response message. **Tags:** atp.Splitkey=operationResponseTriggering atp.Status=candidate

Table 6.190: DdsServiceInstanceOperationCp

Class	DdsServiceInstanceFieldCp			
Note	This element represents a field as part of the DDS Service Instance. **Tags:** atp.Status=candidate			
Base	*ARObject*, *DdsAbstractServiceInstanceElementCp*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DdsCpConsumedServiceInstance.consumedField, DdsCpProvidedServiceInstance.providedField			
Attribute	**Type**	**Mult.**	**Kind**	**Note**



△

Class	DdsServiceInstanceFieldCp			
getter	DdsServiceInstance OperationCp	0..1	ref	Reference to the field getter operation. **Tags:** atp.Splitkey=getter atp.Status=candidate
notifier	DdsServiceInstance EventCp	0..1	ref	Reference to the field notifier event. **Tags:** atp.Splitkey=notifier atp.Status=candidate
setter	DdsServiceInstance OperationCp	0..1	ref	Reference to the field setter operation. **Tags:** atp.Splitkey=setter atp.Status=candidate

Table 6.191: DdsServiceInstanceFieldCp

[constr_3800] Existence of `DdsCpServiceInstance.discoveryType`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `discoveryType` shall exist.⌋

[constr_3801] Existence of `DdsCpServiceInstance.serviceInterfaceId`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `serviceInterfaceId` shall exist.⌋

[constr_3802] Existence of `DdsCpServiceInstance.majorVersion`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `AbstractServiceInstance.majorVersion` shall exist.⌋

[constr_3803] Existence of `DdsCpServiceInstance.serviceMinorVersion`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `serviceMinorVersion` shall exist.⌋

[constr_3804] Existence of `DdsCpServiceInstance.transportProtocol`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsCpServiceInstance`, the attribute in the role `transportProtocol` shall exist.⌋

[constr_3805] Existence of `DdsServiceInstanceEventCp.eventTriggering`

Imposition time: CP: `IT_EcuExt`

⌈For each `DdsServiceInstanceEventCp`, the reference in the role `eventTriggering` shall exist.⌋

[constr_3806] Existence of DdsServiceInstanceOperationCp.operationRequestTriggering

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceOperationCp, the reference in the role operationRequestTriggering shall exist.⌋

[constr_3807] Existence of DdsServiceInstanceFieldCp reference

Imposition time: CP: IT_EcuExt

⌈For each DdsServiceInstanceFieldCp, at least one of the references in the role

- notifier
- getter
- setter

shall exist.⌋

[constr_3808] Existence of DdsCpProvidedServiceInstance.providedServiceInstanceId

Imposition time: CP: IT_EcuExt

⌈For each DdsCpServiceInstance, the attribute in the role providedServiceInstanceId shall exist.⌋

[constr_3809] Existence of DdsCpConsumedServiceInstance.requiredServiceInstanceId

Imposition time: CP: IT_EcuExt

⌈For each DdsCpConsumedServiceInstance, the attribute in the role requiredServiceInstanceId shall exist.⌋

6.7.4.6.16.1 DDS Provided Service Instances

The DdsCpProvidedServiceInstance is used to define a DDS Service provider. This meta-class represents the ability to describe the existence and configuration of a provided service instance in a concrete implementation on top of DDS.

[constr_3810] Value range of DdsCpProvidedServiceInstance.providedServiceInstanceId

Imposition time: CP: IT_SysDesc

⌈The value of DdsCpProvidedServiceInstance.providedServiceInstanceId shall be in the range of 0..65534.⌋

[TPS_SYST_03134] DdsCpProvidedServiceInstance deployment ⌈

The deployment of a DdsCpProvidedServiceInstance to an EcuInstance is achieved via reference localUnicastAddress.

The ApplicationEndpoint referenced by each of them defines the corresponding UDP port, while the NetworkEndpoint referenced by the ApplicationEndpoint specifies the IP address at which the Service is accessible.

With the SocketAddress.connector reference the deployment to an EcuInstance is achieved.⌋

Please note that the Service provider and the Service consumer do not need to find each other if the DdsCpProvidedServiceInstance contains staticRemoteUnicastAddress reference to an ApplicationEndpoint. Furthermore, if this reference is set, the Server will not get the client's address using the DDS Participant Discovery protocol.

The server knows the address of the Client from the configuration and all necessary socket connections can be set up from this configuration information.

[TPS_SYST_03135] Static unicast configuration between DdsCpProvidedServiceInstance and DdsCpConsumedServiceInstance ⌈If the DdsCpProvidedServiceInstance contains a staticRemoteUnicastAddress reference to an ApplicationEndpoint the SoAd will setup one single SocketConnection between the localUnicastAddress and the staticRemoteUnicastAddress.⌋

[TPS_SYST_03136] Maximal number of unicast clients that may connect to the local unicast server address ⌈If the DdsCpProvidedServiceInstance references an ApplicationEndpoint in the role localUnicastAddress then the attribute ApplicationEndpoint.maxNumberOfConnections defines the maximal number of clients (with a dynamic address) that will be able to connect to this local server address. The SoAd will setup a SocketConnection for each potential client with localUnicastAddress and ANY remote Address (ANY Port and ANY IP-Address).⌋

[constr_3811] DdsCpProvidedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpProvidedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_3812] DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_3813] `DdsCpProvidedServiceInstance` unicast address references target

Imposition time: CP: `IT_SysDesc`

⌈Each of the following references:

- `localUnicastAddress`
- `staticRemoteUnicastAddress`

If specified, shall refer to an `ApplicationEndpoint` defining a UDP Port⌋

[TPS_SYST_03137] `DdsCpProvidedServiceInstance` Unicast Address UDP port configuration rules ⌈The UDP port defined in accordance with [constr_3813] shall be defined in accordance with following rules:

- portNumber = 7400 + 11 + 250*`DdsCpDomain.domainId` + 2*`domainParticipantId`
- `DdsCpDomain.domainId` belongs to the interval `[0, 230)`
- `domainParticipantId` belong to the interval `[0, 120)`

⌋

For further details about [TPS_SYST_03137], refer to [31].

[constr_3814] Multiple `SocketAddress` entries with the same IP Address and Port in the context of a given `EcuInstance`

Imposition time: CP: `IT_SysDesc`

⌈If there are two or more `SocketAddress` entities within the scope of one `SoAdConfig` in the scope of one `EcuInstance` that have the same static (fixed at configuration time) IP Address and Port in the aggregated `ApplicationEndpoint` and `NetworkEndpoint`, (e.g., `192.168.1.1` and `7431`, respectively) then:

- only one of these `SocketAddress` elements shall be referenced by `DdsCpProvidedServiceInstance`s or `DdsCpConsumedServiceInstance`s in the role `localUnicastAddress`.
- only one of these `SocketAddress` elements shall be referenced by `DdsCpProvidedServiceInstance`s or `DdsCpConsumedServiceInstance`s in the role `staticRemoteUnicastAddress`.

⌋

[constr_3815] DDS Service communication is restricted to one VLAN

Imposition time: CP: `IT_SysDesc`

⌈`ApplicationEndpoint`s elements that are referenced by:

- `localUnicastAddress` and `staticRemoteUnicastAddress` shall belong to the same VLAN.

⌋

The VLAN (e.g., EthernetPhysicalChannel) membership of a given ApplicationEndpoint is determined by the SocketAddress object that aggregates it through a specific reference to an identified EthernetPhysicalChannel.

[TPS_SYST_03138] PduTriggering that is used for DDS ClientServer Communication ⌈If a given PduTriggering is defined in accordance with [TPS_SYST_02081], it shall be employed in the context of DDS Client-Server Communication.⌋

6.7.4.6.16.2 DDS Consumed Service Instances

The DdsCpConsumedServiceInstance is used to define a DDS Service provider. This meta-class represents the ability to describe the existence and configuration of a required service instance in a concrete implementation on top of DDS.

[constr_3816] Value range of DdsCpConsumedServiceInstance.requiredServiceInstanceId

Imposition time: CP: IT_SysDesc

⌈The value of DdsCpConsumedServiceInstance.requiredServiceInstanceId shall be in the range of 0..65534 or "ALL".⌋

[TPS_SYST_03139] DdsCpConsumedServiceInstance deployment ⌈The deployment of a DdsCpConsumedServiceInstance to an EcuInstance is achieved via reference localUnicastAddress.

The ApplicationEndpoint referenced by each of them defines the corresponding UDP port, while the NetworkEndpoint referenced by the ApplicationEndpoint specifies the IP address at which the Service consumer is located.

This Endpoint information is transported via the DDS Participant Discovery protocol.

With the SocketAddress.connector reference the deployment to an EcuInstance is achieved.⌋

Please note that the Service provider and the Service consumer do not need to find each other if the DdsCpConsumedServiceInstance contains a staticRemoteUnicastAddressreference to an ApplicationEndpoint. Furthermore, if this reference is set, the client will not get the server's address from the DDS Participant Discovery protocol engine.

The client already knows the address of the service provider from the configuration and all necessary socket connections can be set up from this configuration information.

[TPS_SYST_03140] Static unicast configuration between DdsCpConsumedServiceInstance and DdsCpProvidedServiceInstance ⌈If the DdsCpConsumedServiceInstance contains a staticRemoteUnicastAddress reference to an ApplicationEndpoint the SoAd will setup one single SocketConnection between the localUnicastAddress and the staticRemoteUnicastAddress.⌋

[TPS_SYST_03141] Maximal number of unicast servers that may connect to the local unicast client address ⌈If the DdsCpConsumedServiceInstance references an ApplicationEndpoint in the role localUnicastAddress then the attribute ApplicationEndpoint.maxNumberOfConnections defines the maximal number of remote endpoints that will be able to connect to this local server address. The SoAd will setup a SocketConnection for each potential client with localUnicastAddress. The local Address of the SocketConnection is derived from localUnicastAddress.⌋

[constr_3817] DdsCpConsumedServiceInstance.localUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpConsumedServiceInstance.localUnicastAddress shall point to an IP Unicast address.⌋

[constr_3818] DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast

Imposition time: CP: IT_SysDesc

⌈If defined, the DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall point to an IP Unicast address.⌋

[constr_3819] DdsCpConsumedServiceInstance unicast address references target

Imposition time: CP: IT_SysDesc

⌈

- localUnicastAddress
- staticRemoteUnicastAddress

If specified, shall refer to an ApplicationEndpoint defining a UDP Port.⌋

[TPS_SYST_03142] DdsCpConsumedServiceInstance Unicast Address UDP port configuration rules ⌈The UDP port defined in accordance with [constr_3819] shall be defined in accordance with following rules:

- portNumber = 7400 + 11 + 250*DdsCpDomain.domainId + 2*domainParticipantId
- DdsCpDomain.domainId belongs to the interval [0, 230)
- domainParticipantId belong to the interval [0, 120)

⌋

For further details about [TPS_SYST_03142], refer to [31].

[constr_3820] DDS Service communication is restricted to one VLAN

Imposition time: CP: IT_SysDesc

⌈ApplicationEndpoints elements that are referenced by:

- localUnicastAddress and staticRemoteUnicastAddress shall belong to the same VLAN.

⌋

6.7.4.7 StaticSocketConnection

Currently only the SOME/IP data communication that is subject to Service Discovery is described with ProvidedServiceInstances and ConsumedServiceInstances as mentioned in chapter 6.7.4.3.1.

But there is also the need to describe data exchange between two SocketAddresses where the SOME/IP Service Discovery protocol is not used. This may include for example the exchange of NM Data (see 6.7.4.9), DoIP data (see 6.7.4.10), SOME/IP Service Discovery messages (see 6.7.4.8), DDS data (see 6.7.4.6) and DDS Services (see 6.7.4.6.16).

Note: Differently from SOME/IP, for Service oriented communication on DDS the definition of DdsCpProvidedServiceInstances and DdsCpConsumedServiceInstances is needed also for complete static configuration, so also in this case StaticSocketConnections shall be used.

But also the data exchange that is not serialized according to the SOME/IP rules may be exchanged via the so called StaticSocketConnections. And even the exchange of SOME/IP serialized data which transmission is not subject to Service Discovery may be described with StaticSocketConnections. In this case the ServiceId and MessageId can be derived from the headerId of the SoConIPduIdentifier. If such SOME/IP serialized data exchange is described via a StaticSocketConnection then the definition of ProvidedServiceInstances and ConsumedServiceInstances does not need to be defined for this data exchange since Service Discovery is not used and all necessary information for the configuration of the COM Stack is available in the StaticSocketConnection itself and in the referenced SoConIPduIdentifiers.

[TPS_SYST_02251] Data exchange not regulated by the Service Discovery protocol between two communication endpoints ⌈Data exchange that is NOT subject to Service Discovery between two SocketAddresses is described with the StaticSocketConnection element that is aggregated by a SocketAddress in the role staticSocketConnection. The aggregating SocketAddress describes one communication endpoint. The referenced StaticSocketConnection.remoteAddress defines the second communication endpoint.⌋

Class	**StaticSocketConnection**			
Note	Definition of static SocketConnection between the Socket that is defined by the aggregating Socket Address and the remoteAddress.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SocketAddress.staticSocketConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
iPduIdentifier	SoConIPduIdentifier	*	ref	Assignment of IPduIdentifiers that are transmitted over the static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduIdentifier, iPduIdentifier.soConIPdu Identifier, iPduIdentifier.variationPoint.shortLabel vh.latestBindingTime=postBuild
remoteAddress	SocketAddress	0..1	ref	RemoteAddress of the static SocketConnection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=remoteAddress, remoteAddress.socket Address, remoteAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
tcpConnect Timeout	TimeValue	0..1	attr	Specifies the time in seconds how long TCP connect attempts are repeated to reach SOAD_SOCON_ONLINE. This attribute is restricted to socket connection groups which are initiating a TCP connection and are under control of SoAd.
tcpRole	TcpRoleEnum	0..1	attr	Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.

Table 6.192: StaticSocketConnection

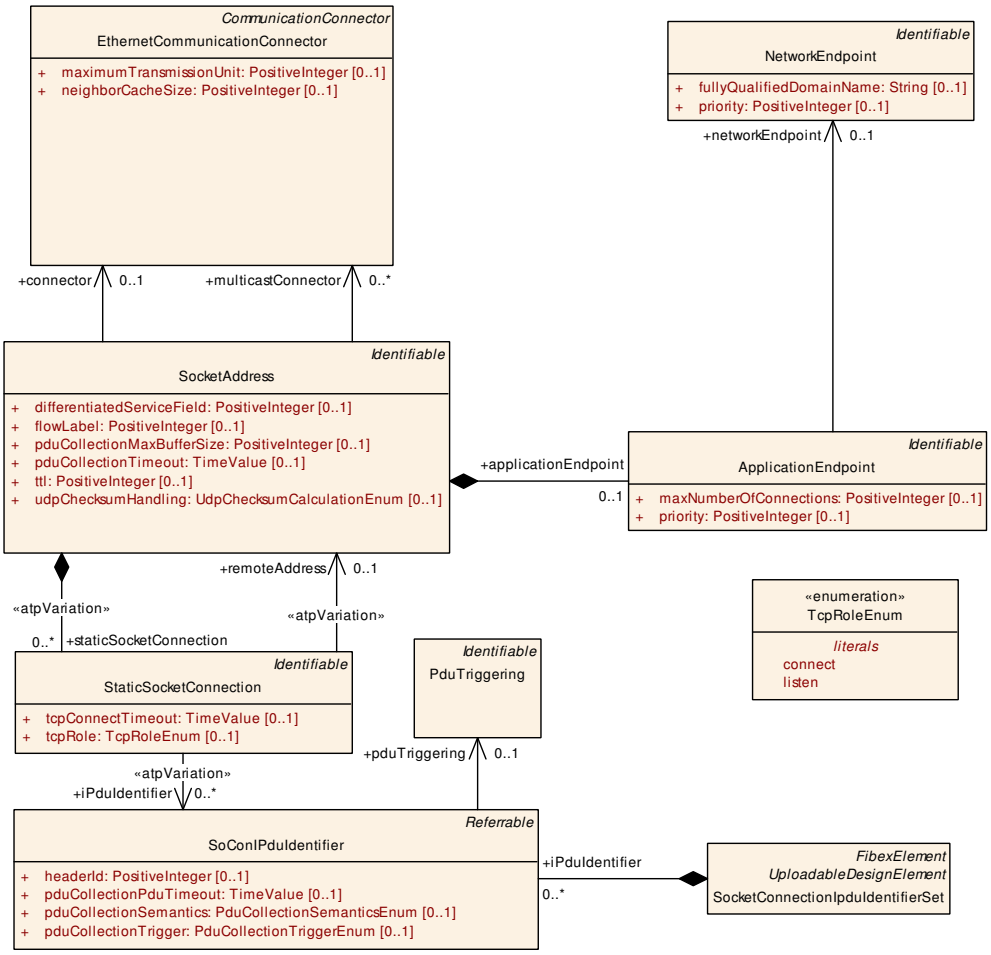

Figure 6.57: Model of StaticSocketConnection

UDP doesn't establish a connection before sending data, it is Connectionless. TCP on the other hand establishes a TCP connection. The client initiates the TCP communication and therefore it needs to be configured which endpoint acts as client and which acts as server.

[TPS_SYST_02252] Description of a TCP Client ⌈The `SocketAddress` that aggregates the `StaticSocketConnection` with `tcpRole` set to `connect` defines the TCP client.⌋

[TPS_SYST_02253] Description of a TCP Server ⌈The `SocketAddress` that aggregates the `StaticSocketConnection` with `tcpRole` set to `listen` defines the TCP server.⌋

[constr_5091] Relevance of `tcpRole` attribute

Imposition time: CP: `IT_SysDesc`

⌈The attribute `tcpRole` shall only exist if the `StaticSocketConnection` is aggregated by a `SocketAddress` that defines a TCP Port in the aggregated `ApplicationEndpoint`.⌋

[constr_5092] Local and remoteAddress of a StaticSocketConnection shall define the same transport protocol

Imposition time: CP: IT_SysDesc

⌈The transport protocol that is defined by the SocketAddress that aggregates the StaticSocketConnection shall be the same in the SocketAddress that is referenced by the same StaticSocketConnection in the role remoteAddress.⌋

[TPS_SYST_02254] Pdus transported over the StaticSocketConnection ⌈A Pdu that is transported over the StaticSocketConnection is described by SoConIPduIdentifier that is referenced in the role iPduIdentifier.⌋

Please note that the remoteAddress is allowed to be defined as ANY (TpPort = 0 and ipv4address/ipv6address = ANY). In other words the remote port and/or remote IP address is configured at runtime and the ECU is able to receive all packets that are addressed to its local address no matter what the remote address is.

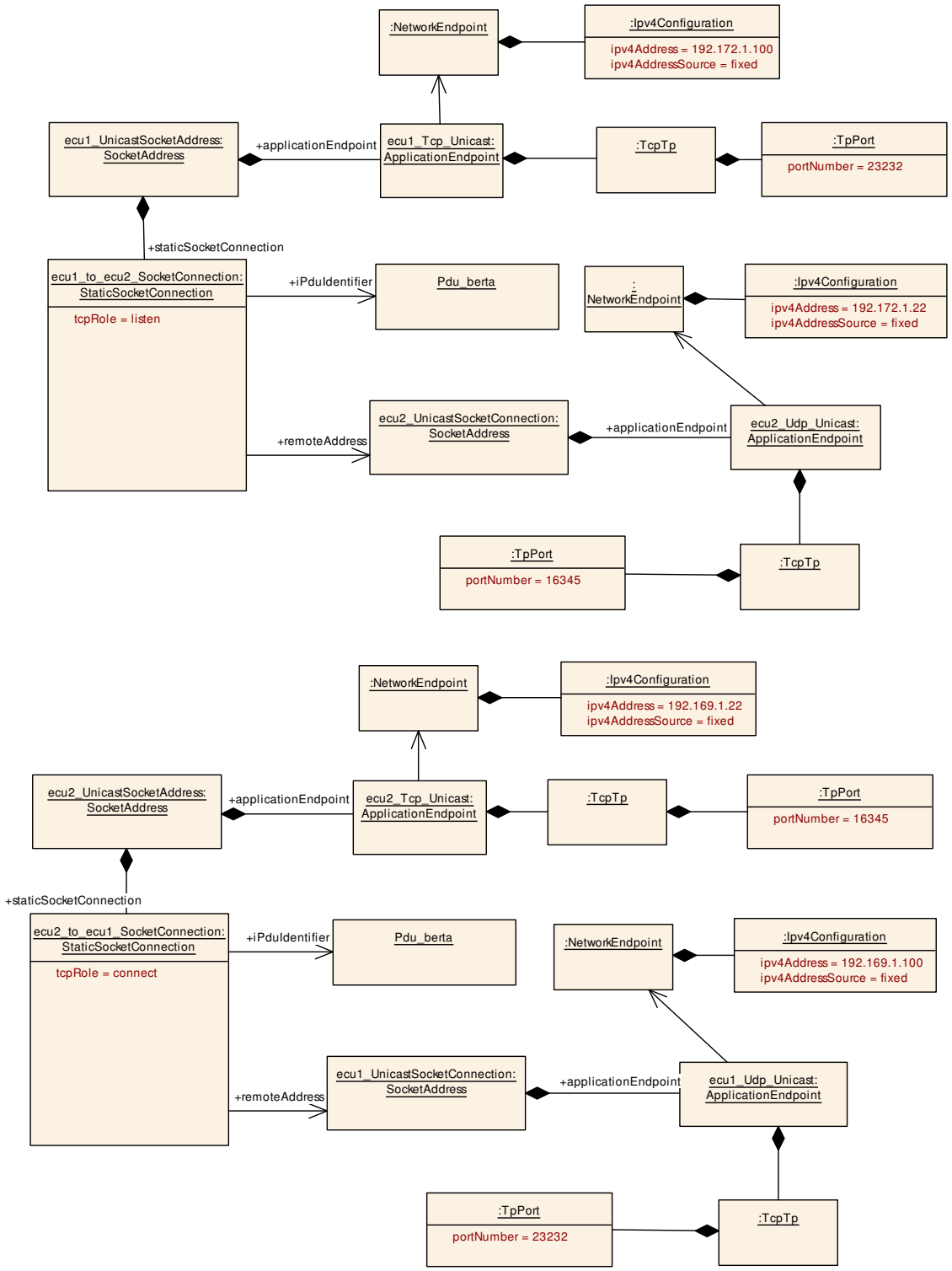

Figure 6.58: Example for the modeling of TCP StaticSocketConnections in TCP server role and TCP client role

The example in Figure 6.58 shows a `StaticSocketConnection` with the name *ecu2_to_ecu1_SocketConnection* on *ecu1_UnicastSocketAddress* that defines the TCP server. The socket connection is established to to *ecu2_UnicastSocketAddress* `remoteAddress`.

It also shows the StaticSocketConnection with the name *ecu1_to_ecu2_SocketConnection* on *ecu2_UnicastSocketAddress* that defines the TCP client. The socket connection is established to the *ecu1_UnicastSocketAddress* remoteAddress.

6.7.4.8 Service Discovery Message Configuration

If Service Discovery is used a Service Discovery Instance is configurable on an EcuInstance for a certain VLAN using the respective ApplicationEndpoint. The Service Discovery Instance refers to the configuration of a Service Discovery for a VLAN.

[TPS_SYST_02116] Modeling of Service Discovery Pdus ⌈A Service Discovery Instance configuration requires all of the following:

- one Tx GeneralPurposePdu (unicast and multicast transmission) of category = SD (see [TPS_SYST_02411])
- one Rx GeneralPurposePdu (unicast reception) of category = SD (see [TPS_SYST_02412])
- one Rx GeneralPurposePdu (multicast reception) of category = SD (see [TPS_SYST_02413])

⌋

[TPS_SYST_02117] Length of GeneralPurposePdu with category SD ⌈The length attribute for GeneralPurposePdus with category = SD shall be set to at most EthernetCommunicationConnector.maximumTransmissionUnit minus the sum of the length of the following headers:

- IP Header (for IpV4: 20 bytes, for IpV6: 40 bytes)
- IpV4 and IpV6 : any optional additional headers, e.g. for IPSec
- Udp Header (8 bytes)
- Socket Adaptor PDU Header (8 bytes)

⌋

[TPS_SYST_02411] Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are transmitted by an EcuInstance ⌈For each GeneralPurposePdu with category SD that is transmitted by an EcuInstance over IP unicast or IP multicast a PduTriggering needs to be defined on the EthernetPhysicalChannel (VLAN) that is referenced from the CommunicationConnector of the EcuInstance on which the Service Discovery Instance is configured:

- the PduTriggering for the Tx GeneralPurposePdu references the IPduPort with communicationDirection out of the EcuInstance

⌋

[TPS_SYST_02412] Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are received via IP unicast by an EcuInstance ⌈For each GeneralPurposePdu with category SD that is received by an EcuInstance over IP unicast, a PduTriggering needs to be defined on the EthernetPhysicalChannel (VLAN) that is referenced from the CommunicationConnector of the EcuInstance on which the Service Discovery Instance is configured:

- the PduTriggering for the Rx GeneralPurposePdu references the IPduPort with communicationDirection in of the EcuInstance

⌋

[TPS_SYST_02413] Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are received via IP mutlicast by an EcuInstance ⌈For each GeneralPurposePdu with category SD that is received by an EcuInstance over IP multicast, a PduTriggering needs to be defined on the EthernetPhysicalChannel (VLAN) that is referenced from the CommunicationConnector of the EcuInstance on which the Service Discovery Instance is configured:

- the PduTriggering for the Rx GeneralPurposePdu references the IPduPort with communicationDirection in of the EcuInstance

⌋

[TPS_SYST_02414] StaticSocketConnection for IP unicast/IP multicast transmission and IP unicast reception of GeneralPurposePdus with category SD ⌈A UDP Socket is used for transmission and reception of SD messages using a local unicast address. One StaticSocketConnection shall be created for each EcuInstance that participates in the SD message exchange to define:

- IP unicast transmission of SD messages (SubscribeEventGroup, StopSubscribeEventGroup, SubscribeEventGroupAck) by the EcuInstance
- IP unicast reception of SD messages (SubscribeEventGroup, StopSubscribeEventGroup, SubscribeEventGroupAck) by the EcuInstance
- IP multicast transmission of SD messages (OfferService, StopOfferService, FindService) by the EcuInstance

The PduTriggering that is defined in [TPS_SYST_02411] for transmitted GeneralPurposePdu and the PduTriggering for received GeneralPurposePdu that is defined in [TPS_SYST_02412] shall be assigned to the StaticSocketConnection via SoConIPduIdentifier. The StaticSocketConnection shall be aggregated on the local unicast SocketAddress of the EcuInstance.⌋

[TPS_SYST_02415] `StaticSocketConnection` for IP multicast reception of `GeneralPurposePdu`s with category SD ⌈A UDP Socket is used for reception of SD messages using a local multicast address. One `StaticSocketConnection` shall be created for each `EcuInstance` that participates in the SD message exchange to define:

- IP multicast reception of SD messages (OfferService, StopOfferService, FindService) by the `EcuInstance`

The `PduTriggering` for the received `GeneralPurposePdu`s that is defined in [TPS_SYST_02413] shall be assigned to the `StaticSocketConnection` via `SoConIPduIdentifier`. The `StaticSocketConnection` shall be aggregated on the multicast `SocketAddress` over which the `EcuInstance` receives the SD messages.⌋

[constr_3267] `PduTriggering`s in Service Discovery `StaticSocketConnection`s

Imposition time: CP: `IT_SysDesc`

⌈SD `StaticSocketConnection`s defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall only refer to `PduTriggering`s which point to `GeneralPurposePdu`s of `category` SD.⌋

[constr_3268] Service Discovery `StaticSocketConnection` aggregation by a `SocketAddress`

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall be aggregated by a `SocketAddress` that in turn aggregates an `ApplicationEndpoint` that defines a Udp Port.⌋

[constr_3269] Service Discovery `StaticSocketConnection` `remoteAddress` reference to a `TpPort`

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` with Udp Port `portNumber` set to 0. This means that any remote port number is accepted for receiving and for sending, i.e., that the remote port number is configured at runtime.⌋

[constr_3270] Service Discovery `StaticSocketConnection.remoteAddress` reference to an IP Address

Imposition time: CP: `IT_SysDesc`

⌈Each SD `StaticSocketConnection` defined according to [TPS_SYST_02414] and [TPS_SYST_02415] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Address ANY (IPv4 or IPv6). This means that any remote IP address is accepted for receiving and for sending, i.e., that the remote IP address is configured at runtime.⌋

[constr_3272] SoConIPduIdentifier.headerId setting for SD StaticSocketConnections

Imposition time: CP: IT_SysDesc

⌈The SoConIPduIdentifier.headerId of SD StaticSocketConnections defined in [TPS_SYST_02414] and [TPS_SYST_02415] shall always be set to 0xFFFF8100 for SD messages.⌋

[constr_3273] Service Discovery multicast StaticSocketConnection's aggregation by an ApplicationEndpoint

Imposition time: CP: IT_SysDesc

⌈The SD StaticSocketConnection for multicast defined in [TPS_SYST_02415] shall be aggregated by an ApplicationEndpoint that points to a NetworkEndpoint that defines an IP Multicast Address.⌋

[constr_3274] Service Discovery unicast StaticSocketConnection's aggregation by an ApplicationEndpoint

Imposition time: CP: IT_SysDesc

⌈The SD StaticSocketConnection for unicast defined in [TPS_SYST_02414] shall be aggregated by an ApplicationEndpoint that points to a NetworkEndpoint that defines an IP Unicast Address.⌋

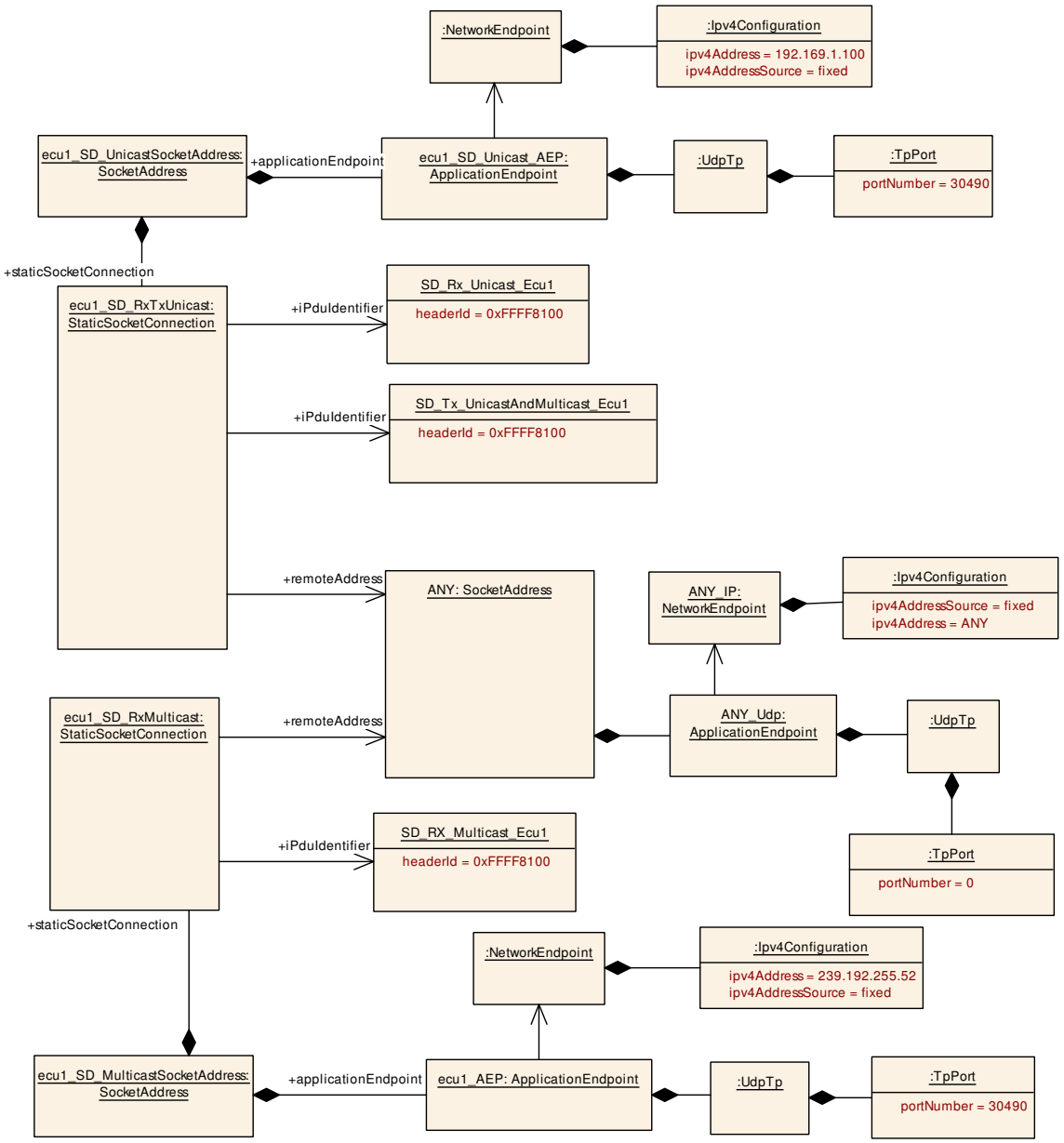

Figure 6.59: Example for the modeling of SD StaticSocketConnections

6.7.4.9 NmPdu data exchange over StaticSocketConnection

The data exchange of `NmPdu` on an Ethernet channel is described with the `StaticSocketConnection` as defined in chapter 6.7.4.7.

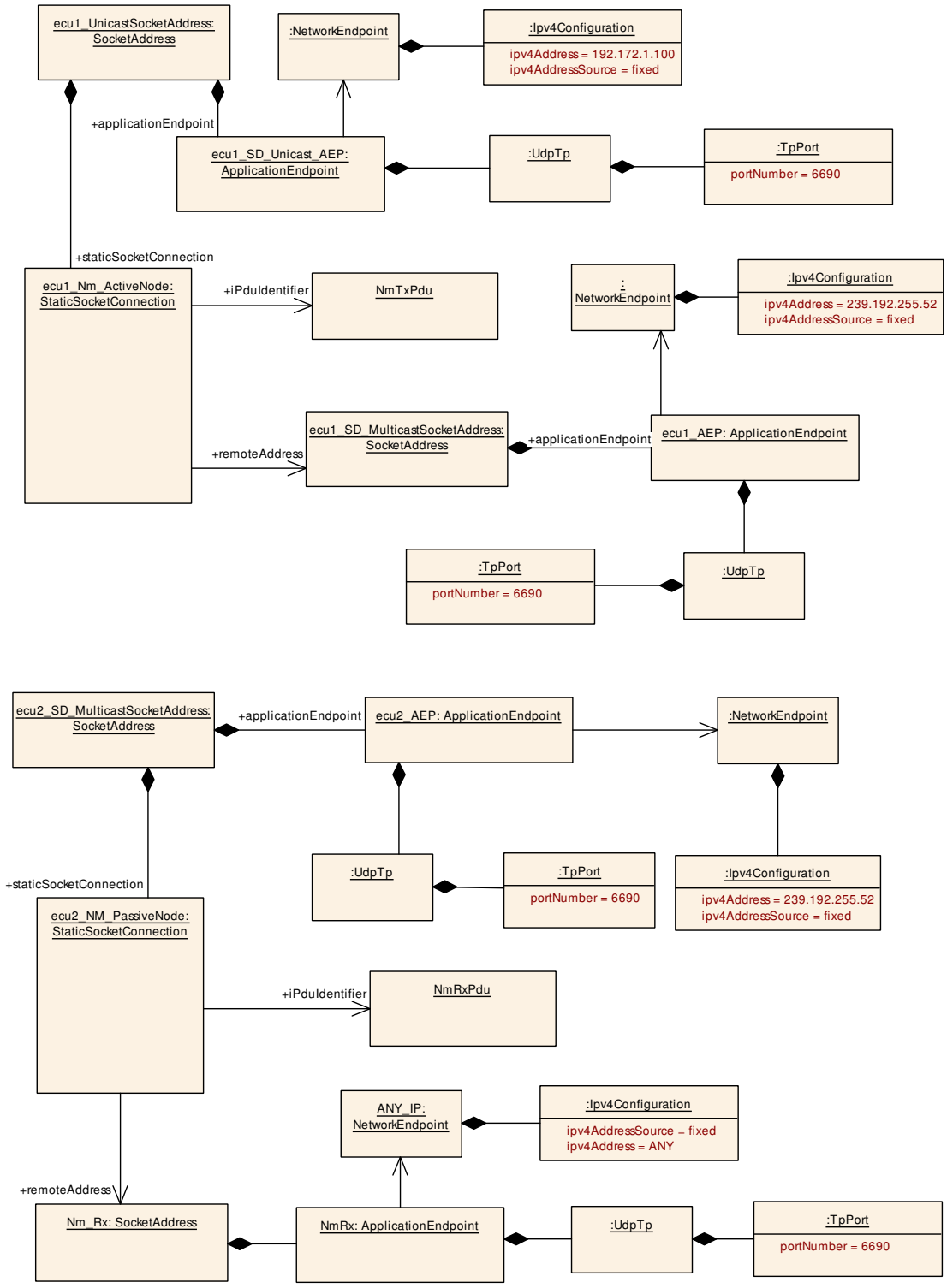

Figure 6.60: Example for the modeling of NM StaticSocketConnections

The example in Figure 6.60 shows a possible configuration where the active `NmNode` sends the `NmPdu` to a remote multicast address. Each passive and active `NmNode` receives the `NmPdu` over a local multicast address from all other nodes (remote Address = ANY).

6.7.4.10 Diagnostics over IP

DoIpConfig defines a DoIP module configuration for a specific EcuInstance. DoIP supports the communication of internal testers and external testers with an ECU/DoIP Node. Each DoIP node might define several logical DoIpInterfaces that may share the same physical Ethernet interface/MAC Address. The DoIP node is able to communicate on each of its DoIpInterfaces independently. I.e. the DoIP functionalities on each DoIpInterface are isolated from each other.

More details about DoIpInterfaces can be found in AUTOSAR_SWS_DiagnosticOverIP [34].

Class	**DoIpConfig**			
Note	This element defines the DoIp configuration for a specific Ecu.			
Base	*ARObject*			
Aggregated by	EcuInstance.doIpConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
doipInterface	DoIpInterface	*	aggr	DoIP node consists of one or several DoIPInterfaces over which the ECU is able to communicate via DoIP independently. I.e. DoIP functionalities on each IP interface are isolated from each other. **Tags:** atp.Splitkey=doipInterface, doipInterface.shortName
doIpProtocolVersion	PositiveInteger	0..1	attr	Configures the DoIP protocol version used in the generic DoIP header. The valid range of this parameter is defined by the always latest release of ISO 13400-2 and can be extended with every new release of the ISO document. As example a value of 0x03 defines the ISO 13400-2:2019 release.
logicAddress	DoIpLogicAddress	0..1	aggr	Describes the logical address of the DoIP entity, i.e. the Local Address that will route diagnostic requests to the Dcm of the DoIP entity. **Tags:** atp.Splitkey=logicAddress, logicAddress.shortName

Table 6.193: DoIpConfig

Class	**DoIpInterface**			
Note	A logical interface over which the DoIP Node is able to communicate via DoIP independently from other existing DoIpInterfaces.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpConfig.doipInterface			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aliveCheckResponseTimeout	TimeValue	0..1	attr	This attribute defines the timeout in seconds for waiting for response to an Alive Check request before the connection is considered to be disconnected. Represents parameter T_TCP_AliveCheck of ISO 13400-2:2012.

▽

△

Class	DoIpInterface			
doipChannel Collection	DoIpTpConfig	0..1	ref	Configuration of DoIPChannels available in an DoIp Interface. Each DoIPChannel describes a connection between a doIpSourceAddress and a doIpTargetAddress and the exchange of DcmIPdus between the PduR and DoIP. A DoIP channel is constituted by the set of all DoIp TpConnection elements via which the configured Ecu Instance sends or receives SDUs that are sharing the same local diagnosis address and tester address. **Tags:** atp.Splitkey=doipChannelCollection
doIpRouting Activation	DoIpRoutingActivation	*	aggr	Collection of DoIpRoutingActivation possibilities defined in the DoIpInterface. **Tags:** atp.Splitkey=doIpRoutingActivation, doIpRouting Activation.shortName
generalInactivity Time	TimeValue	0..1	attr	This attribute defines the timeout in seconds for maximum inactivity of a TCP socket connection before the DoIP module will close the according socket connection. Represents parameter T_TCP_General_Inactivity of ISO 13400-2:2012
initialInactivity Time	TimeValue	0..1	attr	This attribute defines the timeout in seconds used for initial inactivity of a connected TCP socket connection directly after socket connection. Represents parameter T_TCP_Initial_Inactivity of ISO 13400-2:2012
initialVehicle Announcement Time	TimeValue	0..1	attr	This attribute defines the waiting time in seconds for sending first vehicle announcement message after IP address assignment. Represents parameter A_DoIP_ Announce_Wait of ISO 13400-2:2012
isActivationLine Dependent	Boolean	0..1	attr	This attribute defines whether the network interface • is started "on-demand" when an activation line is sensed or • is always available.
maxTester Connections	PositiveInteger	0..1	attr	Maximum amount of tester connections that shall be maintained at one time before alive check is performed.
socket Connection	StaticSocketConnection	*	ref	DoIP Connections in the DoIpInterface that define the Do IP Pdus that are sent and received via SoAd over TCP or UDP. **Tags:** atp.Splitkey=socketConnection
useMacAddress ForIdentification	Boolean	0..1	attr	This attribute defines whether a configured EID at vehicle identification response/vehicle announcement is used or the MAC address. TRUE: Use MAC Address instead of EID for Vehicle identification/announcement. FALSE: Use configured EID for vehicle identification/announcement.
useVehicle Identification SyncStatus	Boolean	0..1	attr	This attribute defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.
vehicle Announcement Count	PositiveInteger	0..1	attr	This attribute defines the number of vehicle announcement messages on IP address assignment. Represents parameter A_DoIP_Announce_Num of ISO 13400-2:2012.
vehicle Announcement Interval	TimeValue	0..1	attr	This attribute defines the waiting time in seconds for sending subsequent vehicle announcement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012

Table 6.194: DoIpInterface

Class	**DoIpRoutingActivation**			
Note	This meta-class defines a DoIP routing activation possibility that activates the routing to the referenced doIPTargetAddress. This means that the diagnostic request messages related to the specified do IPTargetAddress received by socketConnections that are referenced by the same DoIpInterface that aggregates this DoIpRoutingActivation are activated.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpInterface.doIpRoutingActivation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
doIpTarget Address	DoIpLogicTarget AddressProps	*	ref	Reference to DoIPTargetAddress which is activated on this DoIpRoutingActivation. **Tags:** atp.Splitkey=doIpTargetAddress

Table 6.195: DoIpRoutingActivation

The DoIpInterface defines DoIpChannels with the doipChannelCollection reference that points to a DoIpTpConfig element that collects a number of DoIpTpConnections.

DoIpTpConnection describes a unidirectional connection between a doIpSourceAddress and a doIpTargetAddress and the exchange of a DcmIPdu that is defined with the tpSdu between the PduR and DoIP. The DiagnosticConnection with references to the DoIpTpConnection defines the related request and response messages.

A DoIpChannel in the Ecu configuration is constituted by the set of all DoIpTpConnection elements via which the EcuInstance that aggregates the DoIpConfig that references the DoIpTpConnection via the DoIpInterface sends or receives tpSdus that share the same local diagnosis address and tester address.

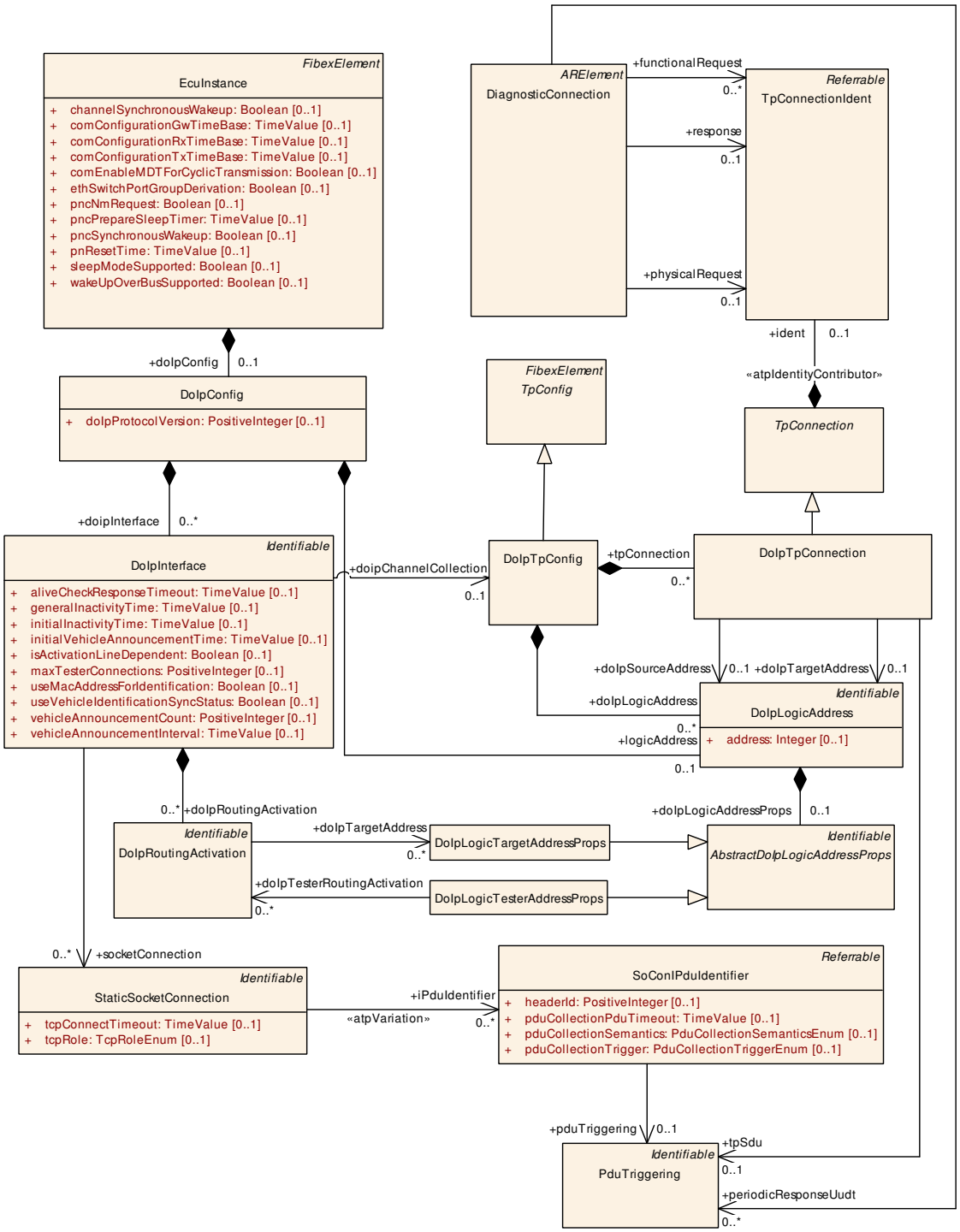

Figure 6.61: DoIP

Diagnostic messages are passed from a tester, through a DoIP gateway to the internal vehicle network. Before this happens the routing on a socket in the DoIP gateway needs to be activated. The tester sends a routing activation request and the DoIP entity responds to it. After the routing is activated the diagnostic communication starts.

[TPS_SYST_02303] Modeling of `DoIpRoutingActivation`s ⌈The DoIP routing activation possibilities in a `DoIpInterface` are described by the `DoIpRoutingActivation` that is aggregated in the role `doIpRoutingActivation` in the `DoIpInterface`.⌋

Class	**DoIpTpConfig**			
Note	This element defines exactly one DoIpTp Configuration that is used to configure all DoIPChannels available in a DoIpInterface. Each DoIPChannel describes a connection between a doIpSourceAddress and a doIpTargetAddress and the exchange of DcmIPdus between the PduR and DoIP. **Tags:** atp.recommendedPackage=TpConfigs			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
doIpLogic Address	DoIpLogicAddress	*	aggr	Collection of logical DoIP Addresses. **Tags:** atp.Splitkey=doIpLogicAddress, doIpLogic Address.shortName
tpConnection	DoIpTpConnection	*	aggr	Collection of unidirectional connections between a source address and a target address. **Tags:** atp.Splitkey=tpConnection, tp Connection.ident.shortName

Table 6.196: DoIpTpConfig

Class	**DoIpTpConnection**			
Note	A connection identifies the sender and the receiver of this particular communication. The DoIp module routes a tpSdu through this connection.			
Base	*ARObject*, *TpConnection*			
Aggregated by	DoIpTpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
doIpSource Address	DoIpLogicAddress	0..1	ref	Reference to the address of the sender of the tpSdu. **Tags:** atp.Splitkey=doIpSourceAddress
doIpTarget Address	DoIpLogicAddress	0..1	ref	Reference to the address of the receiver of the tpSdu. **Tags:** atp.Splitkey=doIpTargetAddress
tpSdu	PduTriggering	0..1	ref	This reference is used to describe the data exchange between DoIp and the PduR. **Tags:** atp.Splitkey=tpSdu

Table 6.197: DoIpTpConnection

Class	**DoIpLogicAddress**			
Note	The logical DoIP address.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpConfig.logicAddress, DoIpTpConfig.doIpLogicAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
address	Integer	0..1	attr	The logical DoIP address.

▽

△

Class	DoIpLogicAddress			
doIpLogic AddressProps	AbstractDoIpLogic AddressProps	0..1	aggr	Collection of additional LogicAddress properties. **Tags:** atp.Splitkey=doIpLogicAddressProps, doIpLogic AddressProps.shortName

Table 6.198: DoIpLogicAddress

[constr_9105] Existence of DoIpTpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConfig, the aggregation of at least one DoIpTpConnection in the role tpConnection shall exist.⌋

[constr_9106] Existence of DoIpTpConnection.doIpSourceAddress

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role doIpSourceAddress shall exist.⌋

[constr_9107] Existence of DoIpTpConnection.doIpTargetAddress

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role doIpTargetAddress shall exist.⌋

[constr_9108] Existence of DoIpTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈For each DoIpTpConnection, the reference to DoIpLogicAddress in the role tpSdu shall exist.⌋

Class	*AbstractDoIpLogicAddressProps* (abstract)			
Note	Abstract meta-class that collects common properties for all specialized DoIpLogicAddressProps.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	DoIpLogicTargetAddressProps, DoIpLogicTesterAddressProps			
Aggregated by	DoIpLogicAddress.doIpLogicAddressProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.199: AbstractDoIpLogicAddressProps

Class	DoIpLogicTargetAddressProps			
Note	This meta-class acts as a target for references to the DoIpLogicTargetAddress and collects DoIpLogic TargetAddress specific settings.			
Base	*ARObject*, *AbstractDoIpLogicAddressProps*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpLogicAddress.doIpLogicAddressProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.200: DoIpLogicTargetAddressProps

Class	DoIpLogicTesterAddressProps			
Note	This meta-class acts as a target for references to the DoIpLogicTesterAddress and collects DoIpLogic TesterAddress specific settings.			
Base	*ARObject*, *AbstractDoIpLogicAddressProps*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DoIpLogicAddress.doIpLogicAddressProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
doIpTester Routing Activation	DoIpRoutingActivation	*	ref	Reference to a DoIPRoutingActivation describing the possible routing activations of the DoIPTester. **Tags:** atp.Splitkey=doIpTesterRoutingActivation

Table 6.201: DoIpLogicTesterAddressProps

[constr_3212] Limitation of DoIpTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈DoIpTpConnection shall only reference PduTriggerings of DcmIPdus or UserDefinedIPdus in the role tpSdu.⌋

The diagnostic data is routed from the DoIP module to SoAd and back. The communication of diagnostic data over IP is described with StaticSocketConnections that contain SoConIPduIdentifiers with references to PduTriggerings of GeneralPurposePdus of category DoIP.

Please note that there is no connection between GeneralPurposePdus of category DoIP and the DoIpTpConnection in the System Description. The DoIP module evaluates the header of an incoming GeneralPurposePdu and knows from the included information the further processing.

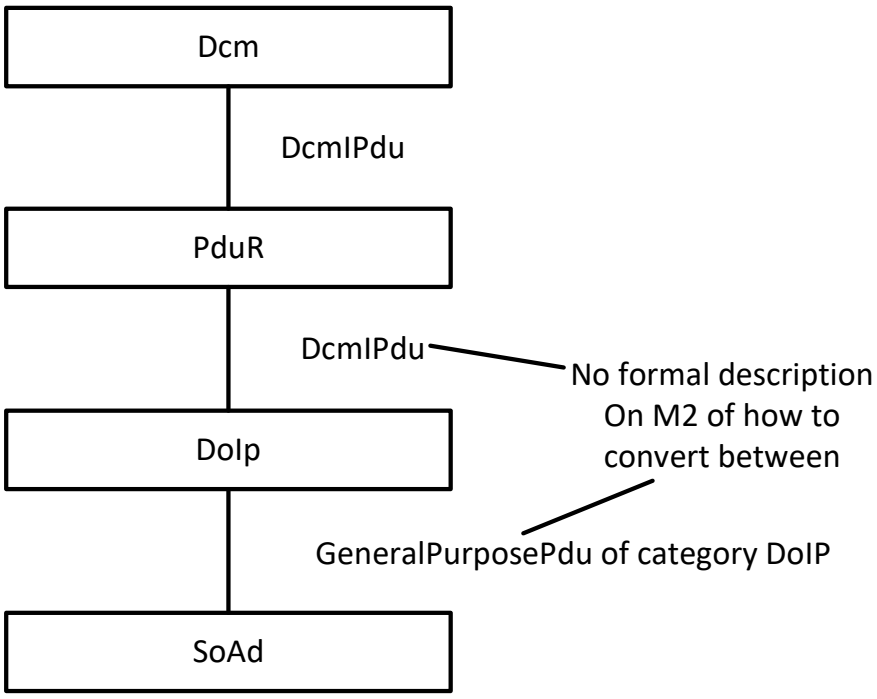

Figure 6.62: Routing of Dcm Pdus in AUTOSAR Stack for communication over IP

The DoIpInterface.socketConnection defines which StaticSocketConnection belongs to which DoIpInterface.

6.7.4.11 Transport Layer Security

AUTOSAR supports the configuration of *Transport Layer Security* for the information exchange between two *sockets*[2] that are modeled as ApplicationEndpoints.

Please note that currently the DTLS (***Datagram*** *Transport Layer Security*) variant is not supported on the *AUTOSAR classic platform*.

It is a common use case that only one end of a TLS-based connection is actually modeled in an AUTOSAR model. The other end may exist off-board, e.g. as a diagnostic tester.

It is therefore important that the modeling does not rely on or imply knowledge about both ends of such a TLS-based connection.

An AUTOSAR model that only describes one end of the communication is positively required to work, independently of the availability of a formal modeling of the other end.

AUTOSAR provides two alternatives for modeling TLS in System Models:

- Modeling using CryptoServicePrimitives (see section 6.7.4.11.1)
- Modeling using IANA [35] Parameters (see section 6.7.4.11.2).

[2]TLS connections are - by design - limited to a 1:1 pattern. A 1:n or n:1 communication pattern is not supported by TLS

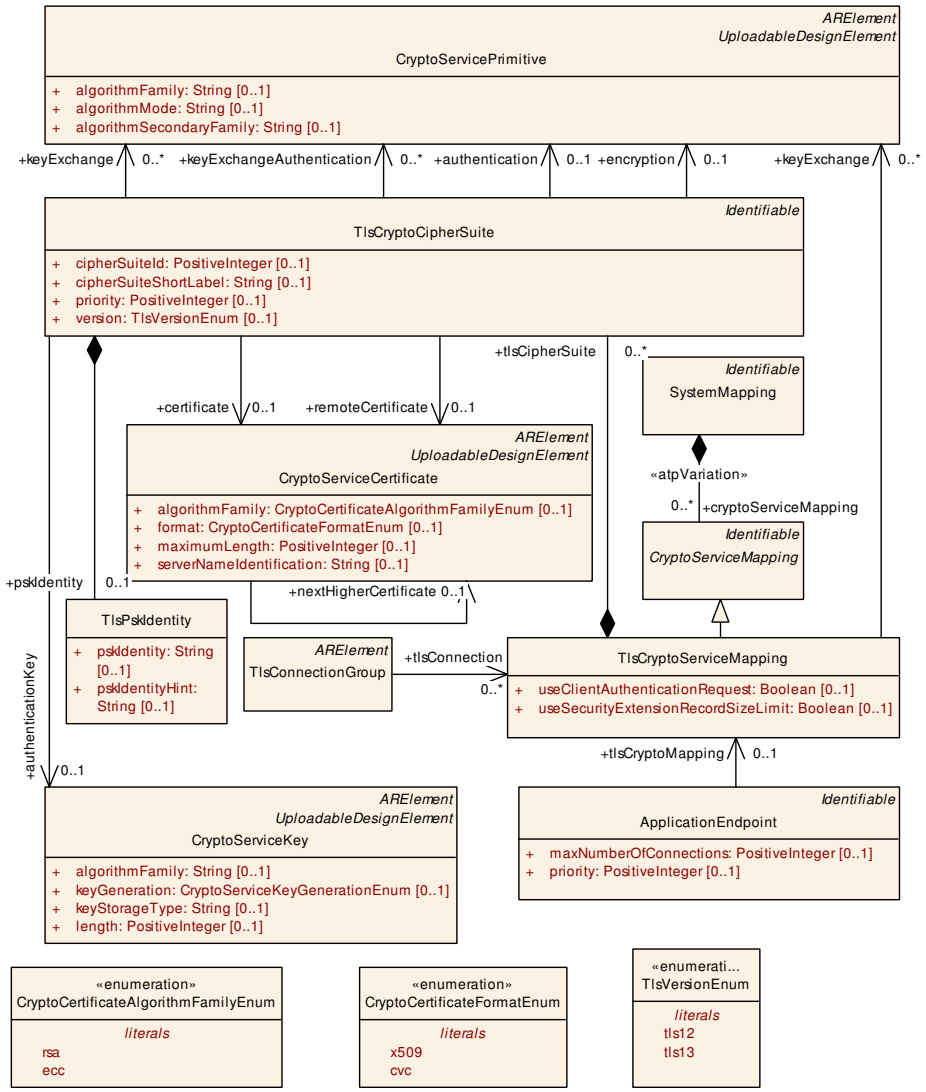

Figure 6.63: Modeling of crypto infrastructure for *Transport Layer Security*

Class	**TlsCryptoServiceMapping**			
Note	This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS).			
Base	*ARObject*, *CryptoServiceMapping*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cryptoServiceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **Tags:** atp.Splitkey=keyExchange
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	This aggregation represents the collection of supported cipher suites. **Tags:** atp.Splitkey=tlsCipherSuite, tlsCipherSuite.shortName

△

Class	**TlsCryptoServiceMapping**			
useClient Authentication Request	Boolean	0..1	attr	Defines if client authentication shall be applied for this TLS connection.
useSecurity Extension RecordSize Limit	Boolean	0..1	attr	Defines if the security extension for max_fragment_length shall be supported as defined in IETF RFC 8449, chapter 4.1.

Table 6.202: TlsCryptoServiceMapping

Class	**TlsConnectionGroup**			
Note	Defines a collection of TlsCryptoServiceMappings which will not be active at the same time during runtime. **Tags:** atp.recommendedPackage=TlsConnectionGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tlsConnection	TlsCryptoService Mapping	*	ref	Reference target is member of a TlsConnectionGroup. **Tags:** atp.Splitkey=tlsConnection

Table 6.203: TlsConnectionGroup

The attribute `TlsCryptoServiceMapping.useSecurityExtensionRecordSizeLimit` is defining the behavior for IETF RFC 8449 [36].

A TLS connection is established between two communication endpoints that assume the dedicated roles of *server* and *client*.

These roles cannot be swapped while the connection exists, i.e. a *server* remains the *server* for the full amount of time the connection exists.

[TPS_SYST_05029] Semantics of meta-class `TlsCryptoServiceMapping` ⌈As a sub-class of `CryptoServiceMapping`, meta-class `TlsCryptoServiceMapping` has the ability to collect the TLS-related configuration aspects from either the perspective of the *client* or the *server*.

In the case of TLS, the collection boils down to the aggregation of meta-class `TlsCryptoCipherSuite` in the role `tlsCipherSuite` plus the ability (by means of the role `keyExchange`) to define handshake properties that are shared for each of the aggregated `tlsCipherSuite`s.⌋

[constr_1670] Prohibition of usage of `tlsCryptoMapping` in case of UDP socket connections

Imposition time: CP: `IT_SysDesc`

⌈A `TlsCryptoServiceMapping` may only be referenced by an `ApplicationEndpoint` in the role `tlsCryptoMapping` if that `ApplicationEndpoint` aggregates a `TcpTp` in the role `tpConfiguration`.⌋

[constr_1671] Supported values of `TlsCryptoServiceMapping.category`

Imposition time: CP: `IT_SysDesc`

⌈The only supported values of attribute `TlsCryptoServiceMapping.category` are:

- **`TLS_SERVER`**: the `TlsCryptoServiceMapping` assumes the role of the *server* in the TLS connection.
- **`TLS_CLIENT`**: the `TlsCryptoServiceMapping` assumes the role of the *client* in the TLS connection.

⌋

[constr_5319] TCP endpoint using `TLS_SERVER` role can only serve provided service instances

Imposition time: CP: `IT_SysDesc`

⌈An `ApplicationEndpoint` that refers to `TlsCryptoServiceMapping` with `category` `TLS_SERVER` in the role `tlsCryptoMapping` is only allowed to be referenced by `ProvidedServiceInstance`s in the role `localUnicastAddress` in case that the `ProvidedServiceInstance` does not have a `remoteUnicastAddress` defined.⌋

[constr_5320] TCP endpoint using `TLS_CLIENT` role can only serve consumed service instances

Imposition time: CP: `IT_SysDesc`

⌈An `ApplicationEndpoint` that refers to `TlsCryptoServiceMapping` with `category` `TLS_CLIENT` in the role `tlsCryptoMapping` is only allowed to be referenced by `ConsumedServiceInstance`s in the role `localUnicastAddress` in case that the `ConsumedServiceInstance` does not have a `remoteUnicastAddress` defined.⌋

The reason for [constr_5319] and [constr_5320] is that in the Service Discovery case the TLS *client* needs to establish the TLS connection and a TCP endpoint can only take one role: `TLS_CLIENT` or `TLS_SERVER`. If a `TlsCryptoServiceMapping` would act as `TLS_CLIENT` and would refer to a `ProvidedServiceInstance` then this `TLS_CLIENT` would need to establish the TLS connection. But in this case the `TLS_CLIENT` would not know to which remote service *client* a connection needs to be established since different `ConsumedServiceInstance`s may directly call methods of the `ProvidedServiceInstance` without any registration.

The usage of a cipher suite in the context of setting up a TLS connection is formalized by means of meta-class `TlsCryptoCipherSuite`.

[TPS_SYST_05030] Semantics of `TlsCryptoCipherSuite` ⌈The creation of a TLS connection requires the usage of a suite of cryptographic operations in specific roles, also known as a *cipher suite*.

Meta-class TlsCryptoCipherSuite represents a given cipher suite for a TLS connection. TlsCryptoCipherSuite references meta-class CryptoServicePrimitive in three dedicated roles that represent the steps of the creation of a TLS connection.

More specifically, the cryptographic operations for setting up a TLS connection involve the following steps:

- **Key exchange**: these CryptoServicePrimitives may be used for the handshake phase of the TLS connection. Different alternatives exist for executing this phase and therefore the multiplicity of this reference is 0..*.
- **Authentication** of communication partners during the operational phase of the TLS connection. This part is similar to freshness calculation for SecOC-based communication. For this purpose a single CryptoServicePrimitive is used on each end of the communication.
- **Encryption** of content exchanged between the communication partners that have established the TLS connection. For this purpose a single CryptoServicePrimitive is used on each end of the communication.

⌋

Please note that according to TLS each TlsCryptoCipherSuite may support multiple PSKs or certificates. The model restricts the multiplicity of references from TlsCryptoCipherSuite to CryptoServiceCertificate and TlsPskIdentity to 0..1. A TLS configuration with multiple PSKs or Certificates per CipherSuite can be achieved by having multiple TlsCryptoCipherSuite's that share the same set of properties but reference different CryptoServiceCertificate's or TlsPskIdentity's, respectively.

Class	**TlsCryptoCipherSuite**			
Note	This meta-class represents a cipher suite for describing cryptographic operations in the context of establishing a connection of ApplicationEndpoints that is protected by TLS.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TlsCryptoServiceMapping.tlsCipherSuite, TlsSecureComProps.tlsCipherSuite			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the generation and verification of MACs. **Tags:** atp.Splitkey=authentication
authentication Key	CryptoServiceKey	0..1	ref	This reference identifies the private key or pre-shared key used for authentication. **Tags:** atp.Splitkey=authenticationKey
certificate	CryptoService Certificate	0..1	ref	This reference identifies the applicable local certificate. **Tags:** atp.Splitkey=certificate
cipherSuiteId	PositiveInteger	0..1	attr	Identification of the CipherSuite according to the IANA assignments list.

▽

△

Class	TlsCryptoCipherSuite			
cipherSuite ShortLabel	String	0..1	attr	Name of the CipherSuite according to the IANA assignments list.
ellipticCurve	CryptoEllipticCurve Props	*	ref	This references point to the properties of elliptic curves. **Tags:** atp.Splitkey=ellipticCurve
encryption	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the execution of encryption. **Tags:** atp.Splitkey=encryption
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **Tags:** atp.Splitkey=keyExchange
keyExchange Authentication	CryptoServicePrimitive	*	ref	This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm. **Tags:** atp.Splitkey=keyExchangeAuthentication
priority	PositiveInteger	0..1	attr	This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.
props	TlsCryptoCipherSuite Props	0..1	aggr	The aggregated TlsCryptoCipherSuiteProps provide details for the TLS Cipher Suite. **Tags:** atp.Splitkey=props, props.shortName
pskIdentity	TlsPskIdentity	0..1	aggr	Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key. **Tags:** atp.Splitkey=pskIdentity
remote Certificate	CryptoService Certificate	0..1	ref	This reference identifies the applicable remote certificate. **Tags:** atp.Splitkey=remoteCertificate
signature Scheme	CryptoSignature Scheme	*	ref	This reference points to the properties of a TLS Signature Scheme. **Tags:** atp.Splitkey=signatureScheme
version	TlsVersionEnum	0..1	attr	This attribute supports the definition of the applicable version of TLS.

Table 6.204: TlsCryptoCipherSuite

[constr_9270] Existence of `TlsCryptoCipherSuite.version`

Imposition time: CP: `IT_SysDesc`

⌈For each `TlsCryptoCipherSuite`, the attribute `version` shall exist.⌋

Enumeration	TlsVersionEnum
Note	This meta-class has the ability to identify a specific version of the transport-layer security (TLS) protocol.
Aggregated by	TlsCryptoCipherSuite.version
Literal	*Description*
tls12	TLS version 1.2 **Tags:** atp.EnumerationLiteralIndex=0
tls13	TLS version 1.3 **Tags:** atp.EnumerationLiteralIndex=2

Table 6.205: TlsVersionEnum

Class	TlsPskIdentity			
Note	This element is used to describe the pre-shared key shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.			
Base	*ARObject*			
Aggregated by	TlsCryptoCipherSuite.pskIdentity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pskIdentity	String	0..1	attr	This attribute provides the key identification.
pskIdentityHint	String	0..1	attr	This attribute provides the identity hint for a pre-shared key.

Table 6.206: TlsPskIdentity

[constr_9271] Existence of `TlsPskIdentity.pskIdentity`

Imposition time: CP: `IT_SysDesc`

⌈For each `TlsPskIdentity`, the attribute `pskIdentity` shall exist.⌋

Class	TlsCryptoCipherSuiteProps			
Note	This meta-class provides attributes to specify details of TLS Cipher Suites.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TlsCryptoCipherSuite.props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpTlsUse SecurityExtensionForce EncryptThen Mac	Boolean	0..1	attr	Defines if the security extension according to IETF RFC 7366 shall be supported. This is useful for cipher suites using CBC mode.

Table 6.207: TlsCryptoCipherSuiteProps

Class	CryptoEllipticCurveProps			
Note	This meta-class provides attributes to specify the properties of elliptic curves. **Tags:** atp.recommendedPackage=CryptoEllipticCurveProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
namedCurveId	PositiveInteger	0..1	attr	Defines the value of one specific NamedCurve Id.

Table 6.208: CryptoEllipticCurveProps

Class	CryptoSignatureScheme			
Note	This meta-class provides attributes to specify the TLS Signature Scheme. **Tags:** atp.recommendedPackage=CryptoSignatureSchemas			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
signature SchemeId	PositiveInteger	0..1	attr	Defines the value of one specific TLS Signature Scheme.

Table 6.209: CryptoSignatureScheme

[TPS_SYST_05031] Existence of `TlsCryptoCipherSuite.keyExchange` vs. `TlsCryptoServiceMapping.keyExchange` ⌈The role `TlsCryptoServiceMapping.keyExchange` has been introduced as an optimization.

It is assumed that the references for key exchange look pretty similar if not identical for many concrete `TlsCryptoCipherSuite`s.

Adding these references in an identical form to a bunch of `TlsCryptoCipherSuite`s does not really make sense. Therefore, `TlsCryptoServiceMapping` allows to define these references as well with the intention to make them valid for all `TlsCryptoServiceMapping.tlsCipherSuite`.

A mixture of references in the role `TlsCryptoCipherSuite.keyExchange` and `TlsCryptoServiceMapping.keyExchange` is supported for the case of a given collection of `TlsCryptoCipherSuite`s⌋

[TPS_SYST_05032] Semantics of `CryptoServiceCertificate` ⌈Meta-class `CryptoServiceCertificate` represents a cryptographic certificate needed for the creation of a TLS connection between *server* and *client*.⌋

Class	**CryptoServiceCertificate**			
Note	This meta-class represents the ability to model a cryptographic certificate. **Tags:** atp.recommendedPackage=CryptoServiceCertificates			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
algorithmFamily	CryptoCertificate AlgorithmFamilyEnum	0..1	attr	This attribute represents a description of the family of crypto algorithm used to generate public key and signature of the cryptographic certificate.
format	CryptoCertificateFormat Enum	0..1	attr	This attribute can be used to provide information about the format used to create the certificate
maximum Length	PositiveInteger	0..1	attr	This attribute represents the ability to define the maximum length of the certificate in bytes.
nextHigher Certificate	CryptoService Certificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain. **Tags:** atp.Splitkey=nextHigherCertificate
serverName Identification	String	0..1	attr	Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.

Table 6.210: CryptoServiceCertificate

Enumeration	**CryptoCertificateAlgorithmFamilyEnum**
Note	This meta-class defies possible cryptographic algorithm families used to create public keys and signatures within the certificate.
Aggregated by	CryptoServiceCertificate.algorithmFamily

▽

△

Enumeration	**CryptoCertificateAlgorithmFamilyEnum**
Literal	***Description***
ecc	The cryptographic operations in the certificate are executed using elliptic curves (ecc) **Tags:** atp.EnumerationLiteralIndex=2
rsa	The cryptographic operations in the certificate are executed using the RSA approach. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.211: CryptoCertificateAlgorithmFamilyEnum

Enumeration	**CryptoCertificateFormatEnum**
Note	This meta-class defines possible formats of cryptographic certificates.
Aggregated by	CryptoServiceCertificate.format
Literal	***Description***
cvc	The certificate has been created in Card Verifiable Certificate (CVC) format **Tags:** atp.EnumerationLiteralIndex=2
x509	The certificate is created in X.509 format. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.212: CryptoCertificateFormatEnum

[constr_1672] Existence of `TlsCryptoCipherSuite.certificate` and `TlsCryptoCipherSuite.pskIdentity` in the *server* role

Imposition time: CP: `IT_SysDesc`

⌈**Either**

- the reference to `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite.certificate`
- the aggregation of `TlsPskIdentity` in the role `TlsCryptoCipherSuite.pskIdentity`

shall exist if the `TlsCryptoCipherSuite` is aggregated by a `TlsCryptoServiceMapping` that has attribute `category` set to the value `TLS_SERVER`.⌋

In other words two different approaches are supported by TLS for the handling of key exchange: Pre-shared secret and certificate.

The *server* may optionally request a certificate from the *client*. If this option is not used then other documented approaches for completing the handshake phase are foreseen for the specific case.

[TPS_SYST_05033] Existence of `TlsCryptoCipherSuite.certificate` and `TlsCryptoCipherSuite.pskIdentity` in the *client* role ⌈The *client* (`TlsCryptoServiceMapping` has the attribute `category` set to the value `TLS_CLIENT`) has the following authentication options:

- the reference to `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite.certificate` exists,

- the aggregation of `TlsPskIdentity` in the role `TlsCryptoCipherSuite.pskIdentity` exists,
- neither `TlsCryptoCipherSuite.certificate` nor `TlsCryptoCipherSuite.pskIdentity` exists and `TlsCryptoServiceMapping.useClientAuthenticationRequest` is set to *false*. In this case the handshake is provided on the basis of the *server* certificate only.

⌋

In the pre-shared Key approach the *client* indicates which key to use by including a `pskIdentity` in the ClientKeyExchange message. To help the *client* in selecting which identity to use, the *server* can provide a `pskIdentityHint` in the ServerKeyExchange message. Please note that the usage of `pskIdentityHint`s is restricted for usage with TLS 1.2.

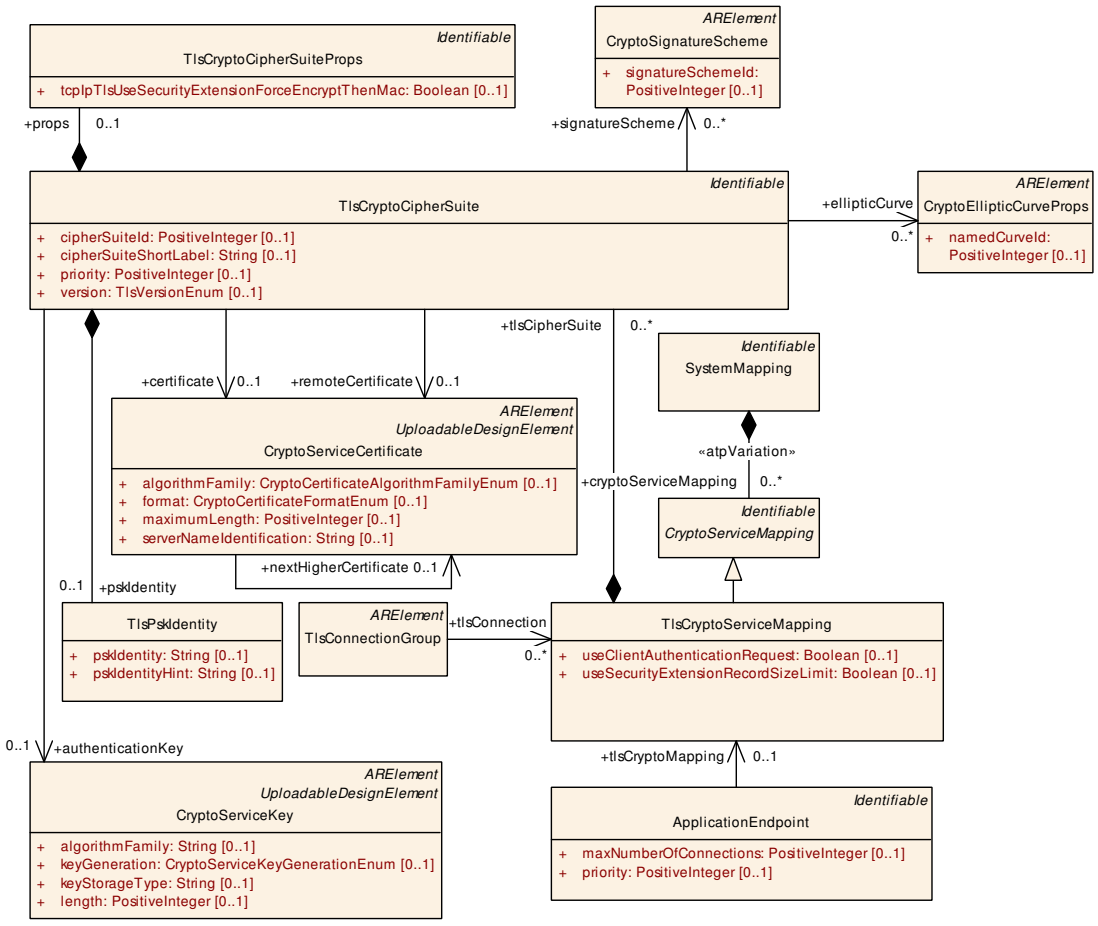

Figure 6.64: Modeling of crypto infrastructure for *Transport Layer Security*

AUTOSAR provides two alternatives for modeling TLS in System Models:

6.7.4.11.1 Modeling TLS using CryptoServicePrimitives

Instances of TlsCryptoServiceMapping and TlsCryptoCipherSuite allow to refer to CryptoServicePrimitives via

- TlsCryptoServiceMapping.keyExchange
- TlsCryptoCipherSuite.keyExchange
- TlsCryptoCipherSuite.keyExchangeAuthentication
- TlsCryptoCipherSuite.authentication
- TlsCryptoCipherSuite.encryption.

Setting up these references to CryptoServicePrimitives with specific properties allows to define *how* a TLS Connection and its associated TLS cipher suites are working by defining the CryptoServicePrimitive they employ. Instances of the CryptoServicePrimitive class directly map to CsmPrimitive containers in the Csm configuration, which makes the translation into Csm configurations straightforward.

Using CryptoServicePrimitive elements for modeling TLS, however, makes it hard to infer *what* a TLS Connection and its associated TLS cipher suites are from the information *how* they work. It is, for instance, difficult to determine the IANA id [35] of a TLS cipher suite by examining the properties of the CryptoServicePrimitive elements the associated TlsCryptoCipherSuite is referencing.

Defining CryptoServicePrimitive elements couples the TLS configuration of the System Model tightly to the Csm module configuration which demands deep knowledge of the Csm for setting up the TLS part of the system model. Moreover, software suppliers might deviate from the Csm standard when it comes to the configuration of CsmPrimitive containers. As a consequence, two stack vendors might require differently configured CryptoServicePrimitive elements for modeling the same cipher suite.

For all these shortcomings, it is recommended to define TLS connections that use standard cipher suites by defining IANA parameters [35] instead of referencing to CryptoServicePrimitives . Defining CryptoServicePrimitives can be the solution to model non-standard cipher suites, however they still require custom support from vendors.

6.7.4.11.2 Modeling TLS using IANA Parameters

IANA provides a set of TLS Parameters [35] to specify the properties of standardized cipher suites, which are partly also available in the TLS part of AUTOSAR System Models:

- TlsCryptoCipherSuite.cipherSuiteId
- TlsCryptoCipherSuite.cipherSuiteShortLabel

and, if the cipher suite requires key exchange:

- CryptoEllipticCurveProps.namedCurveId
- CryptoSignatureScheme.signatureSchemeId.

The specification of these parameters defines *what* a TLS Connection and their cipher suites are in contrast to the first approach which defines *how* they operate. The specification of a standardized cipher suite allows to derive the CsmPrimitives required for its operation without tight coupling to the specifics of the Csm configuration.

[constr_3668] Existence of TlsCryptoCipherSuite.cipherSuiteShortLabel

Imposition time: CP: IT_SysDesc

⌈If a TlsCryptoCipherSuite.cipherSuiteShortLabel is defined then:

- the attribute TlsCryptoCipherSuite.cipherSuiteId shall be defined as well
- the value of TlsCryptoCipherSuite.cipherSuiteShortLabel shall match the *Description* value corresponding to the *Value* field defined in TlsCryptoCipherSuite.cipherSuiteId according to TlsCryptoCipherSuite Parameter set defined in [35].

⌋

Common parameters:

The TLS part of the system model specifies a set of entities and parameters that can be configured in both alternatives. They allow the definition of certificates or pre-shared Keys which the cipher suites are using and the definition of extensions e.g. for increasing the security of a TLS connection:

- TlsCryptoCipherSuite.priority
- TlsCryptoCipherSuite.version
- TlsCryptoCipherSuite.pskIdentity
- TlsCryptoCipherSuite.certificate
- TlsCryptoCipherSuite.remoteCertificate
- TlsCryptoServiceMapping.useClientAuthenticationRequest
- TlsCryptoServiceMapping.useSecurityExtensionRecordSizeLimit
- All parameters specified in TlsCryptoCipherSuiteProps.

For Details on TlsCryptoCipherSuite.cipherSuiteId and TlsCryptoCipherSuite.cipherSuiteShortLabel see [35] section "TLS Cipher Suites".

For Details on CryptoEllipticCurveProps and namedCurveId see [35] section "TLS Supported Groups".

For Details on `CryptoSignatureScheme` and `signatureSchemeId` see [35] section "TLS SignatureScheme".

6.7.4.12 IPsec

IPsec is a protocol suite that provides cryptographic protection for IP datagrams in IPv4 and IPv6 network packets.

IPsec uses a security association to specify security properties that are shared between the communicating parties. The security association defines a relationship between two or more parties and determines which security services will be used to communicate securely. In other words the security association serves as a “contract” between the different devices.

A single security association protects data in one communication direction. Two security associations shall be present to secure traffic in both directions. Each security association can provide encryption, data integrity and data authentication.

In addition the senders and receivers of IP datagrams can determine the required protection for an IP packet according to IPsec security policies.

These are rules that define how datagrams are processed that are received by a device. For example, security policies are used to decide if a particular packet needs to be dropped or needs to be processed by IPsec.

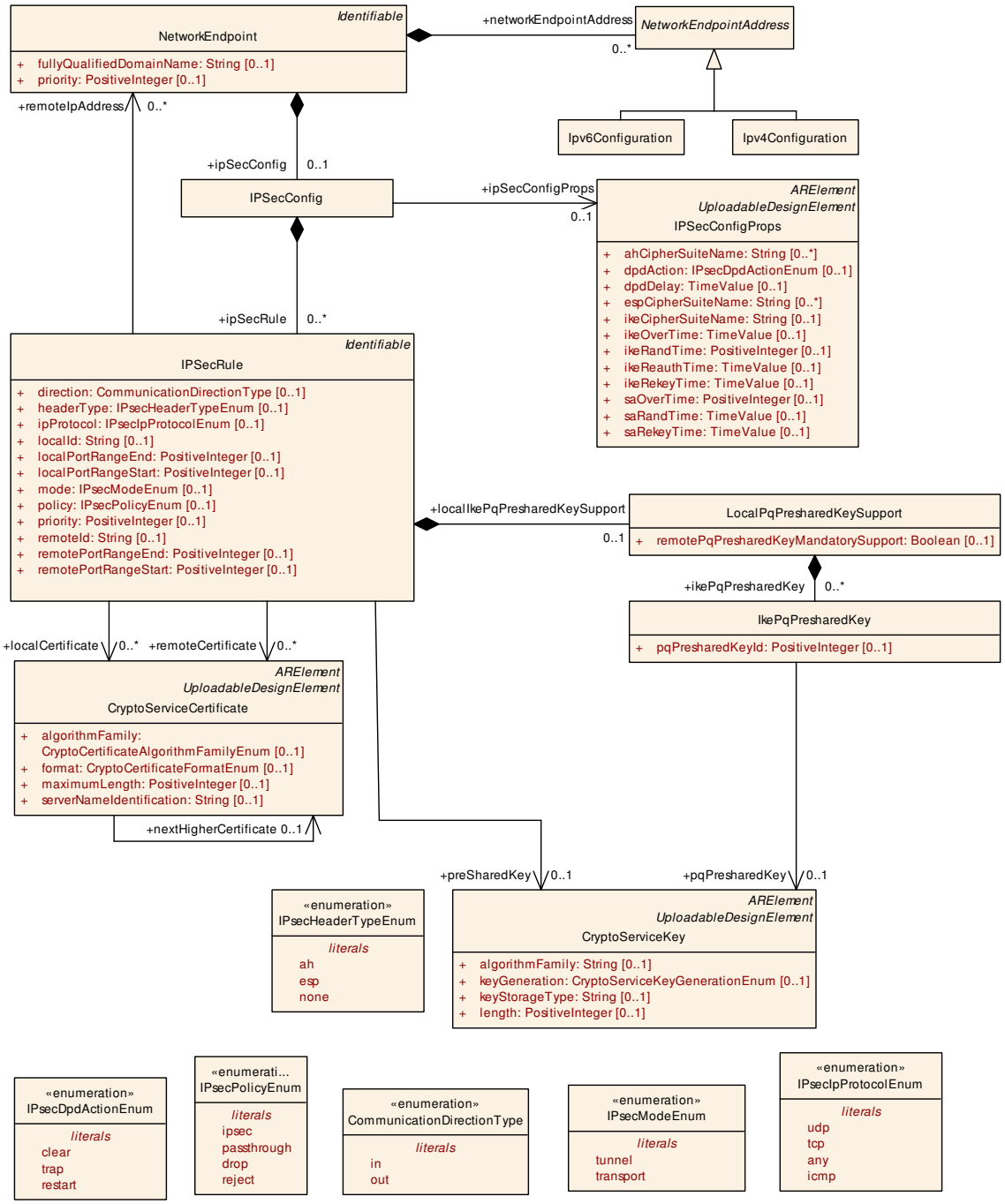

Figure 6.65: IPsec configuration model

[TPS_SYST_02265] Configuration of IPsec ⌈The `IPSecConfig` meta-class that is aggregated by a `NetworkEndpoint` in the role `ipSecConfig` provides the ability to define IPsec settings that are necessary to configure IPsec security associations and IPsec security policies.⌋

[TPS_SYST_02266] Definition of `IPSecRule`s ⌈The `IPSecConfig` meta-class may contain one or several `IPSecRule`s. Each `IPSecRule` defines the network connection that is monitored by IPsec by defining the local endpoint and the remote endpoint.

Each endpoint is defined by the IP Address and the Tcp/Udp Port. The communication direction for which the `IPSecRule` is valid is defined by the `direction` attribute.⌋

Class	**IPSecConfig**			
Note	IPsec is a protocol that is designed to provide "end-to-end" cryptographically-based security for IP network connections.			
Base	*ARObject*			
Aggregated by	NetworkEndpoint.ipSecConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ipSecConfig Props	IPSecConfigProps	0..1	ref	Global IPsec configuration settings that are valid for all IPSecRules that are defined on the NetworkEndpoint. **Tags:** atp.Splitkey=ipSecConfigProps
ipSecRule	IPSecRule	*	aggr	IPSec rules and filters that are defined in the IPSecConfig for a specific NetworkEndpoint. **Tags:** atp.Splitkey=ipSecRule, ipSecRule.shortName

Table 6.213: IPSecConfig

Class	**IPSecRule**			
Note	This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPSecConfig.ipSecRule			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
direction	CommunicationDirectionType	0..1	attr	This attribute defines the direction in which the traffic is monitored. If this attribute is not set a bidirectional traffic monitoring is assumed.
headerType	IPsecHeaderTypeEnum	0..1	attr	Header type specifying the IPsec security mechanism.
ipProtocol	IPsecIpProtocolEnum	0..1	attr	This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.
localCertificate	CryptoServiceCertificate	*	ref	This reference identifies the applicable certificate used for a local authentication. **Tags:** atp.Splitkey=localCertificate
localId	String	0..1	attr	This attribute defines how the local participant should be identified for authentication.
localIkePqPresharedKeySupport	LocalPqPresharedKeySupport	0..1	aggr	Post-quantum pre-shared key configuration to support IETF RFC 8784 to ensure resistance against post-quantum attacks by combining a post-quantum pre-shared key (PPK) with the exchanged information (via e.g., DH/ECDH) during key derivation. **Tags:** atp.Splitkey=localIkePqPresharedKeySupport
localPortRangeEnd	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an end value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.

▽

△

Class	**IPSecRule**			
localPortRange Start	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines a start value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
mode	IPsecModeEnum	0..1	attr	This attribute defines the type of the connection.
policy	IPsecPolicyEnum	0..1	attr	An IPsec policy defines the rules that determine which type of IP traffic needs to be secured using IPsec and how that traffic is secured.
preSharedKey	CryptoServiceKey	0..1	ref	This reference identifies the applicable cryptographic key used for authentication. **Tags:** atp.Splitkey=preSharedKey
priority	PositiveInteger	0..1	attr	This attribute defines the priority of the IPSecRule (SPD entry). The processing of entries is based on priority, starting with the highest priority "0".
remote Certificate	CryptoService Certificate	*	ref	This reference identifies the applicable certificate used for a remote authentication. **Tags:** atp.Splitkey=remoteCertificate
remoteId	String	0..1	attr	This attribute defines how the remote participant should be identified for authentication.
remoteIp Address	NetworkEndpoint	*	ref	Definition of the remote NetworkEndpoint. With this reference the connection between the local Network Endpoint and the remote NetworkEndpoint is described on which the traffic is monitored. **Tags:** atp.Splitkey=remoteIpAddress
remotePort RangeEnd	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an end value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
remotePort RangeStart	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines a start value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.

Table 6.214: IPSecRule

Class	**IPSecConfigProps**
Note	This element holds all the attributes for configuration of IPsec that are independent of specific IPsec rules. **Tags:** atp.recommendedPackage=IPSecConfigProps
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*

▽

△

Class	**IPSecConfigProps**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ahCipherSuite Name	String	*	attr	AH (Authentication Header) algorithm to be used for the connection, e.g. HMAC/SHA2-256
dpdAction	IPsecDpdActionEnum	0..1	attr	This attribute defines what to do if the peer is considered dead. If not configured "restart" shall be assumed.
dpdDelay	TimeValue	0..1	attr	This attribute describes the interval to check the liveness of a peer actively using IKEv2 INFORMATIONAL exchanges. Active DPD checking is only enforced if no IKE or ESP/AH packet has been received for the configured DPD delay. In not configured the value "5 minutes" shall be assumed.
espCipherSuite Name	String	*	attr	ESP (Encapsulating Security Payload) algorithm that provides encryption and optional authentication for the connection, e.g. AES-128+SHA2-256.
ikeCipherSuite Name	String	0..1	attr	IKE encryption/authentication algorithms to be used for the connection.
ikeOverTime	TimeValue	0..1	attr	This attribute describes the hard deadline when an SA becomes invalid in percentage. Example: ikeOverTime of max(ikeReauthTime, ikeRekey Time). Default: 10 %
ikeRandTime	PositiveInteger	0..1	attr	This attribute defines in percentage by how long before the expiration of ikeReauthTime and ikeRekeyTime will be rekeyed/reauthenticated. Default: 10%
ikeReauthTime	TimeValue	0..1	attr	This attribute defines the absolute time after which an IKE SA will be reauthenticated. 0 means reauthentication is disabled.
ikeRekeyTime	TimeValue	0..1	attr	This attribute defines the absolute time after which an IKE SA will be rekeyed. 0 means rekey is disabled.
saOverTime	PositiveInteger	0..1	attr	This attribute describes the hard deadline when an IPsec SA becomes invalid in percentage. Example: saOverTime * saRekeyTime. Default: 110%
saRandTime	TimeValue	0..1	attr	This attribute defines by how long before the expiration of saRekeyTime will be rekeyed.
saRekeyTime	TimeValue	0..1	attr	This attribute defines the absolute time after which an IPsec SA will be rekeyed. 0 means rekey is disabled.

Table 6.215: IPSecConfigProps

Enumeration	**IPsecIpProtocolEnum**
Note	Definition of supported TcpIp protocols that are supported in Security Policy Database (SPD) entries in IPSec configurations.
Aggregated by	IPSecRule.ipProtocol
Literal	***Description***

▽

△

Enumeration	**IPsecIpProtocolEnum**
any	ANY protocol **Tags:** atp.EnumerationLiteralIndex=3
icmp	Internet Control Message Protocol (ICMP) **Tags:** atp.EnumerationLiteralIndex=2
tcp	TCP Protocol **Tags:** atp.EnumerationLiteralIndex=1
udp	UDP Protocol **Tags:** atp.EnumerationLiteralIndex=0

Table 6.216: IPsecIpProtocolEnum

[TPS_SYST_02270] Definition of general IPsec configuration settings ⌈General configuration properties that are independent of particular `IPSecRule`s are collected in the `IPSecConfigProps` element that is referenced from the `IPSecConfig` in the role `ipSecConfigProps`.⌋

[TPS_SYST_02267] IPsec policy ⌈The `IPSecRule.policy` attribute defines how IP packets are handled that are going over the network connection defined by the `IPSecRule`. In detail it defines whether the IP packet is processed by IPsec or not.⌋

Enumeration	**IPsecPolicyEnum**
Note	Defines the filter actions that are supported by IPsec.
Aggregated by	IPSecRule.policy
Literal	***Description***
drop	Signifying that packets should be discarded **Tags:** atp.EnumerationLiteralIndex=3
ipsec	Signifying that packets should be protected. **Tags:** atp.EnumerationLiteralIndex=1
passthrough	Signifying that no IPsec processing should be done at all. **Tags:** atp.EnumerationLiteralIndex=2
reject	Signifying that packets should be discarded and a diagnostic ICMP returned. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.217: IPsecPolicyEnum

IPsec can be configured to operate in two different modes, Tunnel and Transport mode. With tunnel mode, the entire IP packet is protected by IPsec. IPsec wraps the original packet, encrypts it and adds a new IP header to it. The tunnel mode is most commonly used between VPN gateways and the IP addresses of the newly added outer IP header are that of the VPN Gateways. In other words the traffic between the two VPN Gateways is protected and each gateway acts as a proxy for the hosts behind it.

The transport mode provides the protection of the Data Payload of the IP datagram with an AH or ESP header. The IP Header remains the same and IPsec inserts its header between the IP header and the upper level headers. The IPsec transport mode

can be used when securing traffic between two hosts or between a host and a VPN gateway.

[TPS_SYST_02268] IPsec mode ⌈The `IPSecRule.mode` attribute defines whether the IP packet is processed in the `transport` or `tunnel` mode.⌋

Please note that AUTOSAR currently supports only the `transport mode` as configuration option.

Enumeration	**IPsecModeEnum**
Note	This enumeration describes the supported IPSec modes.
Aggregated by	IPSecRule.mode
Literal	***Description***
transport	Signifying that the IPSec transport mode is used. With the transport mode the original IP header is retained and only the IP payload and ESP trailer is encrypted. **Tags:** atp.EnumerationLiteralIndex=1
tunnel	Signifying that the IPSec tunnel mode is used. With tunnel mode, the entire original IP packet is protected by IPSec. This means IPSec wraps the original packet, encrypts it, adds a new IP header and sends it to the other side. **Tags:** atp.EnumerationLiteralIndex=0

Table 6.218: IPsecModeEnum

IPsec uses two protocols:

- AH - Authentication Header
- ESP - Encapsulating Security Payload

The AH protocol provides a mechanism for authentication only and authenticates the entire IP packet, including the outer IP header.

The ESP protocol provides data confidentiality (encryption) and/or authentication (data integrity, data origin authentication, and replay protection).

When ESP is used in transport mode, the IP payload is encrypted and the original IP header is moved to the front of the message. The ESP header is inserted after the IP header and is signed together with the IP payload. The original IP header remains unprotected.

When ESP is used in tunnel mode a new IP Header is created and the ESP header is added in front of the original IP Packet. The entire original IP packet is encrypted and signed in this mode.

[TPS_SYST_02269] IPsec AH and ESP protocol configuration ⌈In the `IPSecRule` it is possible to define the IPsec protocol that shall be used to protect IP packets that are going over the defined network connection. The attribute `headerType` defines whether AH, ESP or neither one is used.⌋

Enumeration	**IPsecHeaderTypeEnum**
Note	IPsec Header Type options
Aggregated by	IPSecRule.headerType
Literal	***Description***
ah	Authentication Header (AH) **Tags:** atp.EnumerationLiteralIndex=0
esp	Encapsulating Security Payloads (ESP) **Tags:** atp.EnumerationLiteralIndex=1
none	No header **Tags:** atp.EnumerationLiteralIndex=2

Table 6.219: IPsecHeaderTypeEnum

[TPS_SYST_02271] IPsec AH and ESP CipherSuites ⌈The attributes `ahCipherSuiteName` and `espCipherSuiteName` define the supported AH and ESP algorithms.⌋

The naming convention for `ahCipherSuiteName`, `espCipherSuiteName` and `IPSecConfigProps.ikeCipherSuiteName` shall follow the naming convention for cryptographic primitives that is defined in [37].

[TPS_SYST_02272] IPsec Internet Key Exchange protocol configuration ⌈In the `IPSecRule` it is possible to define how IKE protocol authenticates the remote party and how the local party authenticates itself to the remote party. In other words both sides use the same method. The usage of the `IPSecRule.preSharedKey` reference defines that the pre-shared key is used. The usage of the `IPSecRule.localCertificate` and `IPSecRule.remoteCertificate` defines that Digital Signature Authentication is used.⌋

[constr_5163] Existence of attribute `IPSecRule.headerType`

Imposition time: CP: `IT_SysDesc`

⌈For each `IPSecRule`, the attribute `headerType` shall exist.⌋

[constr_5164] Existence of attribute `IPSecRule.ipProtocol`

Imposition time: CP: `IT_SysDesc`

⌈For each `IPSecRule`, the attribute `ipProtocol` shall exist.⌋

[constr_5165] Existence of attribute `IPSecRule.policy`

Imposition time: CP: `IT_SysDesc`

⌈For each `IPSecRule`, the attribute `policy` shall exist.⌋

Please note that the supported IKE CipherSuites are configured with the `IPSecConfigProps.ikeCipherSuiteName`. The `IPSecConfigProps` contains additional IKE specific configuration settings.

Enumeration	**IPsecDpdActionEnum**
Note	Potential Dead Peer Detection (Dpd) Actions
Aggregated by	IPSecConfigProps.dpdAction
Literal	***Description***
clear	Deletes the SA. **Tags:** atp.EnumerationLiteralIndex=0
restart	Immediately tries to establish the connection. **Tags:** atp.EnumerationLiteralIndex=2
trap	tries to establish the connection after traffic is sent to the peer. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.220: IPsecDpdActionEnum

[TPS_SYST_02273] Protection of `ProvidedServiceInstance` by IPsec ⌈To describe the protection of an `ProvidedServiceInstance` by IPsec the `ProvidedServiceInstance` needs to point to an `ApplicationEndpoint` in the `localUnicastAddress` role and this `ApplicationEndpoint` shall point to a `NetworkEndpoint` that aggregates the `IPSecConfig` and describes the IPsec Security Associations.⌋

[TPS_SYST_02274] Protection of `ConsumedServiceInstance` by IPsec ⌈To describe the protection of an `ConsumedServiceInstance` by IPsec the `ConsumedServiceInstance` needs to point to an `ApplicationEndpoint` in the `localUnicastAddress` role and this `ApplicationEndpoint` shall point to a `NetworkEndpoint` that aggregates the `IPSecConfig` and describes the IPsec Security Associations.⌋

Please note that IP Multicast protection by IPsec is not supported.

Class	**LocalPqPresharedKeySupport**			
Note	Post-quantum pre-shared key configuration to support IETF RFC 8784 to ensure resistance against post-quantum attacks by combining a post-quantum pre-shared key (PPK) with the exchanged information (via e.g., DH/ECDH) during key derivation.			
Base	*ARObject*			
Aggregated by	IPSecRule.localIkePqPresharedKeySupport			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ikePqPresharedKey	IkePqPresharedKey	*	aggr	List of post-quantum pre-shared keys (PPK) and their post-quantum pre-shared key identifiers (PKK_IDs). **Tags:** atp.Splitkey=ikePqPresharedKey
remotePqPresharedKeyMandatorySupport	Boolean	0..1	attr	This boolean attribute defines whether the usage of the post-quantum pre-shared keys is mandatory or not for communication with the local node.

Table 6.221: LocalPqPresharedKeySupport

Class	**IkePqPresharedKey**			
Note	Post-quantum pre-shared key (PPK) identified by post-quantum pre-shared key identifier (PKK_ID).			
Base	*ARObject*			
Aggregated by	LocalPqPresharedKeySupport.ikePqPresharedKey			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pqPreshared Key	CryptoServiceKey	0..1	ref	Reference to PPK post-quantum pre-shared key. **Tags:** atp.Splitkey=pqPresharedKey
pqPreshared KeyId	PositiveInteger	0..1	attr	This attribute defines the post-quantum pre-shared key identifier for the referenced pqPresharedKey.

Table 6.222: IkePqPresharedKey

[constr_3827] Existence of `IkePqPresharedKey.pqPresharedKeyId`

Imposition time: CP: `IT_SysDesc`

⌈For each `IkePqPresharedKey`, the attribute `pqPresharedKeyId` shall exist.⌋

[constr_3828] Existence of `IkePqPresharedKey.pqPresharedKey`

Imposition time: CP: `IT_SysDesc`

⌈For each `IkePqPresharedKey`, the reference `pqPresharedKey` shall exist.⌋

6.7.4.13 EthernetFrameType based communication

Please note that with the introduction of the TcpIp Bsw module the description of `AbstractEthernetFrame`s is no longer necessary for configuration of the AUTOSAR TcpIp Stack.

Nevertheless it may be useful to describe the Ethernet FrameType based communication in some cases, e.g. if a new basic software module like Ieee1722Tp is used that is located above the EthDrv and parallel to the TcpIp Stack. The Ethernet FrameType based communication shall be described without `Pdu`s.

[constr_3113] `AbstractEthernetFrame` shall not have a `PduToFrameMapping`

Imposition time: CP: `IT_SysDesc`

⌈It is not allowed to map `Pdu`s into `AbstractEthernetFrame`s.⌋

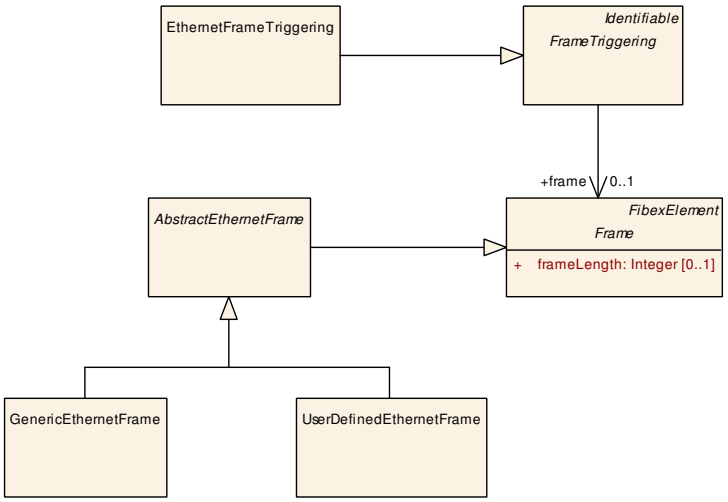

Figure 6.66: EthernetFrameType based communication

Class	***AbstractEthernetFrame*** (abstract)			
Note	Ethernet specific attributes to the Frame.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	GenericEthernetFrame, UserDefinedEthernetFrame			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.223: AbstractEthernetFrame

Class	**EthernetFrameTriggering**			
Note	Ethernet specific Frame element.			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.224: EthernetFrameTriggering

Class	**GenericEthernetFrame**			
Note	This element is used for EthernetFrames without additional attributes that are routed by the EthIf. **Tags:** atp.recommendedPackage=Frames			
Base	*ARObject*, *AbstractEthernetFrame*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.225: GenericEthernetFrame

Class	UserDefinedEthernetFrame			
Note	UserDefinedEthernetFrame allows the description of a frame-based communication to Complex Drivers that are located above the EthDrv. **Tags:** atp.recommendedPackage=Frames			
Base	*ARObject*, *AbstractEthernetFrame*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.226: UserDefinedEthernetFrame

6.7.4.14 Restriction in usage of ProvidedServiceInstance.majorVersion

If several ProvidedServiceInstances are defined with the same serviceIdentifier and different majorVersions and these ProvidedServiceInstances are referencing the same ApplicationEndpoint with the localUnicastAddress (ProvidedServiceInstances are located on the same Socket) then particular restrictions apply for the system configuration.

In such a scenario the same MessageId (headerId) may be used for ServiceInterface elements like an Event in the different ProvidedServiceInstances that have different majorVersions. It may happen that the same headerId is used for Pdus that represent the same ServiceInterface element even if the Pdu layout differs in the different MajorVersion ProvidedServiceInstances because for example the dataType of the Service Interface element was changed from one MajorVersion to the other.

The reason for these restrictions is the AUTOSAR Architecture in the Classic Platform where one part of the SOME/IP Header is evaluated in the SOME/IP Transformer (RequestId, Protocol Version, Interface Version, Message Type, Return Code) and the other part in the SocketAdaptor (MessageId, Length). This means that the Socket Adaptor is not able to evaluate the MajorVersion in the Pdu.

The following restrictions apply in case of **ClientServer communication** (Service Interface Methods, Field Getter, Field Setter): If two or more ProvidedServiceInstances are defined using the same serviceIdentifier and different majorVersions and these ProvidedServiceInstances are referencing the same ApplicationEndpoint via localUnicastAddress then the destination IP address, the destination port number, and the Level 4 protocol (Udp/Tcp) fields of header of IP packets containing call PDUs that are sent to the ProvidedServiceInstances are identical. In such a scenario, the ProvidedServiceInstances may still use identical method Ids (thus identical Header Ids) if the following condition applies:

At any point in time only one of the ProvidedServiceInstances is active, and only clients of that ProvidedServiceInstance send request PDUs to the ProvidedServiceInstance.

In all other cases, the method ids of the two ProvidedServiceInstances should not overlap.

The following restrictions apply in case of **SenderReceiver communication** (Service Interface Events and Notifier): If two or more ProvidedServiceInstances are defined using the same serviceIdentifier and different majorVersions, and these ProvidedServiceInstances are referencing the same ApplicationEndpoint via localUnicastAddress then the source IP address, the source port number, and the Level 4 protocol (Udp) fields of header of IP packets containing event PDUs that are sent to the clients of the ProvidedServiceInstances are identical. In such a scenario, the ProvidedServiceInstances may use identical method Ids (i.e. identical Header Ids) if at least one of the following conditions holds for any pair of the ProvidedServiceInstances:

a) At any point in time only one of the ProvidedServiceInstances is active

b) If two or more ProvidedServiceInstances can send Events at the same time, the ProvidedServiceInstances may still use identical method Ids if at least one of the following IP header fields of the IP packet containing the event is different for any pair of Event PDUs identified by the same Method Id:

b1) Destination IP address (== IP address of client)

b2) Destination port number (== client port number)

In all other cases, the method ids of the two PSIs should not overlap.

In other words if several ProvidedServiceInstances with the same serviceIdentifier and different majorVersions are located on the same Socket for transmission of particular Events, then the different ProvidedServiceInstances need to use a different client Addresses (different Port or IP Address) or the Events of all those ProvidedServiceInstances need to have unique MethodIds.

6.7.4.15 Summary of supported configuration options between ApplicationEndpoints

The following table summarizes the configuration options that are supported by the System Template and were described in the chapters above.

[TPS_SYST_02396] Summary of supported configuration options between ApplicationEndpoints ⌈

Communication Scenario	Configuration on Provider Side	Configuration on Consumer Side	Usage of SD module
dynamic Service Discovery configuration: Service Consumer finds Service Provider at runtime	ProvidedService-Instance with defined localUnicastAddress and without remoteUnicastAddress	ConsumedService-Instance with defined localUnicastAddress and without remoteUnicastAddress	SD module is used and is configured.

▽

△

dynamic event subscription with known peers: Service Consumer knows Service Provider from configuration. But Service Consumer needs to subscribe to EventGroups to receive events from Service Provider	ProvidedService-Instance with defined localUnicastAddress and with defined remoteUnicastAddress	ConsumedService-Instance with defined localUnicastAddress and with defined remoteUnicastAddress.	SD module is used and is configured. Service Offer and Service Find do not need to be performed. The SocketConnections in the SoAd can be setup without ServiceDiscovery since the Service Provider and Service Consumer know each other from the configuration. The subscription to EventGroups is still performed via ServiceDiscovery
Static configuration: Addresses (IP Address, Port) of Data Provider and Data Consumer are predefined in the configuration. Exchanged data is predefined and assigned to the StaticSocketConnection that is defined between Service Consumer and ServiceProvider	Definition of StaticSocketConnection on the SocketAddress belonging to the ServiceProvider. StaticSocketConnection defines the target of the communication with the remoteAddress. A ProvidedService-Instance is not defined.	Definition of StaticSocketConnection. A ConsumedService-Instance is not defined.	SD module is not used

⌋

6.7.4.16 Firewall

AUTOSAR Classic supports the integration of a firewall as a host firewall or as a firewall inside an Ethernet switch.

[TPS_SYST_03096] Firewall on a Host ECU ⌈If an EcuInstance refers to at least one StateDependentFirewall in the role firewallRule, then a firewall is configured for that EcuInstance.⌋

[TPS_SYST_03097] Firewall on an Ethernet Switch ⌈If a CouplingElement where the couplingType is set to the value CouplingElementEnum.switch refers to at least one StateDependentFirewall in the role firewallRule, then a firewall is configured for that Ethernet switch.⌋

The AUTOSAR Firewall module offers a mechanism to dynamically enable/disable Firewall rules at runtime. The Firewall module is connected to the BswM, which switches the Firewall State to allow for dynamic firewall behavior based on the current vehicle state (e.g. driving, parking, in a diagnostic session). The System Template models this aspect with the StateDependentFirewall.firewallStateModeDeclaration reference to a ModeDeclaration of a ModeDeclarationGroup. Please note that in this case the modeling of the Firewall State Machine is not done via a ModeDeclarationGroupPrototype but directly via a ModeDeclarationGroup that is an ARElement.

[constr_9317] `StateDependentFirewall.firewallStateModeDeclaration` reference restriction

Imposition time: CP: `IT_SysDesc`

⌈Each `StateDependentFirewall` shall only reference `ModeDeclaration`s in the role `firewallStateModeDeclaration` that are aggregated by the same `ModeDeclarationGroup`.⌋

In other words all used Firewall States that are modeled as `ModeDeclaration`s shall be collected in the same `ModeDeclarationGroup`.

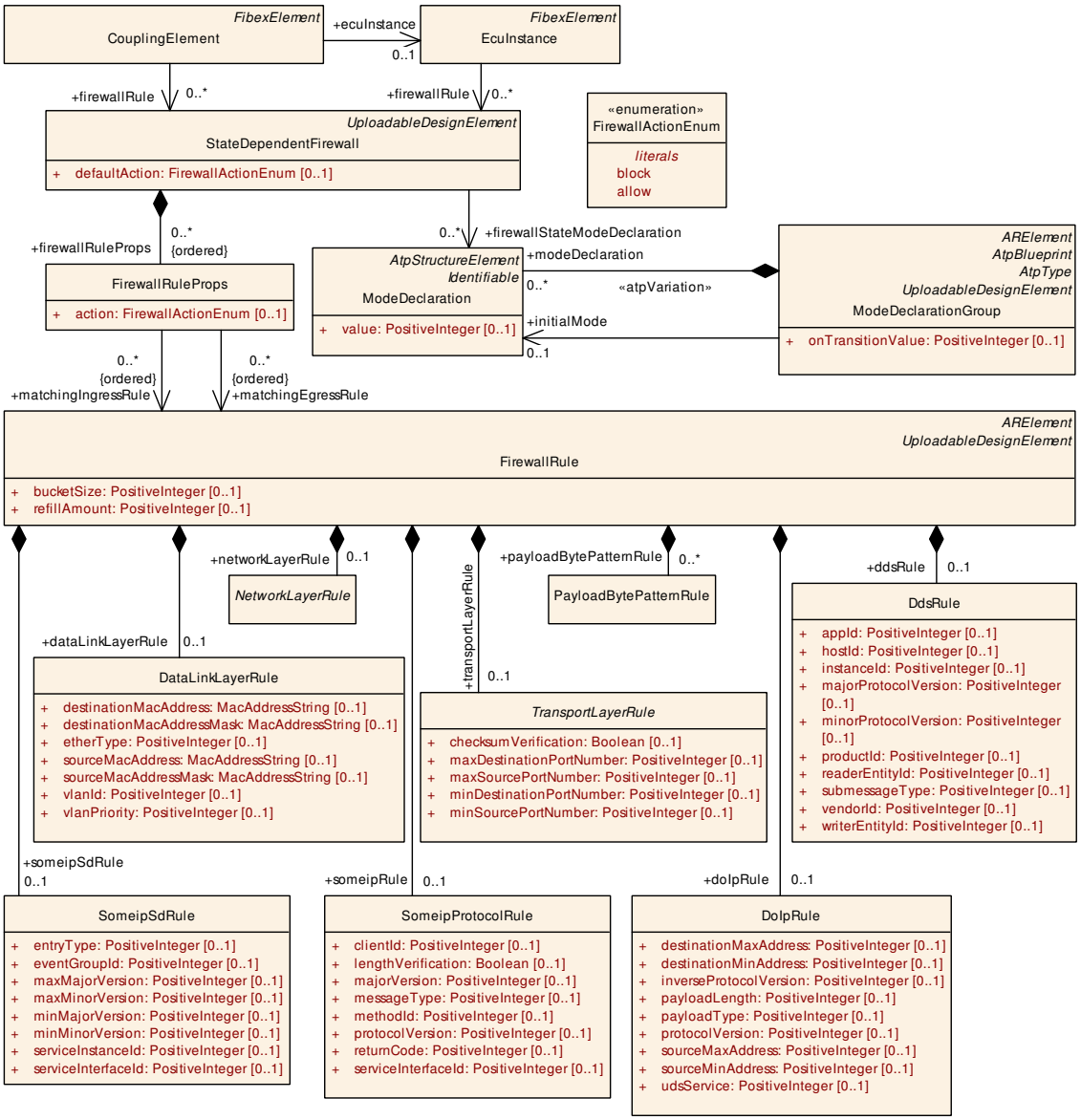

Figure 6.67: Firewall configuration

The details on the configuration of the `StateDependentFirewall` and `FirewallRule`s can be found in the Manifest specification for the Adaptive platform [38].

Class	**StateDependentFirewall**			
Note	Firewall rules that are defined in a firewall state **Tags:** atp.recommendedPackage=StateDependentFirewallRules			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultAction	FirewallActionEnum	0..1	attr	This attribute defines a defaultAction in case that the VehicleMode is not yet set.
firewallRule Props (ordered)	FirewallRuleProps	*	aggr	Collection of firewall rules that apply in the vehicle mode
firewallState Mode Declaration	ModeDeclaration	*	ref	Reference to firewall states in which the Firewall is active. If one of the referenced ModeDeclarations is the current firewall state then the firewall rule shall be considered as active. **Tags:** atp.Splitkey=firewallStateModeDeclaration This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.227: StateDependentFirewall

Class	**FirewallRuleProps**			
Note	Firewall rule that is defined by an action that is performed if the referenced pattern matches.			
Base	*ARObject*			
Aggregated by	StateDependentFirewall.firewallRuleProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
action	FirewallActionEnum	0..1	attr	Action that is performed by the firewall if the matching Rule is fulfilled.
matchingEgress Rule (ordered)	FirewallRule	*	ref	This element defines an egress rule expression against which the network traffic is matched.
matching IngressRule (ordered)	FirewallRule	*	ref	This element defines an ingress rule expression against which the network traffic is matched.

Table 6.228: FirewallRuleProps

Enumeration	**FirewallActionEnum**
Note	List of actions that the Firewall is able to perform.
Aggregated by	FirewallRuleProps.action, StateDependentFirewall.defaultAction
Literal	***Description***
allow	Firewall allows the communication **Tags:** atp.EnumerationLiteralIndex=1
block	Firewall blocks the communication **Tags:** atp.EnumerationLiteralIndex=0

Table 6.229: FirewallActionEnum

Class	FirewallRule			
Note	Firewall Rule that defines the control information in individual packets. **Tags:** atp.recommendedPackage=FirewallRules			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bucketSize	PositiveInteger	0..1	attr	This attribute defines the capacity of the queue for rate limitation (leaky-bucket Algorithm).
dataLinkLayer Rule	DataLinkLayerRule	0..1	aggr	Configuration of rules on the Data Link Layer **Tags:** atp.Splitkey=dataLinkLayerRule
ddsRule	DdsRule	0..1	aggr	Configuration of firewall rules for DDS. **Tags:** atp.Splitkey=ddsRule
doIpRule	DoIpRule	0..1	aggr	Configuration of firewall rules for DoIP messages **Tags:** atp.Splitkey=doIpRule
networkLayer Rule	NetworkLayerRule	0..1	aggr	Configuration of rules on the Network Layer **Tags:** atp.Splitkey=networkLayerRule
payloadByte PatternRule	PayloadBytePattern Rule	*	aggr	Configuration of generic firewall rules **Tags:** atp.Splitkey=payloadBytePatternRule
refillAmount	PositiveInteger	0..1	attr	This attribute defines the output rate that describes how many packets leave the queue per second (leaky-bucket Algorithm).
someipRule	SomeipProtocolRule	0..1	aggr	Configuration of firewall rules for SOME/IP messages **Tags:** atp.Splitkey=someipRule
someipSdRule	SomeipSdRule	0..1	aggr	Configuration of firewall rules for SOME/IP Service Discovery messages **Tags:** atp.Splitkey=someipSdRule
transportLayer Rule	TransportLayerRule	0..1	aggr	Configuration of rules on the Transport Layer **Tags:** atp.Splitkey=transportLayerRule

Table 6.230: FirewallRule

Class	DataLinkLayerRule			
Note	Configuration of filter rules on the DataLink layer			
Base	*ARObject*			
Aggregated by	FirewallRule.dataLinkLayerRule			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
destinationMac Address	MacAddressString	0..1	attr	Filter to match packets with the destination MAC address.
destinationMac AddressMask	MacAddressString	0..1	attr	Filter to match packets with the destination MAC address range. The destinationMacAddress with the destination MacAddressMask defines the MAC address range.
etherType	PositiveInteger	0..1	attr	Filter to match packets based on the EtherType field in the Ethernet frame. The EtherType is used to indicate which protocol is encapsulated in the payload of the frame.
sourceMac Address	MacAddressString	0..1	attr	Filter to match packets with the source MAC address.
sourceMac AddressMask	MacAddressString	0..1	attr	Filter to match packets with the source MAC address range. The sourceMacAddress with the sourceMac AddressMask defines the MAC address range.



△

Class	**DataLinkLayerRule**			
vlanId	PositiveInteger	0..1	attr	Filter of packets with a specific VlanId.
vlanPriority	PositiveInteger	0..1	attr	Filter of packets with a specific Vlan priority.

Table 6.231: DataLinkLayerRule

Class	**DdsRule**			
Note	Configuration of a DDS firewall rule			
Base	*ARObject*			
Aggregated by	FirewallRule.ddsRule			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
appId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the appId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.
hostId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the hostId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.
instanceId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the instanceId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.
majorProtocol Version	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the major ProtocolVersion in the DDSI-RTPS header matches.
minorProtocol Version	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the minor ProtocolVersion in the DDSI-RTPS header matches.
productId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the productId in the DDSI-RTPS header matches.
readerEntityId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the readerEntity ID in a DDSI-RTPS submessage matches
submessage Type	PositiveInteger	0..1	attr	Defines the allowed submessage type in the DDSI-RTPS message
vendorId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the vendorId in the DDSI-RTPS header matches.
writerEntityId	PositiveInteger	0..1	attr	Filter for DDSI-RTPS messages in which the writerEntity ID in a DDSI-RTPS submessage matches

Table 6.232: DdsRule

Class	**DoIpRule**			
Note	Configuration of a generic firewall rule			
Base	*ARObject*			
Aggregated by	FirewallRule.doIpRule			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
destinationMax Address	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the destination Address is smaller or equal than destinationMaxAddress.
destinationMin Address	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the destination Address is greater or equal than destinationMinAddress.
inverseProtocol Version	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the inverseprotocolVersion in the DoIP header matches.
payloadLength	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the payload Length in the DoIP header matches.
payloadType	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the payloadType in the DoIP header matches.

▽

△

Class	**DoIpRule**			
protocolVersion	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the protocol Version in the DoIP header matches.
sourceMax Address	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the source Address is smaller or equal than sourceMaxAddress.
sourceMin Address	PositiveInteger	0..1	attr	Filter to match DoIP messages in which the source Address is greater or equal than sourceMinAddress..
udsService	PositiveInteger	0..1	attr	Filter to match DoIP messages that contain the uds Service.

Table 6.233: DoIpRule

Class	***NetworkLayerRule*** (abstract)			
Note	Configuration of filter rules on the Network layer			
Base	*ARObject*			
Subclasses	Ipv4Rule, Ipv6Rule			
Aggregated by	FirewallRule.networkLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.234: NetworkLayerRule

Class	**Ipv4Rule**			
Note	Configuration of filter rules on IPv4 level.			
Base	*ARObject*, *NetworkLayerRule*			
Aggregated by	FirewallRule.networkLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
checksum Verification	Boolean	0..1	attr	Defines whether a Ipv4 header checksum verification is performed or not.
destinationIp Address	Ip4AddressString	0..1	attr	Filter to match packets with the destination IPv4 address.
destination NetworkMask	Ip4AddressString	0..1	attr	Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destination NetworkMask defines the IP address range.
differentiated ServiceCode Point	PositiveInteger	0..1	attr	Filter to match packets with a DSCP value.
doNotFragment	Boolean	0..1	attr	Filter to match packets that have the doNotFragment bit in the Header set.
explicit Congestion Notification	PositiveInteger	0..1	attr	Filter to match packets with a ECN code point.
icmpRule	IcmpRule	0..1	aggr	Configuration of filter rules for ICMP (Internet Control Message Protocol). **Tags:** atp.Splitkey=icmpRule
internetHeader Length	PositiveInteger	0..1	attr	Filter to match packets with a minimum ipv4 header length.
moreFragments	Boolean	0..1	attr	Filter to match packets that have the moreFragments flag in the Header set.
protocol	PositiveInteger	0..1	attr	Filter to match packets with a IP protocol number .
sourceIp Address	Ip4AddressString	0..1	attr	Filter to match packets with the source IPv4 address.

▽

△

Class	**Ipv4Rule**			
sourceNetwork Mask	Ip4AddressString	0..1	attr	Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetwork Mask defines the IP address range.
ttlMax	PositiveInteger	0..1	attr	Filter to match packets with a maximum ttl value (TimeTo Live defines the lifetime of data on the network).
ttlMin	PositiveInteger	0..1	attr	Filter to match packets with a minimum ttl value (TimeTo Live defines the lifetime of data on the network).

Table 6.235: Ipv4Rule

Class	**Ipv6Rule**			
Note	Configuration of filter rules on IPv6 level.			
Base	*ARObject*, *NetworkLayerRule*			
Aggregated by	FirewallRule.networkLayerRule			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
destinationIp Address	Ip6AddressString	0..1	attr	Filter to match packets with the destination IPv6 address.
destination NetworkMask	Ip6AddressString	0..1	attr	Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destination NetworkMask defines the MAC address range.
flowLabel	PositiveInteger	0..1	attr	Filter to match packets with a defined flow label.
hopLimit	PositiveInteger	0..1	attr	Filter to match packets with a minimum hop limit.
icmpRule	IcmpRule	0..1	aggr	Configuration of filter rules for ICMP (Internet Control Message Protocol). **Tags:** atp.Splitkey=icmpRule
nextHeader	PositiveInteger	0..1	attr	Filter to match packets with a defined type of an extension header.
sourceIp Address	Ip6AddressString	0..1	attr	Filter to match packets with the source IPv6 address.
sourceNetwork Mask	Ip6AddressString	0..1	attr	Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetwork Mask defines the IP address range.
trafficClass	PositiveInteger	0..1	attr	Filter to match packets with a defined traffic class or priority.

Table 6.236: Ipv6Rule

Class	**IcmpRule**			
Note	Configuration of filter rules for ICMP (Internet Control Message Protocol).			
Base	*ARObject*			
Aggregated by	Ipv4Rule.icmpRule, Ipv6Rule.icmpRule			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
checksum Verification	Boolean	0..1	attr	Defines whether a Icmp header checksum verification is performed or not.
code	PositiveInteger	0..1	attr	Filter to match packets with the Icmp code.
type	PositiveInteger	0..1	attr	Filter to match packets with the Icmp type.

Table 6.237: IcmpRule

Class	PayloadBytePatternRule			
Note	Configuration of a generic firewall rule that defines the individual bytes of a message that shall match.			
Base	*ARObject*			
Aggregated by	FirewallRule.payloadBytePatternRule			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
payloadByte PatternRulePart	PayloadBytePattern RulePart	*	aggr	Configuration of bytes in the message, **Tags:** atp.Splitkey=payloadBytePatternRulePart

Table 6.238: PayloadBytePatternRule

Class	PayloadBytePatternRulePart			
Note	Configuration of one byte in the datagram,			
Base	*ARObject*			
Aggregated by	PayloadBytePatternRule.payloadBytePatternRulePart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
offset	PositiveInteger	0..1	attr	This attribute defines the byte offset in the datagram (start byte of the Ethernet frame, i.e. offset 0 corresponds to the first byte of the destination MAC address).
value	PositiveInteger	0..1	attr	This attribute defines the byteValue (0..255) in the datagram.

Table 6.239: PayloadBytePatternRulePart

Class	SomeipProtocolRule			
Note	Configuration of SOME/IP firewall rules			
Base	*ARObject*			
Aggregated by	FirewallRule.someipRule			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientId	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the clientId in the SOME/IP header matches.
length Verification	Boolean	0..1	attr	Defines whether length verification is performed or not.
majorVersion	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the majorVersion in the SOME/IP header matches.
messageType	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the messageType in the SOME/IP header matches.
methodId	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the methodId in the SOME/IP header matches.
protocolVersion	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the protocol Version in the SOME/IP header matches.
returnCode	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the returnCode in the SOME/IP header matches.
serviceInterface Id	PositiveInteger	0..1	attr	Filter for SOME/IP messages in which the service InterfaceId in the SOME/IP header matches.

Table 6.240: SomeipProtocolRule

Class	SomeipSdRule
Note	Configuration of SOME/IP Service Discovery firewall rules
Base	*ARObject*
Aggregated by	FirewallRule.someipSdRule

▽

△

Class	**SomeipSdRule**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
entryType	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the entryType in the SOME/IP header matches.
eventGroupId	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the eventGroup Id in the SOME/IP header matches.
maxMajor Version	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the Major Version in the SOME/IP header is smaller or equal than maxMajorVersion.
maxMinor Version	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the Minor Version in the SOME/IP header is smaller or equal than maxMinorVersion.
minMajor Version	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the Major Version in the SOME/IP header is greater or equal than minMajorVersion.
minMinor Version	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the Minor Version in the SOME/IP header is greater or equal than minMinorVersion.
serviceInstance Id	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the service InstanceId in the SOME/IP header matches.
serviceInterface Id	PositiveInteger	0..1	attr	Filter for SOME/IP SD messages in which the service InterfaceId in the SOME/IP header matches.

Table 6.241: SomeipSdRule

Class	***TransportLayerRule*** (abstract)			
Note	Configuration of filter rules on Transport Layer level.			
Base	*ARObject*			
Subclasses	TcpRule, UdpRule			
Aggregated by	FirewallRule.transportLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
checksum Verification	Boolean	0..1	attr	Defines whether checksum verification is performed or not.
maxDestination PortNumber	PositiveInteger	0..1	attr	Filter to match packets with the maximum destination UDP/TCP port number.
maxSourcePort Number	PositiveInteger	0..1	attr	Filter to match packets with the maximum source UDP/ TCP port number.
minDestination PortNumber	PositiveInteger	0..1	attr	Filter to match packets with the minimum destination UDP/TCP port number.
minSourcePort Number	PositiveInteger	0..1	attr	Filter to match packets with the minimum source UDP/ TCP port number.

Table 6.242: TransportLayerRule

Class	**TcpRule**			
Note	Configuration of TCP filter rules.			
Base	*ARObject*, *TransportLayerRule*			
Aggregated by	FirewallRule.transportLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf ParallelTcp Sessions	PositiveInteger	0..1	attr	This attribute defines the maximal number of TCP Sessions that are allowed to be established.

▽

△

Class	**TcpRule**			
state Management BasedOnTcp Flags	Boolean	0..1	attr	This attribute defines whether the StateManagement is based on TCP flags or not.
timeoutCheck	PositiveInteger	0..1	attr	This attribute defines the TCP Session timeout in seconds

Table 6.243: TcpRule

Class	**UdpRule**			
Note	Configuration of UDP filter rules.			
Base	*ARObject*, *TransportLayerRule*			
Aggregated by	FirewallRule.transportLayerRule			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.244: UdpRule

6.7.4.17 ACL check

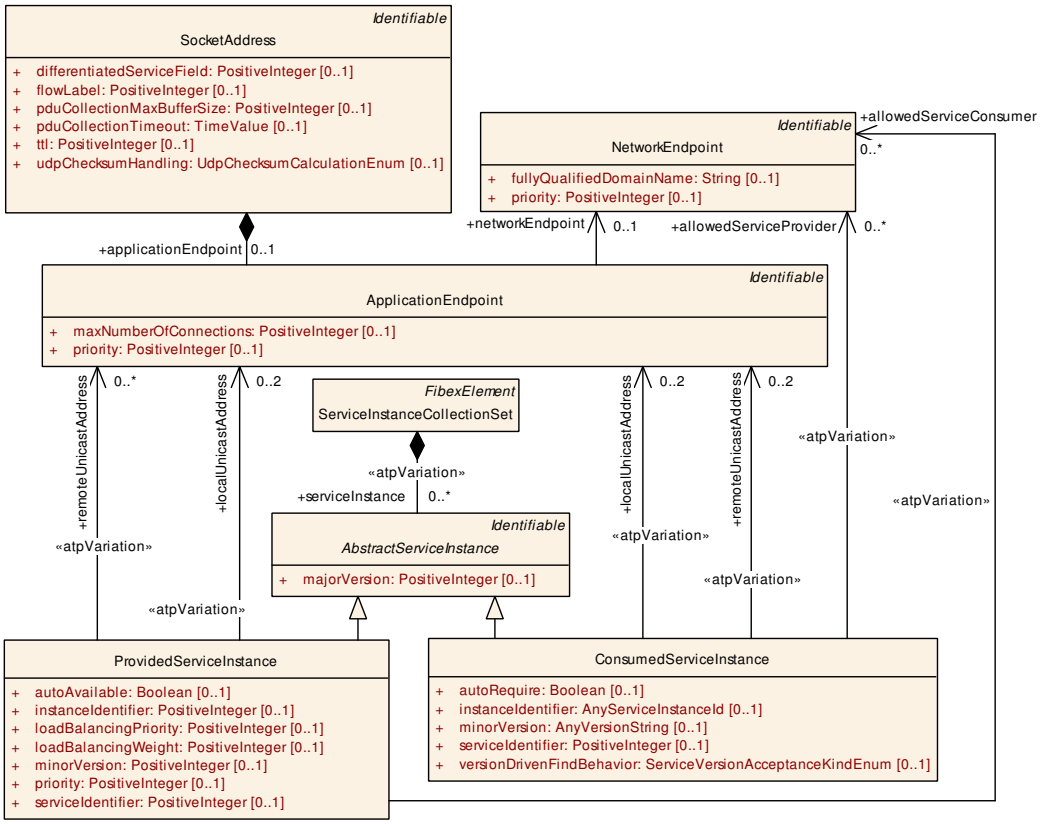

Figure 6.68: ACL check configuration

The ACL check in the AUTOSAR Service Discovery module allows to limit the Ethernet communication, so that a service instance can only be accessed by predefined and known communication partners.

[TPS_SYST_02401] ACL check definition for a `ProvidedServiceInstance` ⌈The `ProvidedServiceInstance.allowedServiceConsumer` reference allows to define a list of `NetworkEndpoint`s on which `ConsumedServiceInstance`s can be located that are allowed to access the `ProvidedServiceInstance`.⌋

[TPS_SYST_02402] ACL check definition for a `ConsumedServiceInstance` ⌈The `ConsumedServiceInstance.allowedServiceProvider` reference allows to define a `NetworkEndpoint` on which `ProvidedServiceInstance` can be located that is allowed to provide data to the `ConsumedServiceInstance`.⌋

6.8 Transport Layer

In AUTOSAR, the Transport Layer has two main purposes: The segmentation and reassembly of messages that are too long to fit into one frame on the underlying communication cluster, and the re-use of fixed frame identifiers for different message content.

According to the AUTOSAR Layered Software Architecture [39], each type of communication cluster has its own definition of the Transport Layer. Consequently, the peculiarities of the cluster types are addressed in the System Template by having different detailed models for FlexRay, CAN, LIN and J1939. However, all models are embedded into the communication model: They use specialized classes of TpConfig as a root element into the TP configuration.

[TPS_SYST_01099] Context of TpConfig ⌈A TpConfig element is existing always in the context of exactly one CommunicationCluster.⌋

All Transport Layers will take IPdus as input elements, which will be transferred in the form of one or more NPdus. A TpConnection (FlexrayTpConnection, CanTpConnection, LinTpConnection, J1939TpConnection) identifies a connection link between different communication nodes and routes the Pdus between them.

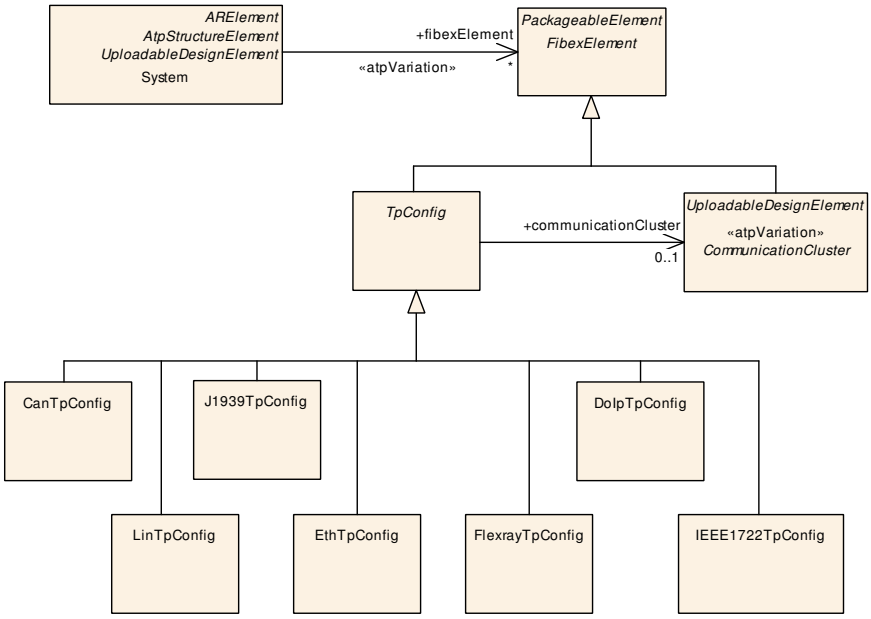

Figure 6.69: Transport Layer Overview

Examples in chapter 6.8.8 and chapter 6.8.9 illustrate the usage of the TP model.

Class	***TpConfig*** (abstract)
Note	Contains all configuration elements for AUTOSAR TP.
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

△

Class	TpConfig (abstract)			
Subclasses	CanTpConfig, DoIpTpConfig, EthTpConfig, FlexrayTpConfig, IEEE1722TpConfig, J1939TpConfig, LinTpConfig, SomeipTpConfig			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communicationCluster	CommunicationCluster	0..1	ref	A TpConfig is existing always in the context of exactly one CommunicationCluster. **Tags:** atp.Splitkey=communicationCluster

Table 6.245: TpConfig

Class	TpAddress			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpAddress, J1939TpConfig.tpAddress, LinTpConfig.tpAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpAddress	Integer	0..1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.

Table 6.246: TpAddress

[constr_9226] Existence of TpConfig.communicationCluster

Imposition time: CP: IT_SysDesc

⌈For each TpConfig, the reference to CommunicationCluster in the role communicationCluster shall exist.⌋

[constr_9227] Existence of TpAddress.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each TpAddress, the attribute tpAddress shall exist.⌋

[constr_3025] Usage of NPdus in TpConnections

Imposition time: CP: IT_SysDesc

⌈In case several TpConnections use the same Frame ID for their communication needs only one NPdu element per Frame Id shall exist. This constraint applies for all supported AUTOSAR transport protocols (CanTp, LinTp, FrTp, and J1939Tp).⌋

Note: Depending on the capabilities of the Basic Software implementations of Tp and Interface the ECU Configuration of the respective BSW Modules may utilize more communication elements (NPdus).

Example for an allowed System Template description where the same FrameId is used by two different TpConnections:

```
TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu2 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu1 ----> FrameId1
```

The following Ecu configuration with additional NPdus can still be derived from the above system description:

```
TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu3 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu4 ----> FrameId1
```

[constr_3090] TpSdu transmission on a PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈The IPdu that is referenced by a TpConnection in the role tpSdu shall be referenced by exactly one PduTriggering aggregated on the PhysicalChannel of the TpConnection.⌋

The corresponding PduTriggering for the IPdu referenced from the TpConnection in the role tpSdu is aggregated by the PhysicalChannel which points to the same CommunicationConnector which is referenced by TpNode that this TpConnection points to.

Please note that with [constr_3090] the multiple transmission of the same TpSdu over a specific channel using TP is only possible if several IPdus and TpConnections are created.

6.8.1 Transport Layer Routing

The transformations in the TP modules take a significant amount of time and resources and therefore two different Transport Layer routing approaches are supported by AUTOSAR.

[TPS_SYST_01100] TP routing without using transport protocol modules (low-level routing) ⌈The behavior can be optimized if source and target use the same transport protocol (e.g. CanTp-to-CanTp routing). In this case the inbound NPDU can be directly forwarded to the PduR and then sent on the outbound bus without any (resource consuming) TP module involvement.⌋

[TPS_SYST_01101] TP routing using transport protocol modules ⌈In case that transport protocol modules are involved in the routing operation the incoming NPDUs need to be:

- forwarded to corresponding inbound TP module and reassembled into an SDU (represented as IPdu)
- the SDU needs to be forwarded to the PduR
- the PduR routes the SDU to the outgoing TP module
- the outbound TP module segments the SDU into NPDUs which are then sent on the target bus.

⌋

6.8.2 FlexRay ISO Transport Layer

The FlexRay ISO 10681-2 Transport Layer supports multiple sessions, i.e. multiple segmented transfers can be handled at the same time. Thus, multiple FlexrayTpConnections can be defined on the same ECU. Each FlexrayTpConnection is controlled by configuration parameters defined in FlexrayTpConnectionControl.

[TPS_SYST_01102] FlexrayTpConnectionControl reuse ⌈The same FlexrayTpConnectionControl may be reused for an arbitrary number of FlexrayTpConnections.⌋

A FlexrayTpConnection defines the way of communication between a sender and a receiver and uses a FlexrayTpPduPool of NPdus to transmit data to the FlexRay Interface.

[TPS_SYST_01103] FlexrayTpConnection shall specify one txPduPool ⌈Each FlexrayTpConnection shall specify one txPduPool with at least one nPdu.⌋

In order to achieve a higher bandwidth a txPduPool may contain more than one transmit NPdu, e.g. if all referenced NPdus are transmitted in different FlexrayFrames in the same cycle.

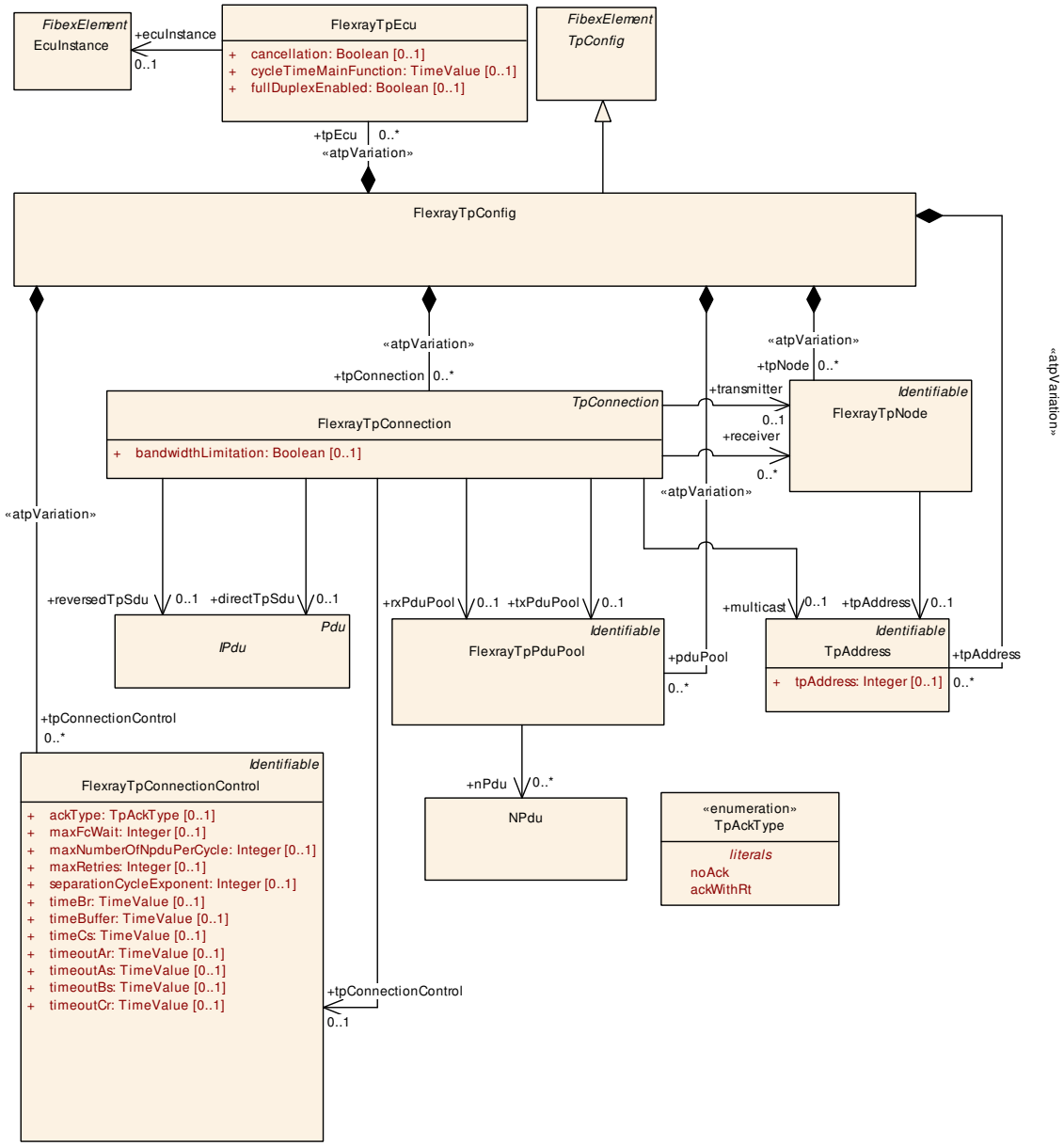

Figure 6.70: FlexRay ISO Transport Layer Configuration (TransportProtocols: FlexRay-IsoTransportProtocol)

`FlexrayTpConnection`s are specifically used for communication between one source and one or several target devices. These communication partners are specified using the `transmitter` and `receiver` associations to `FlexrayTpNode`s, providing the diagnostic `tpAddress` and the connection to the topology.

[TPS_SYST_01104] **`FlexrayTpConnection` with several receivers** ⌈In case of several receivers a multicast `tpAddress` shall be used.⌋

The actual payload to be transported by the `FlexrayTpConnection` is specified by using either one or two references to `IPdu`s, depending on whether the connection shall be used unidirectional (one reference) or bidirectional (two references).

Class	**FlexrayTpConfig**			
Note	This element defines exactly one FlexRay ISO TP Configuration. One FlexRayTpConfig element shall be created for each FlexRay Network in the System that uses Flex Ray Iso Tp. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
pduPool	FlexrayTpPduPool	*	aggr	Configuration of FlexRay TP Pdu Pools. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduPool, pduPool.shortName, pdu Pool.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpAddress	TpAddress	*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	FlexrayTpConnection	*	aggr	Configuration of FlexRay TP Connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection Control	FlexrayTpConnection Control	*	aggr	Configuration of FlexRay TP Connection Controls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnectionControl, tpConnection Control.shortName, tpConnectionControl.variation Point.shortLabel vh.latestBindingTime=postBuild
tpEcu	FlexrayTpEcu	*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpEcu, tpEcu.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	FlexrayTpNode	*	aggr	Senders and receivers of FlexRay TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.247: FlexrayTpConfig

[constr_9228] Existence of FlexrayTpConfig.pduPool

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of FlexrayTpPduPool in the role pduPool shall exist at least once.⌋

[constr_9229] Existence of FlexrayTpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of TpAddress in the role tpAddress shall exist at least once.⌋

[constr_9230] Existence of FlexrayTpConfig.tpEcu

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConfig, the aggregation of FlexrayTpEcu in the role tpEcu shall exist at least once.⌋

Class	**FlexrayTpConnectionControl**			
Note	Configuration parameters to control a FlexRay TP connection. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpConnectionControl			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ackType	TpAckType	0..1	attr	This parameter defines the type of acknowledgement which is used for the specific channel.
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of Flow Control N-PDUs with FlowState "WAIT".
maxNumberOf NpduPerCycle	Integer	0..1	attr	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.
maxRetries	Integer	0..1	attr	This parameter defines the maximum number of retries (if retry is configured for the particular channel).
separationCycle Exponent	Integer	0..1	attr	Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.
timeBr	TimeValue	0..1	attr	Time (in seconds) until transmission of the next Flow Control N-PDU.
timeBuffer	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to get a Tx or Rx buffer. This parameter is equivalent to the temporal distance between two FC.WT N-Pdus in case the buffer request returns busy. Specified in seconds.
timeCs	TimeValue	0..1	attr	Time (in seconds) until transmission of the next ConsecutiveFrame NPdu / LastFrame NPdu.
timeoutAr	TimeValue	0..1	attr	This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the Flex Ray Interface on the receiver side (for FC or AF). Specified in seconds.

▽

△

Class	**FlexrayTpConnectionControl**			
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the Flex Ray Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.

Table 6.248: FlexrayTpConnectionControl

Enumeration	**TpAckType**
Note	Type of Acknowledgement.
Aggregated by	FlexrayTpConnectionControl.ackType
Literal	*Description*
ackWithRt	Acknowledgement with retry. **Tags:** atp.EnumerationLiteralIndex=0
noAck	No acknowledgement. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.249: TpAckType

Class	**FlexrayTpConnection**			
Note	A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	FlexrayTpConfig.tpConnection			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bandwidth Limitation	Boolean	0..1	attr	Specifies whether the connection requires a bandwidth limitation or not.
directTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=directTpSdu
multicast	TpAddress	0..1	ref	TP address for 1:n connections. **Tags:** atp.Splitkey=multicast
receiver	FlexrayTpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver

▽

△

Class	FlexrayTpConnection			
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction. **Tags:** atp.Splitkey=reversedTpSdu
rxPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpRx PduPool) which are defined for receiving data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the rxPduPool holds the actually received NPdus. In case this connection is applied to the receiver the rxPduPool holds the actually sent NPdus. **Tags:** atp.Splitkey=rxPduPool
tpConnection Control	FlexrayTpConnection Control	0..1	ref	Reference to the connection control. **Tags:** atp.Splitkey=tpConnectionControl
transmitter	FlexrayTpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter
txPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpTx PduPool) which are defined for sending data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the txPduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus. **Tags:** atp.Splitkey=txPduPool

Table 6.250: FlexrayTpConnection

[constr_9231] Existence of FlexrayTpConnection.directTpSdu

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to IPdu in the role directTpSdu shall exist.⌋

[constr_9233] Existence of FlexrayTpConnection.receiver

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpNode in the role receiver shall exist at least once.⌋

[constr_9234] Existence of FlexrayTpConnection.tpConnectionControl

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpConnection-Control in the role tpConnectionControl shall exist.⌋

[constr_9235] Existence of FlexrayTpConnection.transmitter

Imposition time: CP: IT_SysDesc

⌈For each FlexrayTpConnection, the reference to FlexrayTpNode in the role transmitter shall exist.⌋

The FlexrayTpConnection refers to the FlexrayTpPduPool in two roles: rxPduPool and txPduPool.

[TPS_SYST_01064] Transmit/Receive Semantics of Pdu Pools ⌈The transmit/receive semantics of Pdu Pools depends on the role of the regarded ECU:

- If the ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus.
- If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.

⌋

The following example shows how this differentiation may be used:

System Description:
SENDER = A
RECEIVER = B
TxPool = PDU_1
RxPool = PDU_2

ECU Extract of A:
SENDER = A
TxPool = PDU_1 -> sent Pdus
RxPool = PDU_2 -> received Pdus

Since on receiver side the PDU_1 is received and PDU_2 is sent (from a local point of view) the export shall look like this:
ECU Extract of B:
RECEIVER = B
TxPool = PDU_1 -> received Pdus
RxPool = PDU_2 -> sent Pdus

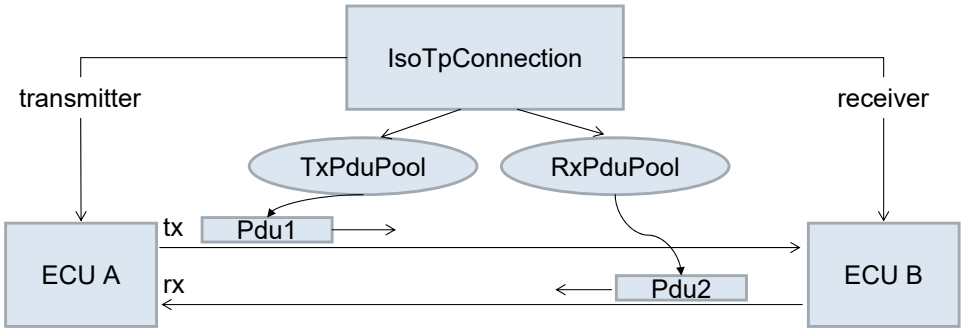

Figure 6.71: IsoTp Example

Class	FlexrayTpPduPool			
Note	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.pduPool			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nPdu	NPdu	*	ref	Reference to NPdus that are part of the PduPool. **Tags:** atp.Splitkey=nPdu This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.251: FlexrayTpPduPool

Class	FlexrayTpNode			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayTpConfig.tpNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
connector	Communication Connector	*	ref	Association to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). **Tags:** atp.Splitkey=connector
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection). **Tags:** atp.Splitkey=tpAddress

Table 6.252: FlexrayTpNode

Class	FlexrayTpEcu			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	FlexrayTpConfig.tpEcu			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
cycleTimeMain Function	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstance	EcuInstance	0..1	ref	Connection to the ECUInstance in the Topology **Tags:** atp.Splitkey=ecuInstance
fullDuplex Enabled	Boolean	0..1	attr	The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.

Table 6.253: FlexrayTpEcu

[constr_9236] Existence of `FlexrayTpEcu.ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayTpEcu`, the reference to `EcuInstance` in the role `ecuInstance` shall exist.⌋

[constr_9237] Existence of `FlexrayTpEcu.fullDuplexEnabled`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayTpEcu`, the attribute `fullDuplexEnabled` shall exist.⌋

[constr_5374] `IPdu` shall only be referenced once from a `FlexrayTpConnection` in the role `directTpSdu` or `reversedTpSdu` on a `FlexrayCluster`

Imposition time: CP: `IT_SysDesc`

⌈Each `IPdu` that is referenced in the role `directTpSdu` or `reversedTpSdu` from a `FlexrayTpConnection` that is aggregated by a `FlexrayTpConfig` that references a `FlexrayCluster` shall not be referenced a second time in the role `directTpSdu` or `reversedTpSdu` from any `FlexrayTpConnection` that is aggregated by a `FlexrayTpConfig` that references the same `FlexrayCluster`.⌋

6.8.3 CAN Transport Layer

The CAN Transport Layer supports multiple sessions by means of CanTpChannels: Each CanTpChannel uses its own resources, such as internal buffer, timer, state machine and thus can operate independently and simultaneously to other CanTpChannels. The same session can be reused for an arbitrary number of CanTpConnections.

Each CanTpConnection uses its own pair of NPdus: One NPdu, the dataPdu is mandatory for each CanTpConnection, the flowControlPdu is optional depending whether only Single Frames are transferred over the connection.

A CanTpConnection is specifically used for communication between source and target devices. These communication partners are specified using the transmitter and receiver associations to CanTpNode, providing the diagnostic tpAddress and the connection to the topology.

[TPS_SYST_01146] Generic CanTpConnections ⌈If the transmitter or the receiver of a CanTpConnection is not specified then the CanTpConnection is a generic one (address information is not determined).⌋

[TPS_SYST_01105] CanTpConnection with several receivers ⌈In case of several receivers a multicast tpAddress shall be used.⌋

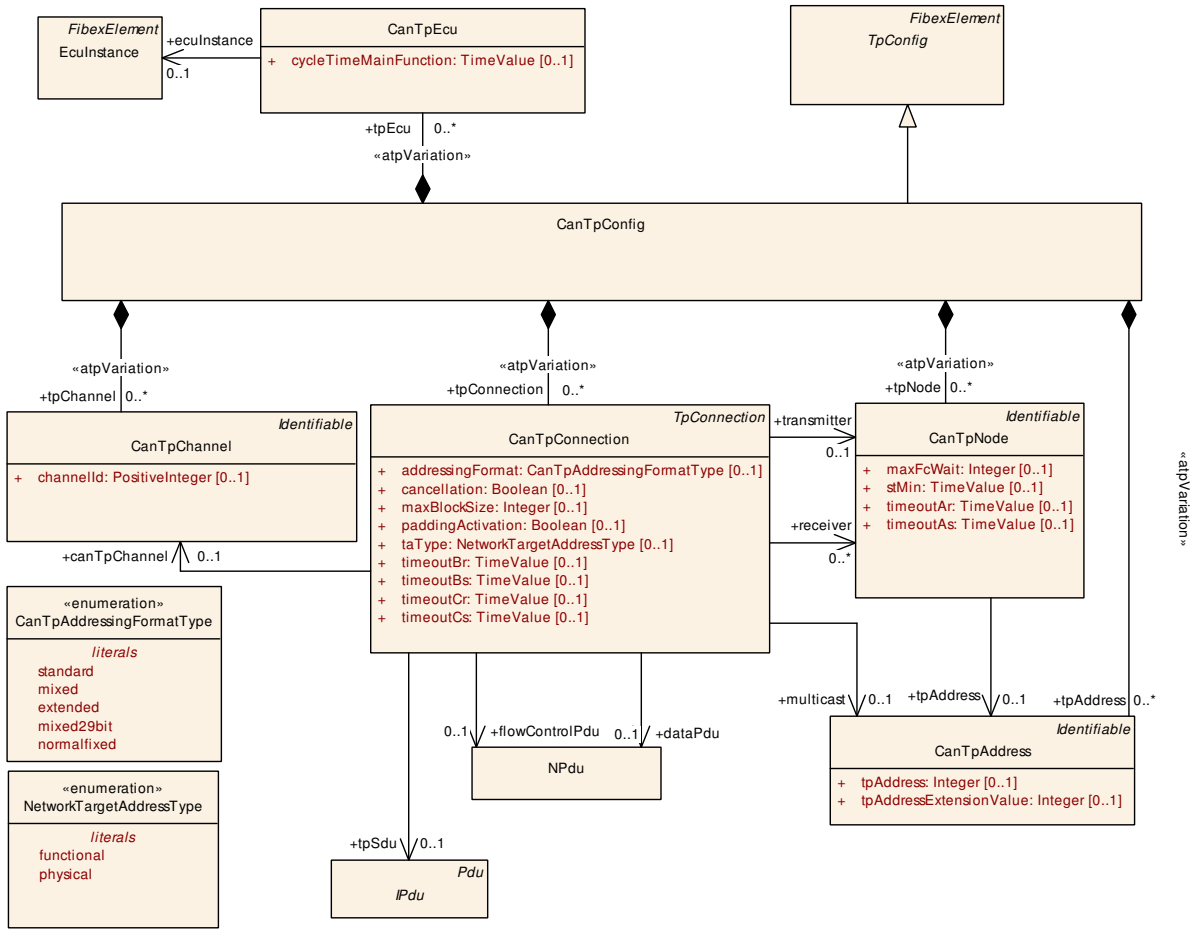

Figure 6.72: CAN Transport Layer Configuration (TransportProtocols: CanTransportProtocol)

The actual payload to be transported by the `CanTpConnection` is specified by the reference `tpSdu` to `IPdu`.

The `N_TAtype` communication models as defined in ISO 15765-2 [40] can be expressed using a combination of the attributes `addressingFormat` (`CanTpAddressingFormatType`) and `taType` (`NetworkTargetAddressType`).

Class	**CanTpConfig**			
Note	This element defines exactly one CAN TP Configuration. One CanTpConfig element shall be created for each CAN Network in the System. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	**CanTpConfig**			
tpAddress	CanTpAddress	*	aggr	Collection of TP Addresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpChannel	CanTpChannel	*	aggr	Configuration of CAN TP channels. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpChannel, tpChannel.shortName, tp Channel.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	CanTpConnection	*	aggr	Senders and receivers of CAN TP messages. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpEcu	CanTpEcu	*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpEcu, tpEcu.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	CanTpNode	*	aggr	Senders and receivers of Can TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.254: CanTpConfig

[constr_9247] Existence of CanTpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpAddress in the role tpAddress shall exist at least once.⌋

[constr_9248] Existence of CanTpConfig.tpChannel

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpChannel in the role tpChannel shall exist at least once.⌋

[constr_9249] Existence of CanTpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpConnection in the role tpConnection shall exist at least once.⌋

[constr_9250] Existence of CanTpConfig.tpEcu

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpEcu in the role tpEcu shall exist at least once.⌋

[constr_9251] Existence of CanTpConfig.tpNode

Imposition time: CP: IT_SysDesc

⌈For each CanTpConfig, the aggregation of CanTpNode in the role tpNode shall exist at least once.⌋

Class	**CanTpChannel**			
Note	Configuration parameters of the CanTp channel. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
channelId	PositiveInteger	0..1	attr	The id of the channel. The value shall be unique for each channel.

Table 6.255: CanTpChannel

Class	**CanTpConnection**			
Note	A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	CanTpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
addressing Format	CanTpAddressing FormatType	0..1	attr	Declares which communication addressing mode is supported.
cancellation	Boolean	0..1	attr	With this switch Tx Cancellation can be turned on or off. Please note that the Rx Cancellation is always enabled.
canTpChannel	CanTpChannel	0..1	ref	Reference to the CanTpChannel on which this CanTp Connection is realized. **Tags:** atp.Splitkey=canTpChannel
dataPdu	NPdu	0..1	ref	Reference to an Data NPdu. **Tags:** atp.Splitkey=dataPdu
flowControlPdu	NPdu	0..1	ref	Reference to the Flow Control NPdu. **Tags:** atp.Splitkey=flowControlPdu
maxBlockSize	Integer	0..1	attr	The maximum number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification. Note: For reasons of buffer length, the CAN Transport Layer can adapt the BS value within the limit of this maximum BS

▽

△

Class	**CanTpConnection**			
multicast	CanTpAddress	0..1	ref	TP address for 1:n connections. **Tags:** atp.Splitkey=multicast
padding Activation	Boolean	0..1	attr	This specifies whether or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload. true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes) false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)
receiver	CanTpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver
taType	NetworkTargetAddress Type	0..1	attr	Network Target Address type.
timeoutBr	TimeValue	0..1	attr	Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
timeoutCs	TimeValue	0..1	attr	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
tpSdu	IPdu	0..1	ref	Reference to an IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=tpSdu
transmitter	CanTpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter

Table 6.256: CanTpConnection

[constr_9252] Existence of `CanTpConnection.addressingFormat`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanTpConnection`, the attribute `addressingFormat` shall exist.⌋

[constr_9253] Existence of `CanTpConnection.canTpChannel`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanTpConnection`, the reference to `CanTpChannel` in the role `canTpChannel` shall exist.⌋

[constr_9254] Existence of `CanTpConnection.dataPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanTpConnection`, the reference to `NPdu` in the role `dataPdu` shall exist.⌋

[constr_9255] Existence of CanTpConnection.paddingActivation

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the attribute paddingActivation shall exist.⌋

[constr_9256] Existence of CanTpConnection.tpSdu

Imposition time: CP: IT_SysDesc

⌈For each CanTpConnection, the reference to IPdu in the role tpSdu shall exist.⌋

Enumeration	**CanTpAddressingFormatType**
Note	Declares which communication addressing mode is supported.
Aggregated by	CanTpConnection.addressingFormat
Literal	***Description***
extended	To use extended addressing format. **Tags:** atp.EnumerationLiteralIndex=0
mixed	To use mixed 11bit addressing format. **Tags:** atp.EnumerationLiteralIndex=1
mixed29bit	To use mixed 29bit addressing format **Tags:** atp.EnumerationLiteralIndex=2
normalfixed	To use normal fixed addressing format **Tags:** atp.EnumerationLiteralIndex=3
standard	To use normal addressing format. **Tags:** atp.EnumerationLiteralIndex=4

Table 6.257: CanTpAddressingFormatType

Class	**CanTpAddress**			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpAddress			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpAddress	Integer	0..1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.
tpAddress ExtensionValue	Integer	0..1	attr	If the mixed addressing format is used, this parameter contains the transport protocol address extension value.

Table 6.258: CanTpAddress

[constr_9257] Existence of CanTpAddress.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each CanTpAddress, the attribute tpAddress shall exist.⌋

Class	**CanTpEcu**			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	CanTpConfig.tpEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cycleTimeMain Function	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstance	EcuInstance	0..1	ref	Connection to the ECUInstance in the Topology **Tags:** atp.Splitkey=ecuInstance

Table 6.259: CanTpEcu

[constr_9258] Existence of `CanTpEcu.ecuInstance`

Imposition time: CP: `IT_SysDesc`

⌈For each `CanTpEcu`, the attribute `ecuInstance` shall exist.⌋

Class	**CanTpNode**			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CanTpConfig.tpNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
connector	Communication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). **Tags:** atp.Splitkey=connector
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.
stMin	TimeValue	0..1	attr	Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
tpAddress	CanTpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection). **Tags:** atp.Splitkey=tpAddress

Table 6.260: CanTpNode

Enumeration	**NetworkTargetAddressType**
Note	Network Target Address type (see ISO 15765-2).
Aggregated by	CanTpConnection.taType
Literal	***Description***
functional	Functional request type **Tags:** atp.EnumerationLiteralIndex=0
physical	Physical request type **Tags:** atp.EnumerationLiteralIndex=2

Table 6.261: NetworkTargetAddressType

[constr_5376] `IPdu` shall only be referenced once from a `CanTpConnection` in the role `tpSdu` on a `CanCluster`

Imposition time: CP: `IT_SysDesc`

⌈Each `IPdu` that is referenced in the role `tpSdu` from a `CanTpConnection` that is aggregated by a `CanTpConfig` that references a `CanCluster` shall not be referenced in the role `tpSdu` from a different `CanTpConnection` that is aggregated by a `CanTpConfig` that references the same `CanCluster`.⌋

6.8.4 LIN Transport Layer

LinTpConnection is used for modeling communication resources required for using the LIN Transport Layer. Contrary to the FlexRay and CAN Transport Layers, LIN TP only supports one session per PhysicalChannel.

An arbitrary number of LinTpConnections per LinTpConfig can be defined since the transmission of data from master to slave, using the MasterRequest frame, and the transmission of data from slave to master, using the SlaveResponse frame, needs to be described per NAD the LinMaster uses to address one or more of its LinSlaves.

LinTpConnection uses the dataPdu reference for specifying exactly one NPdu which is to be used for transmitting the data, and it optionally references a flowControl NPdu in order to handle Flow Control Frames if required.

One LinTpConnection is specifically used for communication between one source and one or several target devices. These communication partners are specified using the transmitter and receiver associations to LinTpNode, providing the diagnostic tpAddress and the connection to the topology. In case of several receivers a multicast tpAddress shall be used.

The actual payload to be transported by the LinTpConnection is specified by the reference linTpNSdu to IPdu.

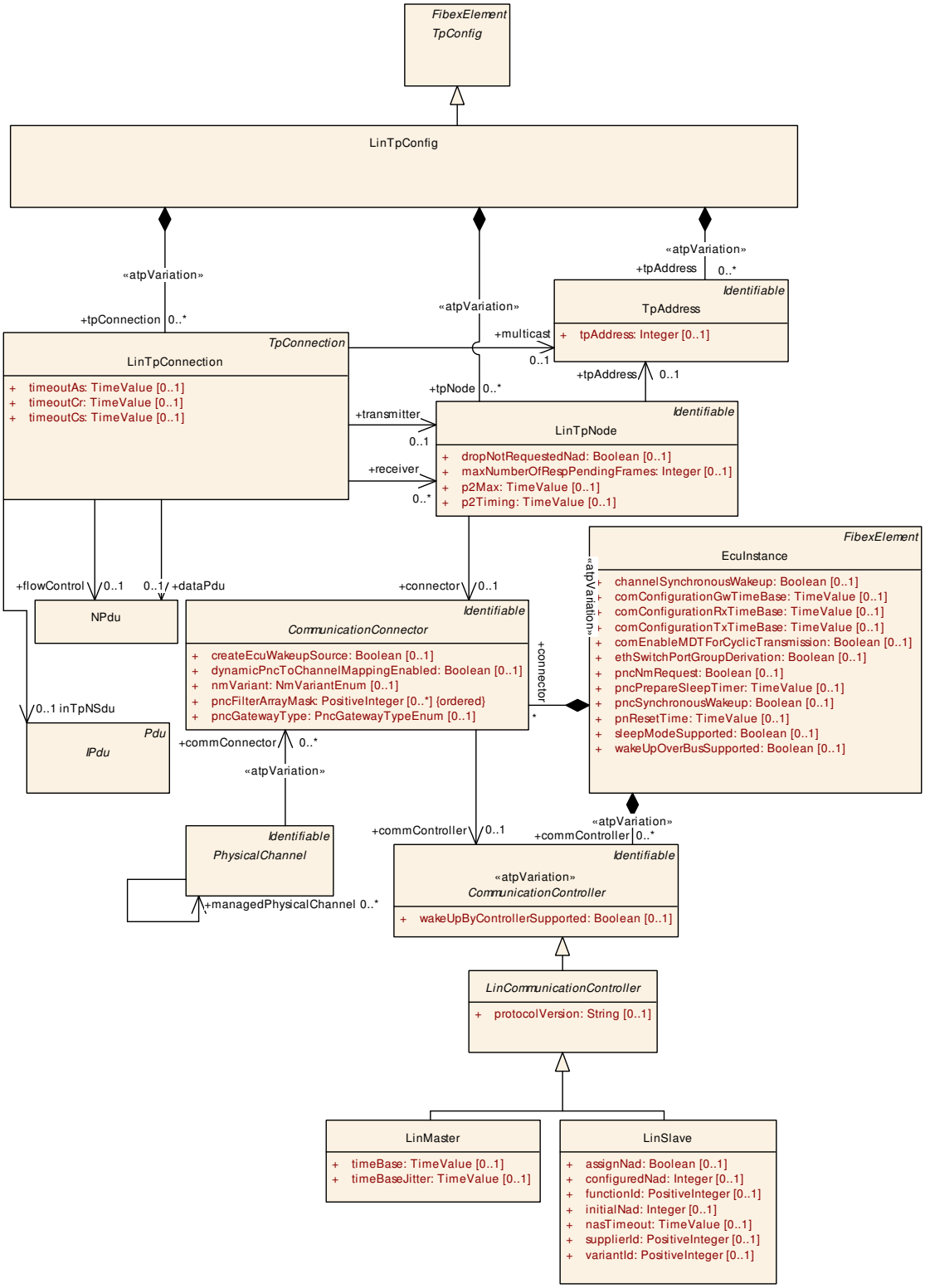

Figure 6.73: LIN Transport Layer Configuration

Class	LinTpConfig			
Note	This element defines exactly one Lin TP Configuration. One LinTpConfig element shall be created for each Lin Network in the System. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpAddress	TpAddress	*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	LinTpConnection	*	aggr	Configuration of LIN TP channels. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpNode	LinTpNode	*	aggr	Senders and receivers of LIN TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tp Node.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.262: LinTpConfig

[constr_9259] Existence of `LinTpConfig.tpAddress`

Imposition time: CP: `IT_SysDesc`

⌈For each `LinTpConfig`, at least one `TpAddress` shall be aggregated by `LinTpConfig` in the role `tpAddress`.⌋

Class	LinTpNode			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LinTpConfig.tpNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	LinTpNode			
connector	Communication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). **Tags:** atp.Splitkey=connector
dropNot RequestedNad	Boolean	0..1	attr	Configures if TP Frames of not requested LIN-Slaves are dropped or not.
maxNumberOf RespPending Frames	Integer	0..1	attr	Configures the maximum number of allowed response pending frames.
p2Max	TimeValue	0..1	attr	After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.
p2Timing	TimeValue	0..1	attr	P2 timeout observation parameter.
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection). **Tags:** atp.Splitkey=tpAddress

Table 6.263: LinTpNode

Class	LinTpConnection			
Note	A LinTP channel represents an internal path for the transmission or reception of a Pdu via LinTp and describes the sender and the receiver of this particular communication. LinTp supports (per Lin Cluster) the configuration of one Rx Tp-SDU and one Tx Tp-SDU per NAD the LinMaster uses to address one or more of its Lin Slaves. To support this an arbitrary number of LinTp Connections shall be described. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	LinTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataPdu	NPdu	0..1	ref	Reference to an NPdu (Single Frame, First Frame or Consecutive Frame). The Single Frame network protocol data unit (SF N_PDU) shall be sent out by the sending network entity and can be received by one or multiple receiving network entities. The Single Frame (SF N_PDU) shall be sent out to transfer a service data unit that can be transferred via a single service request to the data link layer. This network protocol data unit shall be sent to transfer unsegmented messages. The First Frame network protocol data unit (FF N_PDU) identifies the first network protocol data unit (N_PDU) of a segmented message transmitted by a network sending entity and received by a receiving network entity. The Consecutive Frame network protocol data unit (CF N_PDU) transfers segments (N_Data) of the service data unit message data (<MessageData>). All network protocol data units (N_PDUs) transmitted by the sending entity after the First Frame network protocol data unit (FF N_PDU) shall be encoded as Consecutive Frames network protocol data units (CF N_PDUs). **Tags:** atp.Splitkey=dataPdu

▽

△

Class	LinTpConnection			
flowControl	NPdu	0..1	ref	Reference to the Flow Control NPdu. The Flow Control network protocol data unit (FC N_PDU) is identified by the Flow Control protocol control information (FC N_PCI). The Flow Control network protocol data unit (FC N_PDU) instructs a sending network entity to start, stop or resume transmission of CF N_PDUs. The Flow Control network protocol data unit shall be sent by the receiving network layer entity to the sending network layer entity, when ready to receive more data, after correct reception of: a) First Frame network protocol data unit (FF N_PDU) b) the last Consecutive Frame network protocol data unit (CF N_PDU) of a block of Consecutive Frames (CF N_PDU) if further Consecutive Frame network protocol data unit (CF N_PDU) need(s) to be sent. **Tags:** atp.Splitkey=flowControl
linTpNSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. **Tags:** atp.Splitkey=linTpNSdu
multicast	TpAddress	0..1	ref	TP address for 1:n connections. **Tags:** atp.Splitkey=multicast
receiver	LinTpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver
timeoutAs	TimeValue	0..1	attr	Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This attribute defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
timeoutCs	TimeValue	0..1	attr	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
transmitter	LinTpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter

Table 6.264: LinTpConnection

[constr_9260] Existence of LinTpConnection.dataPdu

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, the reference to NPdu in the role dataPdu shall exist.⌋

[constr_9261] Existence of LinTpConnection.linTpNSdu

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, the reference to IPdu in the role linTpNSdu shall exist.⌋

[constr_9262] Existence of LinTpConnection.receiver

Imposition time: CP: IT_SysDesc

⌈For each LinTpConnection, at least one reference to LinTpNode in the role receiver shall exist.⌋

[constr_9263] Existence of `LinTpConnection.transmitter`

Imposition time: CP: `IT_SysDesc`

⌈For each `LinTpConnection`, the reference to `LinTpNode` in the role `transmitter` shall exist.⌋

[constr_5377] `IPdu` shall only be referenced once from a `LinTpConnection` in the role `linTpNSdu` on a `LinCluster`

Imposition time: CP: `IT_SysDesc`

⌈Each `IPdu` that is referenced in the role `linTpNSdu` from a `LinTpConnection` that is aggregated by a `LinTpConfig` that references a `LinCluster` shall not be referenced in the role `linTpNSdu` from a different `LinTpConnection` that is aggregated by a `LinTpConfig` that references the same `LinCluster`.⌋

6.8.5 Ethernet Transport Layer

The Transport Layer in the AUTOSAR Ethernet protocol stack is defined by the TcpIp module. For the transmission of an upper layer module Pdu via an UDP or TCP socket, the AUTOSAR Socket Adaptor specifies a Pdu route which is linked to a socket connection. The upper layer module of the SoAd may use the Interface (IF) API or the Transport Protocol (TP) API for the transmit request and data provision respectively.

With the `EthTpConnection` it is possible to describe in a System Description that the TP API shall be used for a specific Pdu route. If a `PduTriggering` is not referenced by a `EthTpConnection` the IF API will be used.

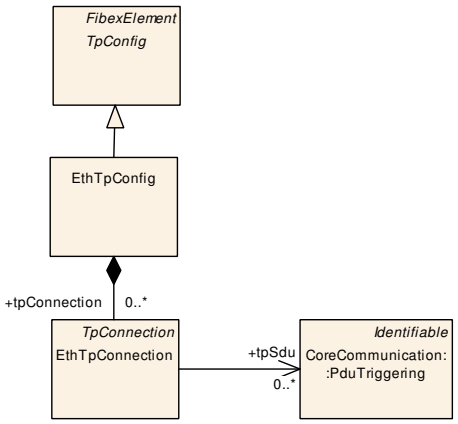

Figure 6.74: Modeling of EthTpConnection

Class	EthTpConfig			
Note	This element defines which PduTriggerings shall be handled using "TP" semantics. **Tags:** atp.recommendedPackage=TpConfigs			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpConnection	EthTpConnection	*	aggr	Senders and receivers of SOME/IP TP messages. **Tags:** atp.Splitkey=tpConnection, tp Connection.ident.shortName

Table 6.265: EthTpConfig

Class	EthTpConnection			
Note	A connection identifies which PduTriggerings shall be handled using the "TP" semantics.			
Base	*ARObject*, *TpConnection*			
Aggregated by	EthTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpSdu	PduTriggering	*	ref	Reference to a PduTriggering that shall be transported using the "TP" semantics. **Tags:** atp.Splitkey=tpSdu

Table 6.266: EthTpConnection

6.8.6 SOME/IP segmenter

On the transmission side SOME/IP TP segments an incoming SOME/IP `IPdu` that does not fit into a single UDP Package into smaller `GeneralPurposeIPdu`s with category SOMEIP_SEGMENTED_IPDU and allows to transport SOME/IP messages over UDP that are greater than 128KB. On the reception side the large `IPdu` is reassembled again. The Message Type field of the SOME/IP header contains a bit, which marks the SOME/IP message as a segment of an original SOME/IP message. Every segmented SOME/IP message adds SOME/IP TP specific fields to the SOME/IP header. These fields contain control information for the segmentation and the reassembly of original, large SOME/IP messages.

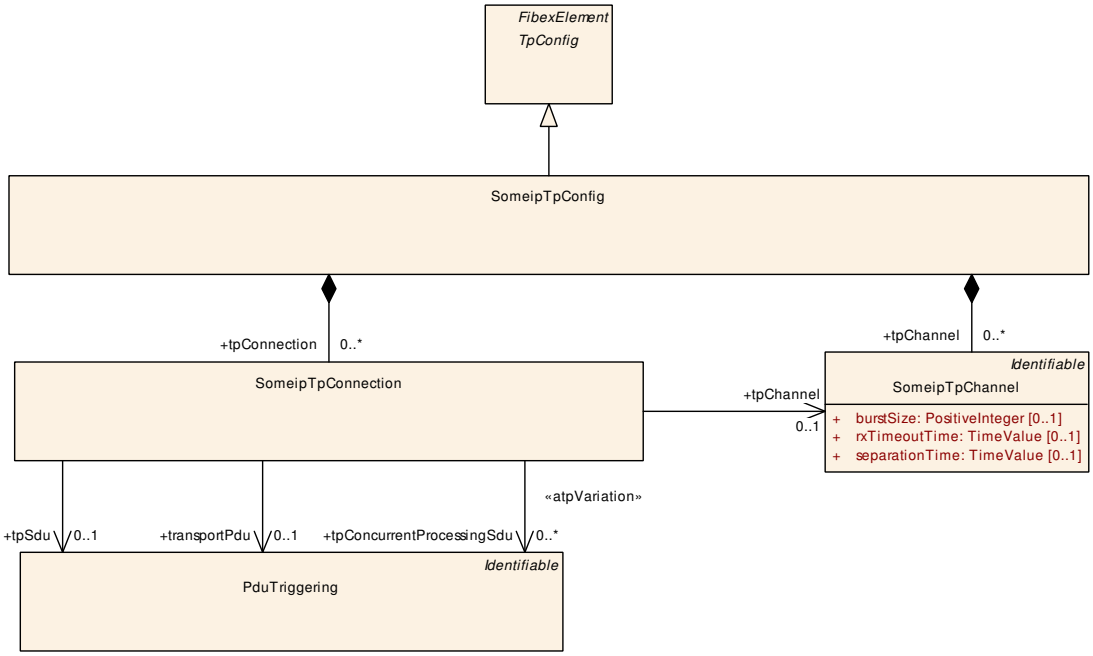

Figure 6.75: SOME/IP Segmenter

Class	**SomeipTpConfig**			
Note	This element defines exactly one SOME/IP TP Configuration. **Tags:** atp.recommendedPackage=TpConfigs			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpChannel	SomeipTpChannel	*	aggr	Definition of SomeipTpChannels that are collecting configuration properties that are valid for a collection of SomeipTpConnections. **Tags:** atp.Splitkey=tpChannel, tpChannel.shortName
tpConnection	SomeipTpConnection	*	aggr	Senders and receivers of SOME/IP TP messages. **Tags:** atp.Splitkey=tpConnection

Table 6.267: SomeipTpConfig

Class	SomeipTpConnection			
Note	A connection identifies the sender and the receiver of this particular communication. The SOME/IP TP module routes a Pdu through this connection.			
Base	*ARObject*			
Aggregated by	SomeipTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tpChannel	SomeipTpChannel	0..1	ref	Assignment of configuration properties valid for this SomeipTpConnection. **Tags:** atp.Splitkey=tpChannel
tpConcurrent ProcessingSdu	PduTriggering	*	ref	Reference to a set of IPdus that are used to implement concurrent processing of several upper layer IPdus related to one transportPdu by the SOME/IP TP segmenter. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConcurrentProcessingSdu, tpConcurrent ProcessingSdu.pduTriggering, tpConcurrentProcessing Sdu.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpSdu	PduTriggering	0..1	ref	Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tp ConcurrentProcessingSdu references shall be used. **Tags:** atp.Splitkey=tpSdu
transportPdu	PduTriggering	0..1	ref	Reference to the segmented IPdu. **Tags:** atp.Splitkey=transportPdu

Table 6.268: SomeipTpConnection

Class	SomeipTpChannel			
Note	This element is used to assign properties to SomeipTpConnections that are referencing this SomeipTp Channel.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SomeipTpConfig.tpChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
burstSize	PositiveInteger	0..1	attr	Specifies the number of segments that shall be transmitted in a burst ignoring separationTime. SeparationTime will then only be applied between bursts. If not configured, SeparationTime will be applied between all frames.
rxTimeoutTime	TimeValue	0..1	attr	Timer to monitor the successful reception. It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received.
separationTime	TimeValue	0..1	attr	Sets the duration of the minimum time in seconds the SOME/IP TP module shall wait between the transmissions of NPdus.

Table 6.269: SomeipTpChannel

[constr_3328] `SomeipTpConnection.transportPdu` reference restriction

Imposition time: CP: `IT_SysDesc`

⌈A `PduTriggering` that is referenced by a `SomeipTpConnection` in the role `transportPdu` shall reference a `GeneralPurposeIPdu` with category SOMEIP_SEGMENTED_IPDU in the role `iPdu`.⌋

[constr_3329] SomeipTpConnection.tpSdu reference restriction

Imposition time: CP: IT_SysDesc

⌈A PduTriggering that is referenced by a SomeipTpConnection in the role tpSdu shall reference an IPdu in the role iPdu.⌋

If the SomeipTpConnection.tpSdu reference is defined, then the SOME/IP TP segmenter supports the processing of exactly one upper layer IPdu.

An alternative is to define concurrent processing of upper layer IPdus. This processing use-case is supported by the possibility to define more than one SomeipTpConnection.tpConcurrentProcessingSdu reference. The SOME/IP TP segmenter is then able to perform concurrent transmission segmentation and reception assembly of as many concurrent sessions as there are tpConcurrentProcessingSdu references defined.

One use-case for such a concurrent processing of upper layer IPdus is described in figure 5.27.

[TPS_SYST_02442] Concurrent processing of upper layer IPdus by the SOME/IP TP segmenter ⌈If a SomeipTpConnection references PduTriggerings in the role tpConcurrentProcessingSdu, then the SOME/IP TP segmenter is enabled to process the referenced PduTriggerings (and the corresponding upper layer IPdus) concurrently.⌋

[constr_9396] SomeipTpConnection.tpConcurrentProcessingSdu reference restriction

Imposition time: IT_SysDesc

⌈A PduTriggering that is referenced by a SomeipTpConnection in the role tpConcurrentProcessingSdu shall reference an IPdu in the role iPdu.⌋

[constr_9397] Exclusive usage of tpSdu or tpConcurrentProcessingSdu

Imposition time: IT_SysDesc

⌈A SomeipTpConnection shall either define one tpSdu reference or a set of tpConcurrentProcessingSdu references, but not both.⌋

[TPS_SYST_02156] Length of GeneralPurposeIPdu with category SOMEIP_SEGMENTED_IPDU ⌈The length of GeneralPurposeIPdu with category SOMEIP_SEGMENTED_IPDU that is referenced by a PduTriggering in the role iPdu that in turn is referenced by a SomeipTpConnection in the role transportPdu defines the maximum size in bytes of a segment.⌋

Please note that the length of a GeneralPurposeIPdu with category SOMEIP_SEGMENTED_IPDU covers 8 bytes of the SOME/IP header, 4 bytes of the TP header, and the segment itself.

[constr_3330] Same `transportPdu` shall not be used in different `SomeipTpConnection`s

Imposition time: CP: `IT_SysDesc`

⌈A `PduTriggering` that is referencing a `GeneralPurposeIPdu` with category SOMEIP_SEGMENTED_IPDU in the role `iPdu` shall be referenced at most once by a `SomeipTpConnection` in the role `transportPdu`.⌋

[constr_5378] `PduTriggering` shall only be referenced once from a `SomeipTpConnection` in the role `tpSdu` or `tpConcurrentProcessingSdu`

Imposition time: CP: `IT_SysDesc`

⌈Each `PduTriggering` that is referenced in the role `tpSdu` or `tpConcurrentProcessingSdu` from a `SomeipTpConnection` shall not be referenced in the role `tpSdu` or `tpConcurrentProcessingSdu` from a different `SomeipTpConnection`.⌋

6.8.7 SAE J1939 Transport Layer

There are two transport protocol variants defined by J1939: BAM (Broadcast Announce Message), which is a broadcast protocol that does not use any flow control, and CMDT (Connection Mode Data Transfer), which is a point-to-point protocol with flow control and acknowledgment.

BAM uses two `NPdu`s for transport, TP.CM (Transport Protocol Command, `flowControlPdu`) and TP.DT (Transport Protocol Data, `dataPdu`). CMDT uses three `NPdu`s, because an additional TP.CM (`flowControlPdu`) in reverse direction is needed for flow control. The length of TP.CM and TP.DT `NPdu`s is fixed to 8 bytes.

[TPS_SYST_01106] Usage of additional `directPdu` in case of variable length `sdu`
⌈In case of variable length `sdu` (with system signals of variable length) an additional `directPdu` is possible:

- it is used if the current length of this `sdu` is up to 8 bytes.
- if the current length of this `sdu` is higher than 8 bytes the `sdu` will be transported via the `dataPdu`.

⌋

[TPS_SYST_01161] `J1939TpConnection.receiver` or `J1939TpConnection.receiver.tpAddress` in case `tpProtocolType` is `bam` or `fd_bam` ⌈If `J1939TpConnection.tpProtocolType` is `bam` or `fd_bam`, the `J1939TpConnection.receiver` or `J1939TpConnection.receiver.tpAddress` can be omitted.⌋

[constr_9408] J1939TpConnection.tpPg in case J1939TpConnection.tpProtocolType is fpp

Imposition time: CP: IT_SysDesc

⌈If J1939TpConnection.tpProtocolType is fpp, only one J1939TpConnection.tpPg shall be configured.⌋

[TPS_SYST_01159] J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress in case tpProtocolType is fpp ⌈If J1939TpConnection.tpProtocolType is fpp and J1939TpConnection.tpPg.pgn indicates PDU2 format (i.e. (pgn & 0x0ff00) >= 0x0f000), J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress can be omitted.⌋

[TPS_SYST_01160] Broadcast transmission in case tpProtocolType is fpp and J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress is omitted ⌈If J1939TpConnection.tpProtocolType is fpp and J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress is omitted, then broadcast transmission is implied.⌋

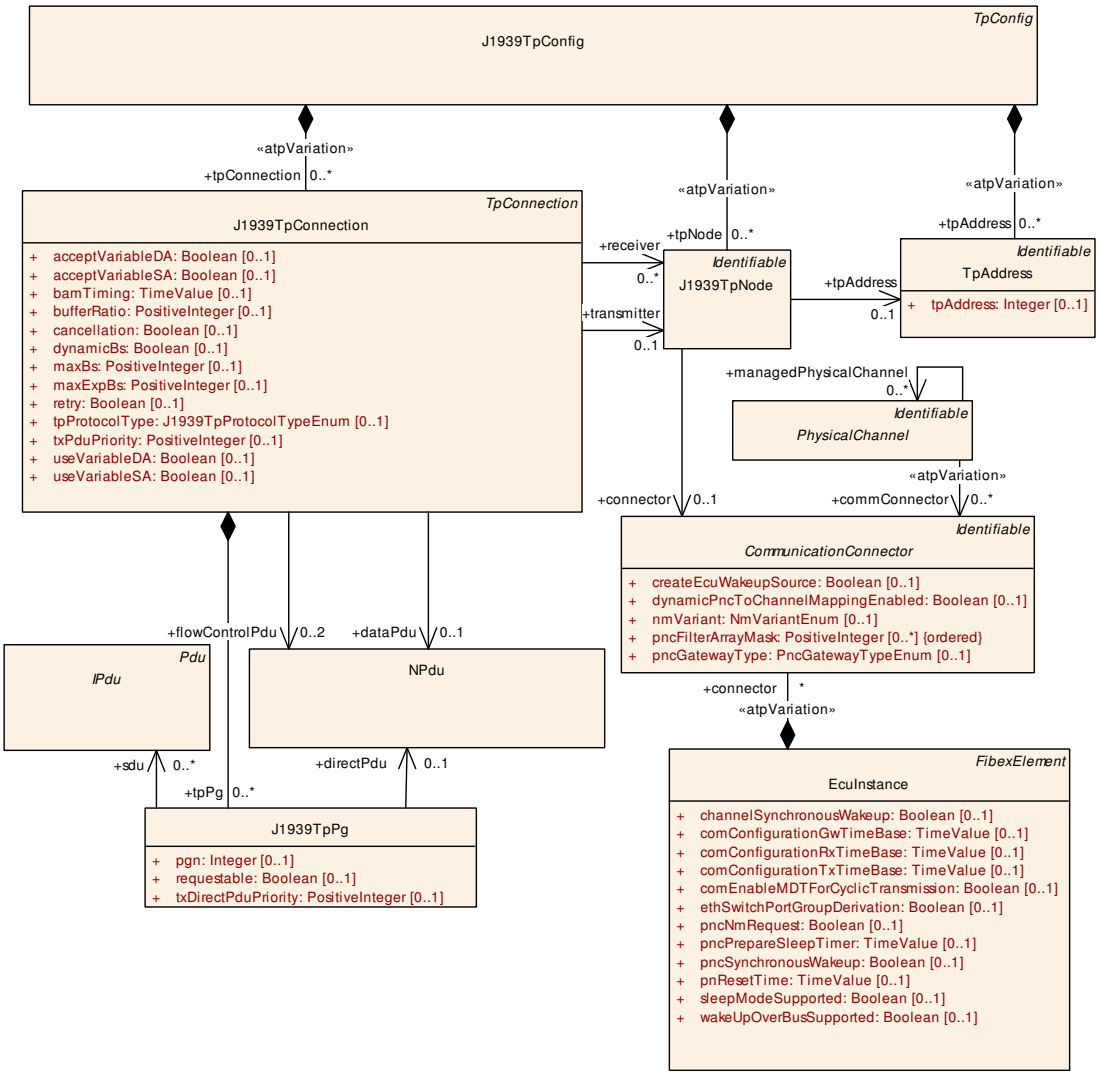

Figure 6.76: J1939 Transport Layer Configuration

A `J1939TpConnection` is specifically used for communication between source and target devices. These communication partners are specified using the `transmitter` and `receiver` associations to `J1939TpNode`, providing the diagnostic `tpAddress` and the connection to the topology.

[TPS_SYST_02190] **`J1939TpConnection.transmitter` reference in case of broadcast connection** ⌈In case of a broadcast connection the `J1939TpConnection` shall only reference the `J1939TpNode` in the role `transmitter`. The reason is that BAM (Broadcast Announce Message) is always directed at the target address 0xff and therefore no `receiver` reference is necessary.⌋

[TPS_SYST_02191] **`J1939TpConnection.transmitter` reference in case that the source is an unknown node** ⌈In case that the source is an unknown node, e.g. an arbitrary tester, the `J1939TpConnection` is allowed to omit the `transmitter` reference to `J1939TpNode`.⌋

[TPS_SYST_02192] **J1939TpConnection.receiver reference in case that the destination is an unknown node** ⌈In case that the destination is an unknown node, e.g. an arbitrary tester, the J1939TpConnection is allowed to omit the receiver reference to J1939TpNode.⌋

[TPS_SYST_02193] **J1939TpConnection.receiver reference in case that the destination is connected to a configured J1939NmNode** ⌈In case that the destination is connected to a configured J1939NmNode, the J1939TpConnection shall reference the J1939TpNode in the role receiver. It means that the receiving J1939TpNode is associated with an EcuInstance via the CommunicationConnector and this EcuInstance is associated with a J1939NmNode via the NmEcu. In this case the nmNodeId of the J1939NmNode corresponds to the TpAddress defined by J1939TpNode.⌋

The Parameter Group (PG) to be transported by the J1939TpConnection is specified by the tpPg aggregation.

[TPS_SYST_01147] **Generic J1939TpConnections** ⌈If the transmitter or the receiver of a J1939TpConnection is not specified then the J1939TpConnection is a generic one (address information is not determined).⌋

Class	**J1939TpConfig**			
Note	This element defines exactly one J1939 TP Configuration. One J1939TpConfig element shall be created for each J1939 Network in the System. **Tags:** atp.recommendedPackage=TpConfigs This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpAddress	TpAddress	*	aggr	Collection of TP Adresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpAddress, tpAddress.shortName, tp Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tpConnection	J1939TpConnection	*	aggr	Configuration of J1939 TP connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ident.short Name, tpConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	J1939TpConfig			
tpNode	J1939TpNode	*	aggr	Senders and receivers of J1939 TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpNode, tpNode.shortName, tpNode.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.270: J1939TpConfig

[constr_9264] Existence of J1939TpConfig.tpAddress

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one TpAddress shall be aggregated in the role tpAddress.⌋

[constr_9265] Existence of J1939TpConfig.tpConnection

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one J1939TpConnection shall be aggregated in the role tpConnection.⌋

[constr_9266] Existence of J1939TpConfig.tpNode

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConfig, at least one J1939TpNode shall be aggregated in the role tpNode.⌋

[TPS_SYST_02407] MetaData support is required if J1939TpConnection.acceptVariableSA is used ⌈The usage of J1939TpConnection.acceptVariableSA requires the support of COM Stack MetaData with MetaDataType = SOURCE_ADDRESS_16. Please note that this affects only J1939TpConnection.tpPg.sdus that in turn are referenced by a PduTriggering that references an IPduPort with communicationDirection = in (in other words Rx Pdus).⌋

[TPS_SYST_02408] MetaData support is required if J1939TpConnection.acceptVariableDA is used ⌈The usage of J1939TpConnection.acceptVariableDA requires the support of COM Stack MetaData with MetaDataType = TARGET_ADDRESS_16. Please note that this affects only J1939TpConnection.tpPg.sdus that in turn are referenced by a PduTriggering that references an IPduPort with communicationDirection = in (in other words Rx Pdus).⌋

[TPS_SYST_02409] MetaData support is required if J1939TpConnection.useVariableSA is used ⌈The usage of J1939TpConnection.useVariableSA requires the support of COM Stack MetaData with MetaDataType = SOURCE_ADDRESS_16. Please note that this affects J1939TpConnection.tpPg.sdus that in turn are referenced by a PduTriggering that

references an IPduPort with communicationDirection = out (in other words Tx Pdus).⌋

[TPS_SYST_02410] MetaData support is required if J1939TpConnection.useVariableDA is used ⌈The usage of J1939TpConnection.useVariableDA requires the support of COM Stack MetaData with MetaDataType = TARGET_ADDRESS_16. Please note that this affects only J1939TpConnection.tpPg.sdus that in turn are referenced by a PduTriggering that references an IPduPort with communicationDirection = out (in other words Tx Pdus).⌋

Class	**J1939TpConnection**			
Note	A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *TpConnection*			
Aggregated by	J1939TpConfig.tpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
acceptVariableDA	Boolean	0..1	attr	The TP message is accepted independently of the actually used destination address (DA). Otherwise, only the destination address configured as receiver.tpAddress is accepted. Only derived for the receiving ECU.
acceptVariableSA	Boolean	0..1	attr	The TP message is accepted independently of the actually used source address (SA). Otherwise, only the source address configured as transmitter.tpAddress is accepted. Only derived for the receiving ECU.
bamTiming	TimeValue	0..1	attr	Timing of TP.DT frames of a BAM transmission. Range 10ms to 50ms, if not configured 50ms are assumed. Relevant for tpProtocolTypes bam, bam_cmdt, fd_bam, and fd_bam_cmdt.
bufferRatio	PositiveInteger	0..1	attr	Defines usage of available data for dynamic block size calculation when protocol retry is enabled. This attribute describes in percent of available buffer that shall be used for retry.
cancellation	Boolean	0..1	attr	Enable support for Tx/Rx cancellation.
dataPdu	NPdu	0..1	ref	Data Message (TP.DT) used by CMDT and BAM. The DataNPdu has a fixed length of 8 bytes. **Tags:** atp.Splitkey=dataPdu
dynamicBs	Boolean	0..1	attr	Enable support for dynamic block size calculation.
flowControlPdu	NPdu	0..2	ref	Reference to the Command NPdus (TP.CM) that are used in the CMDT (Connection Mode Data Transfer) in both directions. BAM uses one TP.CM (Transport Protocol Command). The flowControlNPdu has a fixed length of 8 bytes. Please note that the role name "flowControlIPdu" is misleading and is kept for backward compatibilty reasons. **Tags:** atp.Splitkey=flowControlPdu
maxBs	PositiveInteger	0..1	attr	Set maximum block size (number of packets in TP.CM_CTS).

▽

△

Class	**J1939TpConnection**			
maxExpBs	PositiveInteger	0..1	attr	Set maximum for expected block size (maximum number of packets in TP.CM_RTS).
receiver	J1939TpNode	*	ref	The target of the TP connection. **Tags:** atp.Splitkey=receiver
retry	Boolean	0..1	attr	Enable support for protocol retry.
tpPg	J1939TpPg	*	aggr	J1939 messages (parameter groups, PGs) that can be transferred via this connection. **Tags:** atp.Splitkey=tpPg
tpProtocolType	J1939TpProtocolTypeEnum	0..1	attr	Protocol type used by the J1939TpConnection
transmitter	J1939TpNode	0..1	ref	The source of the TP connection. **Tags:** atp.Splitkey=transmitter
txPduPriority	PositiveInteger	0..1	attr	If configured, TP.CM and TP.DT frames shall be sent with this priority instead of the default priority 7. Range 0 to 7.
useVariableDA	Boolean	0..1	attr	The TP message is sent with variable destination address (DA). Otherwise, the destination address configured as receiver.tpAddress is always used. Only derived for the transmitting ECU.
useVariableSA	Boolean	0..1	attr	The TP message is sent with variable source address (SA). Otherwise, the source address configured as transmitter.tpAddress is always used. Only derived for the transmitting ECU.

Table 6.271: J1939TpConnection

Enumeration	**J1939TpProtocolTypeEnum**
Note	Protocol types that may be used by the J1939TpConnection
Aggregated by	J1939TpConnection.tpProtocolType
Literal	*Description*
bam	Use BAM protocol variant of J1939-21 exclusively. **Tags:** atp.EnumerationLiteralIndex=0
bam_cmdt	Use BAM or CMDT, depending on Meta Data for the destination address. **Tags:** atp.EnumerationLiteralIndex=2
cmdt	Use CMDT protocol variant of J1939-21 (aka RTS/CTS) exclusively. **Tags:** atp.EnumerationLiteralIndex=1
etp	Use ETP protocol of ISO 11783-6. **Tags:** atp.EnumerationLiteralIndex=6
fd_bam	Use FD BAM protocol variant of J1939-22 exclusively. **Tags:** atp.EnumerationLiteralIndex=3
fd_bam_cmdt	Use FD BAM or FD CMDT, depending on Meta Data for the destination address. **Tags:** atp.EnumerationLiteralIndex=5
fd_cmdt	Use FD CMDT protocol variant of J1939-22 (aka FD RTS/CTS) exclusively. **Tags:** atp.EnumerationLiteralIndex=4
fpp	Use fast packet protocol of NMEA 2000. **Tags:** atp.EnumerationLiteralIndex=7

Table 6.272: J1939TpProtocolTypeEnum

[constr_9332] Existence of J1939TpConnection.tpProtocolType

Imposition time: CP: IT_SysDesc

⌈For each J1939TpConnection, the attribute tpProtocolType shall exist.⌋

Class	J1939TpPg			
Note	A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	J1939TpConnection.tpPg			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
directPdu	NPdu	0..1	ref	In case of variable length IPdus (with system signals of variable length), an additional NPdu (with the PGN in the CAN ID) is used for messages with up to 8 bytes. **Tags:** atp.Splitkey=directPdu
pgn	Integer	0..1	attr	Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a Can FrameTriggering with an identifier.
requestable	Boolean	0..1	attr	Parameter Group can be triggered by the J1939 request message.
sdu	IPdu	*	ref	Reference to IPdus that are segmented by the Transport Protocol. If more than one IPdu is referenced, the IPdus are used when the same PGN is received in parallel via different transport protocols (BAM, CMDT, direct) on the same J1939TpConnection. **Tags:** atp.Splitkey=sdu
txDirectPdu Priority	PositiveInteger	0..1	attr	If configured, direct frames shall be sent with this priority instead of the priority configured in the CAN ID.

Table 6.273: J1939TpPg

Class	J1939TpNode			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	J1939TpConfig.tpNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
connector	Communication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). **Tags:** atp.Splitkey=connector
j1939Node	J1939Node	0..1	ref	Reference to J1939Node that has a defined J1939Node Name and is assocated with a J1939Controller Application. **Tags:** atp.Splitkey=j1939Node

▽

△

Class	**J1939TpNode**			
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional only when no TP is sent and only BAM is received. **Tags:** atp.Splitkey=tpAddress

Table 6.274: J1939TpNode

[constr_3210] `J1939TpPg`s with identical `pgn` value

Imposition time: CP: `IT_SysDesc`

⌈For all `J1939TpPg`s where the attribute `pgn` has an identical value the attribute `requestable` shall also have an identical value.⌋

[constr_5379] `IPdu` shall only be referenced once from a `J1939TpPg` in the role `sdu` on a `J1939Cluster`

Imposition time: CP: `IT_SysDesc`

⌈Each `IPdu` that is referenced in the role `sdu` from a `J1939TpPg` that is aggregated by a `J1939TpConfig` that references a `J1939Cluster` shall not be referenced in the role `sdu` from a different `J1939TpPg` that is aggregated by a `J1939TpConfig` that references the same `J1939Cluster`.⌋

6.8.7.1 Configuration of J1939Tp NPDUs

The network layer `NPdu`s used in J1939Tp are identified by their Parameter Group Numbers (PGNs). The modeling of the `CanFrameTriggering`s, `CanFrame`s, `PduToFrameMapping`s, `NPdu`s, and `PduTriggering`s related to J1939Tp `NPdu`s as described in the following paragraphs is optional as the information can be derived from the other elements in the model, e.g. the `J1939TpConnection`, `J1939TpPg` elements.

The PGN `0xEB00` identifies the "Data Transfer" (TP.DT) PDU, which carries the SDU payload both in "Broadcast Announce" (BAM), and in "Connection Mode Data Transfer" (CMDT) connections.

The PGN `0xEC00` is used for the TP.CM/BAM Flow-Control PDUs

- Broadcast Announce (BAM)
- Ready To Send (RTS)
- Clear To Send (CTS)
- Acknowledge (ACK)
- Abort

For the identification of sender and receiver nodes, J1939Tp is using one-byte node addresses. The value `0xFF` represents the "broadcast" receiver address.

J1939Tp NPdus have a priority value assigned, which lies in the range [0..7], where the value 0 represents the highest priority.

The priority, PGN, and sender and receiver addresses of a J1939Tp NPdus are encoded in the 29 bit CAN-Id as shown below:

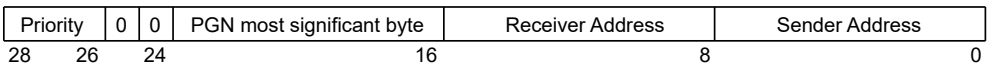

Figure 6.77: J1939TpCanIdentifier

The bits [24..25] identify the data page and extended data page bits. In the following, both are always assumed to be configured as false.

The 29 bits of the Can-Id are modeled in CanFrameTriggering.identifier. Parts of CanFrameTriggering.identifier can be configured as "determined at runtime". This allows a re-use of one and the same CanFrameTriggering for the transmission of multiple NPdus associated with different CAN-Ids at runtime. The configuration is done by setting the varying CAN-Id bits in CanFrameTriggering.rxMask and CanFrameTriggering.txMask to zero.

For instance, a CanFrameTriggering for sending TP.CM/BAM NPdus (PGN is 0xEC00) with a priority value of 0x7 and a sender, receiver address pair of 0x12, 0xFF is modeled as follows:

Parameter	Value
CanFrameTriggering.identifier	0x1CECFF12
CanFrameTriggering.txMask	0x1FFFFFFF

The most significant three bits of CanFrameTriggering.identifier field provide the priority. The three least significant bytes of the identifier field, 0xEC, 0xFF, and 0x12 indicate the most significant byte of the 0xEC00 PGN, the broadcast destination address, and the sender address. The txMask field contains no zero bit which determines the whole CAN-Id at configuration time. A txMask field without any zero bit does not bear any information and can be left out.

For re-using the CanFrameTriggering for different sender node addresses and different priority values it can be defined as follows:

Parameter	Value
CanFrameTriggering.identifier	0x00ECFF00
CanFrameTriggering.txMask	0x03FFFF00

Here, the most significant three bits and the least significant byte of the txMask field are set to zero, indicating that these parts of the CAN-Id are determined at runtime. The corresponding bits in the identifier field are ignored at configuration time (but are by convention set to zero as well).

6.8.7.1.1 Broadcast Announce

A node requires two CanFrameTriggerings for sending TP SDUs via the BAM protocol, one for the TP.CM/BAM Flow-Control PDU, and a second one for the TP.DT PDU. Both CanFrameTriggerings refer to "OUT" FramePorts of the sender node.

As a sender node, a receiver also requires two CanFrameTriggerings, this time referring to "IN" FramePorts of the receiver node.

J1939Tp NPdus usually have their priority field set to 0x7. If multiple sender addresses were used at the sender node (or multiple remote sender addresses were accepted at the receiver node), the least significant byte of the CanFrameTriggerings can be tagged as configured at runtime. The diagram below shows two CanFrameTriggering objects, one for the "BAM" NPdu, and one for the "DT" NPdu. The priority, and destination address parameters are set to 0x7 and 0xFF at configuration time. The PGN of the "BAM" CanFrameTriggering is set to 0xEC00, the PGN of the "DT" CanFrameTriggering is set to 0xEB00. For the NPdus, the corresponding PduTriggering and IPduPort elements have been left out for brevity. In the diagram, the configured "BAM" and "DT" NPdus are used to transport the 0xFEEC PGN. The TP SDU is modeled by an ISignalIPdu with a length of 17 bytes. The ISignalIPdu is referenced by the J1939TpPg element via the sdu reference. Please note that the model allows to use any sub-type of IPdu for modeling TP SDUs, e.g. the type DcmIPdu can be used for TP SDUs that are processed by the diagnostic layer.

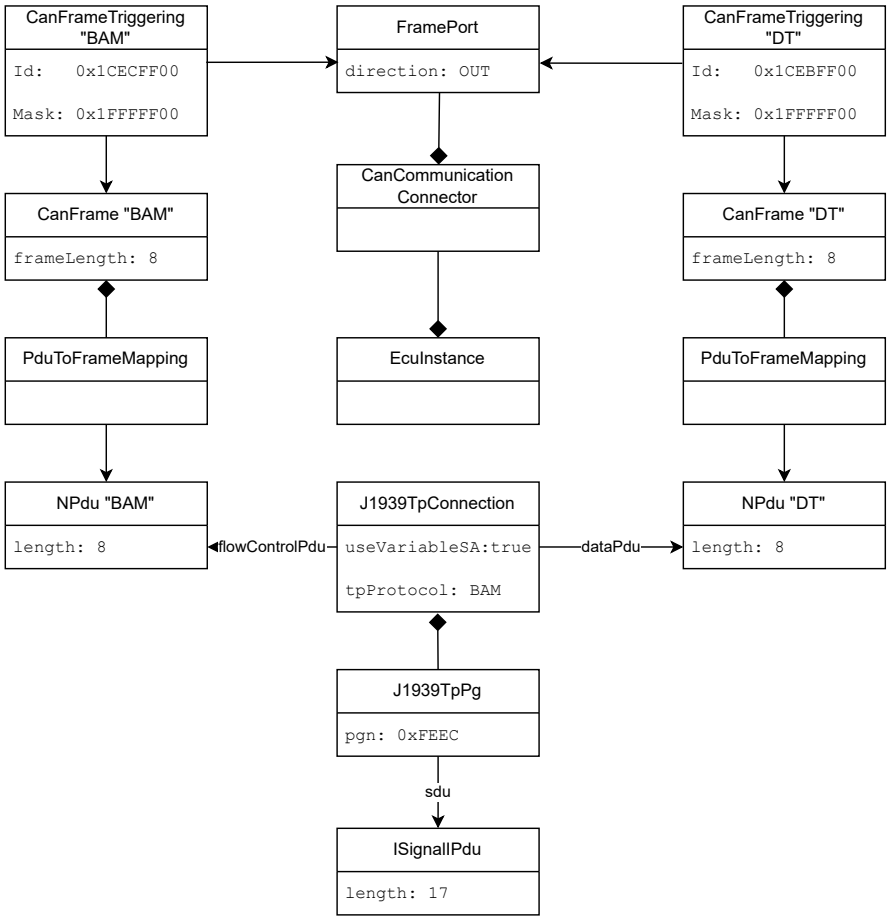

Figure 6.78: J1939TpCanFrameTriggering

For the receiver side, the configuration is mostly identical with the exceptions of the direction of the `FramePort` object, which would be "IN", and that, instead of `J1939TpConnection.useVariableSA`, `J1939TpConnection.acceptVariableSA` needs to be set to `true`.

If the `CanFrameTriggering` objects shall be used for both BAM and CMDT protocols, also bits `[8..15]` are tagged as configured at runtime, as these bits determine the destination address which has to have a value other than `0xFF` for CMDT. The diagram in the next section shows an example for that.

6.8.7.1.2 Connection Mode Data Transfer

In contrast to BAM, the CMDT connection requires `NPdu`s both in Rx and Tx direction, which is shown in the following diagram. The `NPdu` that is sent from the sender to the receiver node is used for "RTS", whereas the `NPdu` that goes into the opposite direction is used for "CTS", "ACK" and "Abort".

The CMDT mode allows to exchange SDUs between individual nodes (i.e. exactly one sender node and one receiver node are involved). In order to re-use NPdus for the communication with different remote nodes, the remote addresses must be determined at runtime. For a sent NPdu, the remote address is the receiver address, while for a received NPdu, the remote address is the sender address.

The diagram below shows the configuration of a "CMDT" J1939TpConnection from the perspective of the sending node with the TpAddress of 0x12, which is reflected in the least significant byte of the "CMDT_Tx" and "DT" CanFrameTriggering.identifier values and in bits [8..15] of the the "CMDT_Rx" CanFrameTriggering.identifier. Moreover, J1939TpConnection.useVariableSA is configured as false and J1939TpConnection.useVariableDA is configured as true which indicates that the sender address (i.e. the local address on transport layer level) is determined at configuration time while the receiver address (i.e. the remote address on transport layer level) is determined at runtime. In this example, the referenced SDU has a type of DcmIPdu, indicating that it is processed by the diagnostic layer.

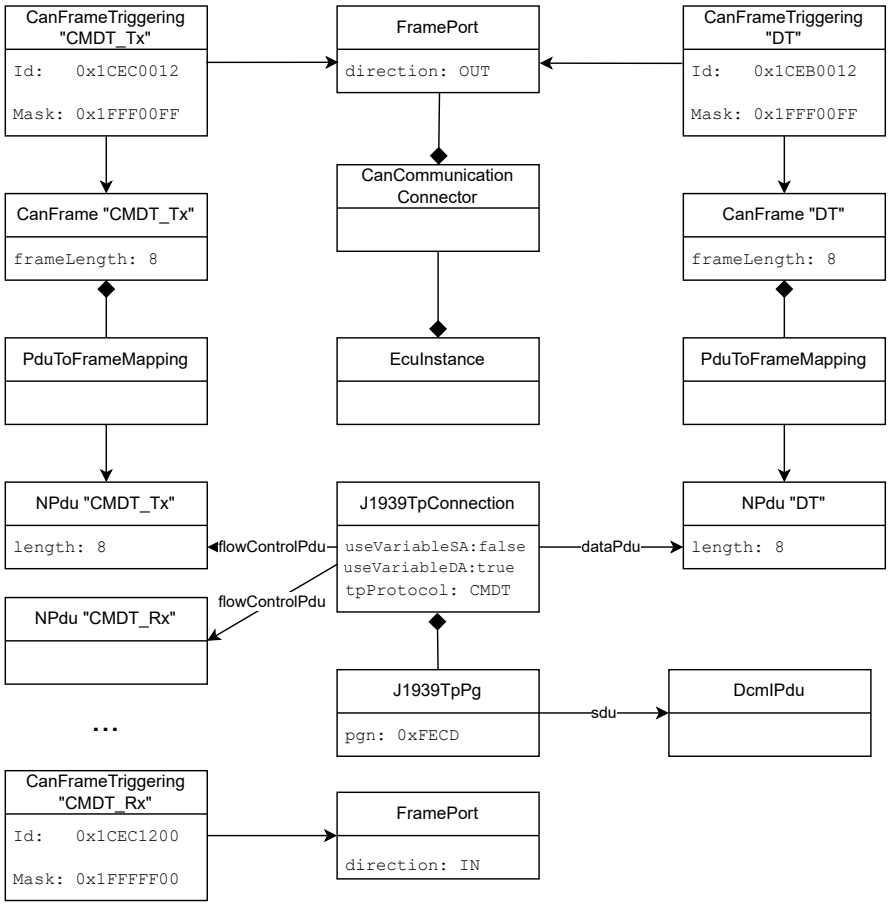

Figure 6.79: J1939TpCanFrameTriggeringCMDT

On the receiver side, the "DT" CanFrameTriggering would be received, and instead of the least significant byte, the bits [8..15] would be set to the TpAddress of the

receiving node. The "CMDT_Tx" and "CMDT_Rx" CanFrameTriggering elements would have the receiver node address configured instead of the sender address.

For the receiver node, J1939TpConnection.acceptVariableSA would be set to true and J1939TpConnection.acceptVariableDA would be set to false, indicating that the sender address (i.e. the remote address) is determined at runtime while the receiver address (i.e. the local address) is determined at configuration time.

6.8.7.1.3 Direct Transmission

J1939Tp supports the transmission of SDUs with variable length, i.e. the length of the SDU payload is determined by the sending node at runtime. If the runtime payload of such an SDU is eight bytes or less, that SDU payload can be transmitted directly, without the additional transmission of Flow-Control PDUs. The following diagram shows a CMDT example with a direct NPdu, the elements that were added or changed with respect to the previous CMDT example are drawn in a different shade.

The J1939TpPg element identified by its "PGN" 0xEF00 also provides a dedicated J1939TpPg.txDirectPduPriority value of 0, indicating that the SDU payload is sent with the highest possible priority when transmitted directly. In order to be able to re-use the sent "Direct" N-PDU for SDUs with different PGNs, for different receiver nodes, and with different priority values, the corresponding bits in the bit mask of CanFrameTriggering "Direct" are set to zero, while only the mask bits [24..25] and [0..7], are configured using fixed values. The bits [0..7] represent the local sender address which is configured as 0x12 in the identifier field. The referenced SDU of type ISignalIPdu indicates that it is processed by the application layer.

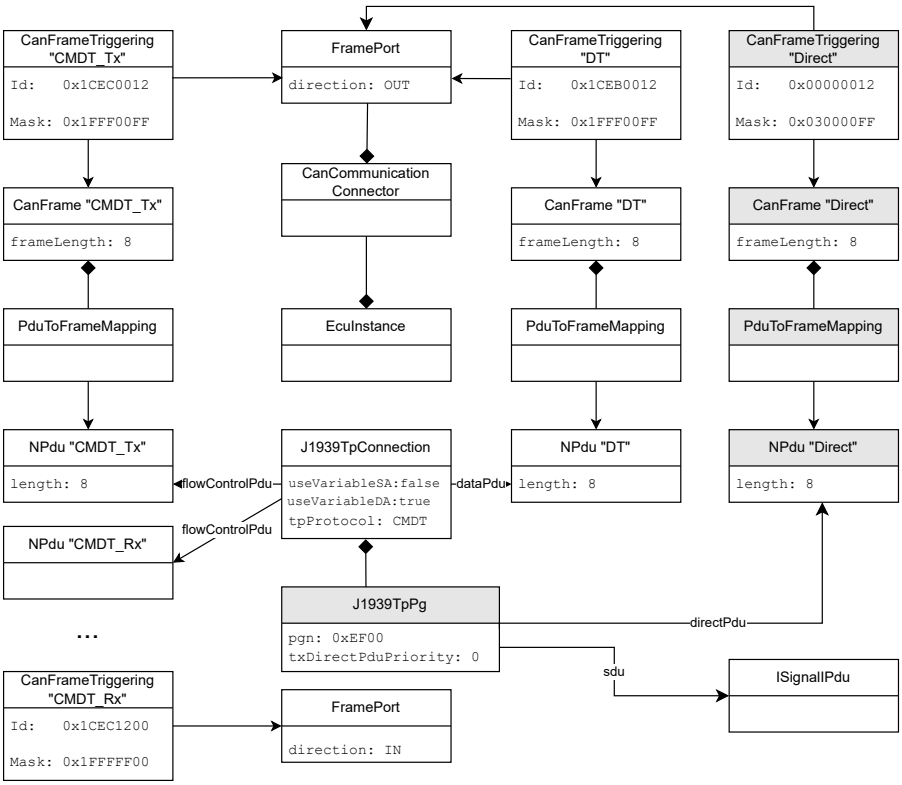

Figure 6.80: J1939TpCanFrameTriggeringCMDTDirect

For the receiver node, the `FramePort` and `IPduPort` objects would have `CommunicationDirectionType.in` configured as their `CommConnectorPort.communicationDirection`. Moreover, the parameter `J1939TpPg.txDirectPduPriority` is ignored, as the transmission priority is always controlled by the sender node.

6.8.8 Unicast TP Example

The example in Figure 6.81 illustrate the usage of the System Template TP model. In this System Description example the Sender ECU (Tester) communicates with the Receiver ECU (Diagnostic Server) via two Gateways (GW1 and GW2).

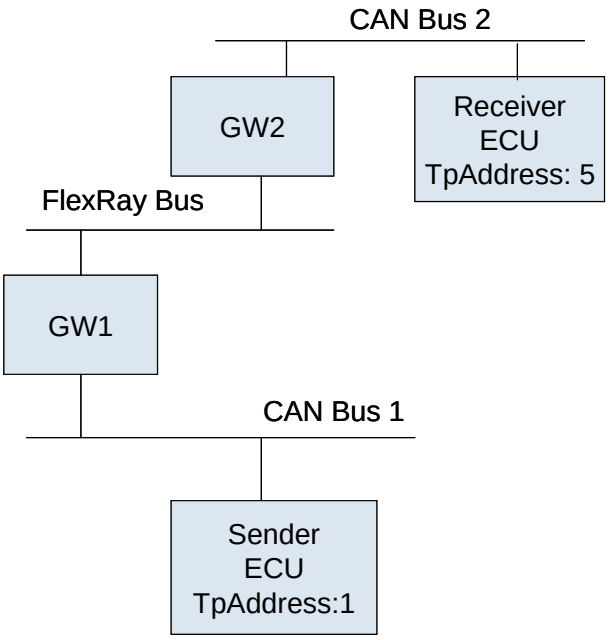

Figure 6.81: TP unicast Example

CAN Bus 1 (CanTpConfig 1):
CanTpConnection (CanTpConnection1)
transmitter TpNode: Sender ECU, TpAddress: 1
receiver TpNode: GW1, TpAddress: 5
CanTpConnection (CanTpConnection2):
transmitter TpNode: GW1, TpAddress: 5
receiver TpNode: Sender ECU, TpAddress: 1

FlexRay Bus (FlexRayTpConfig):
FlexRayTpConnection (FlexrayTpConnection1):
transmitter TpNode: GW1, TpAddress: 1
receiver TpNode: GW2, TpAddress: 5

CAN Bus 2 (CanTpConfig 2):
CanTpConnection (CanTpConnection3):
transmitter TpNode: GW2, TpAddress: 1
receiver TpNode: Receiver ECU, TpAddress: 5
CanTpConnection (CanTpConnection4):
transmitter TpNode: Receiver ECU, TpAddress: 5
receiver TpNode: GW2, TpAddress: 1

DiagnosticConnection:
physicalRequest TpConnection: CanTpConnection3
response TpConnection: CanTpConnection4

Please note that two different CanTpConfig elements are created for the two CAN networks. The TpAddress of the transmitter TpNode is always 1 and the TpAddress of the receiver TpNode is always 5, even in the FlexrayTpConfig where Gateway ECU1 communicates with Gateway ECU2. The original transmitter and the final receiver are addressed in each connection. Please note that for CanTp for each direction an own CanTpConnection is used.

The DiagnosticConnection is modeled only for the last segment to which the Receiver ECU that represents the diagnostic server is connected.

An additional Gateway modeling example is shown in the following Figure 6.82. In this example two CanTpConnections are used. One for the communication on Can1 CanCluster between Ecu1 and the Gw EcuInstance. And a second one for the communication on Can2 CanCluster between the Gw EcuInstance and Ecu2. Since each TpConfig is related to one CommunicationCluster two TpConfig elements are used in the example: one on Can1 and one on Can2. This also means that the Gw is represented by dedicated CanTpNode and CanTpAddress elements on Can1 and Can2, thereby allowing an individual configuration of CanTpNode and CanTpAddress parameters on each cluster.

The reference from the CanTpNode to the CanCommunicationConnector is not shown in Figure 6.82 for visibility reasons. This reference indicates the CommunicationController of the EcuInstance the CanTpConnection that sends or receives the Tp SDU. The communication direction on the CanTpConnection can be derived from the PduTriggering and the reference to the iPduPort.

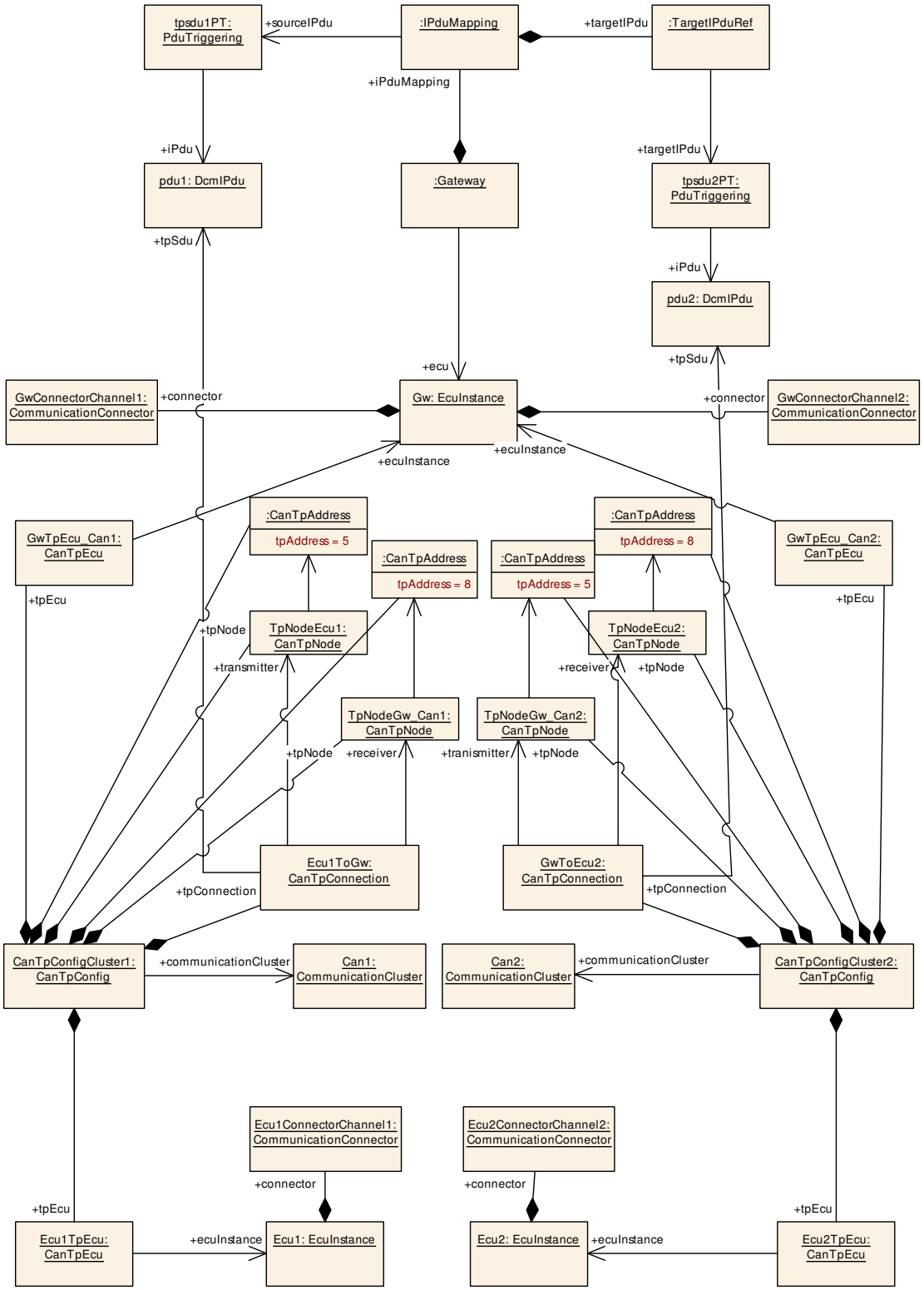

Figure 6.82: TpGateway Example

Please note that for the network only the `CanTpAddress`es of the end-nodes are relevant. The `CanTpAddress` of the Gw itself is not relevant. This is also illustrated in Figure 6.83.

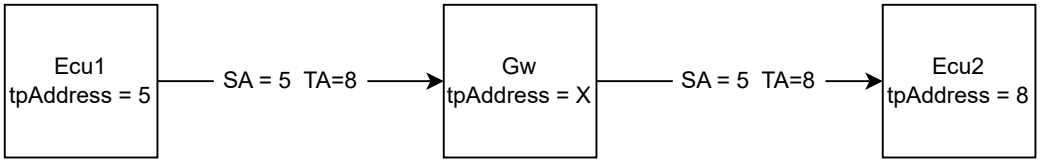

Figure 6.83: Modeling of Target Address in Gateway case

In the example in Figure 6.82 the SA of `CanTpNode` on Ecu1 is 5 and the TA of `CanTpNode` on Ecu2 is 8. The `CanTpConnection` creates always a local connection between 2 `CanTpNode`s. A "Proxy" of the end-node should be available in the configuration content of Ecu1 since the TA is relevant for the configuration. In the example the TpNodeGw_Can1 is such a "Proxy" since it contains the TA = 8. The Figure 6.84 shows how the ECU_SYSTEM_DESCRIPTION for Ecu1 may look like.

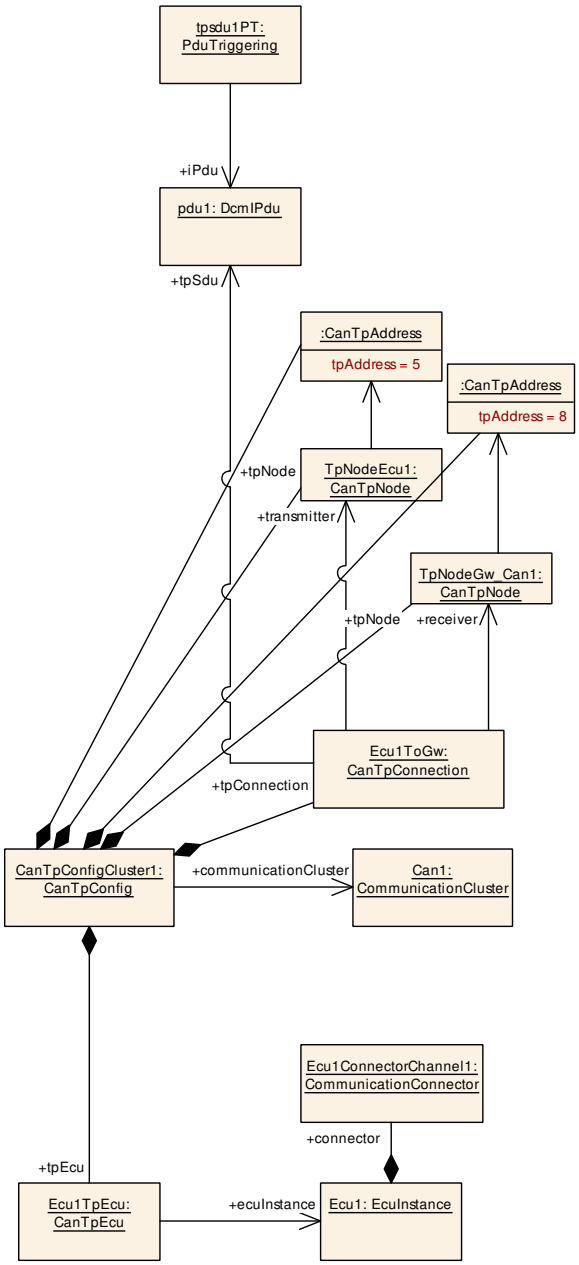

Figure 6.84: Example for ECU_SYSTEM_DESCRIPTION for Ecu1

6.8.9 Multicast TP Example

A second example illustrates the usage of the multicast reference.

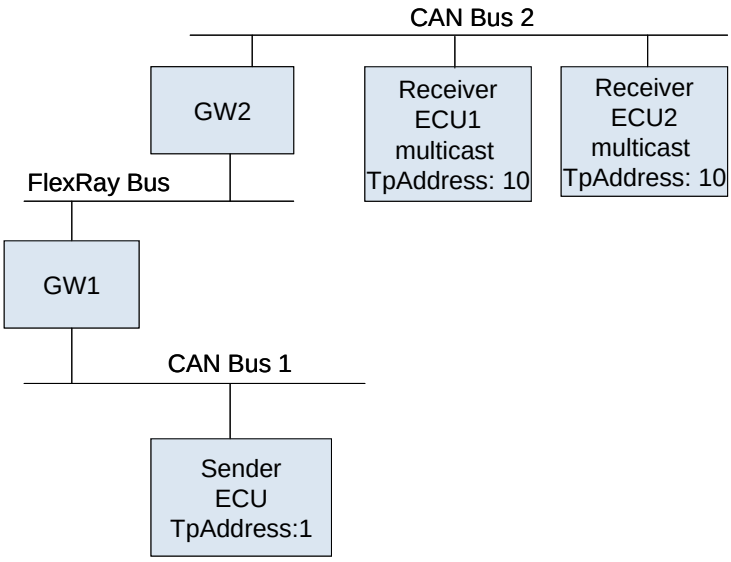

Figure 6.85: TP multicast Example

```
Can Bus 1 (CanTpConfig1):
CanTpConnection
  source TpNode: Sender ECU, TpAddress: 1
  target TpNode: GW1
  multicast TpAddress: 10

FlexRay Bus  (FlexRayTpConfig):
FlexRayTpConnection
  source TpNode: GW1, TpAddress: 1
  target TpNode: GW2
  multicast TpAddress: 10

CAN Bus 2 (CanTpConfig 2):
CanTpConnectionChannel
  source TpNode: GW2, TpAddress: 1
  target TpNode: Receiver ECU1
  target TpNode: Receiver ECU2
  multicast TpAddress: 10
```

Please note that the target `TpNode` does not contain a reference to the `TpAddress`. The multicast `TpAddress` is described by a direct reference from the connection.

6.8.10 Diagnostic Connection

A prominent use of the TP in automotive systems is the implementation of diagnostic communication. Data send from and to the tester frequently exceeds the native size of a communication package on typical bus systems used for this purpose.

However, the mere usage of TP channels for diagnostic purposes is missing one important aspect: TP channels, as defined by the AUTOSAR standard, are unidirectional by nature.

For diagnostic communication, it is very important to be able to define pairs of TP connections that can be taken to send related *request* and *response* messages.

In order to support this use case the meta-class `DiagnosticConnection` has been introduced.

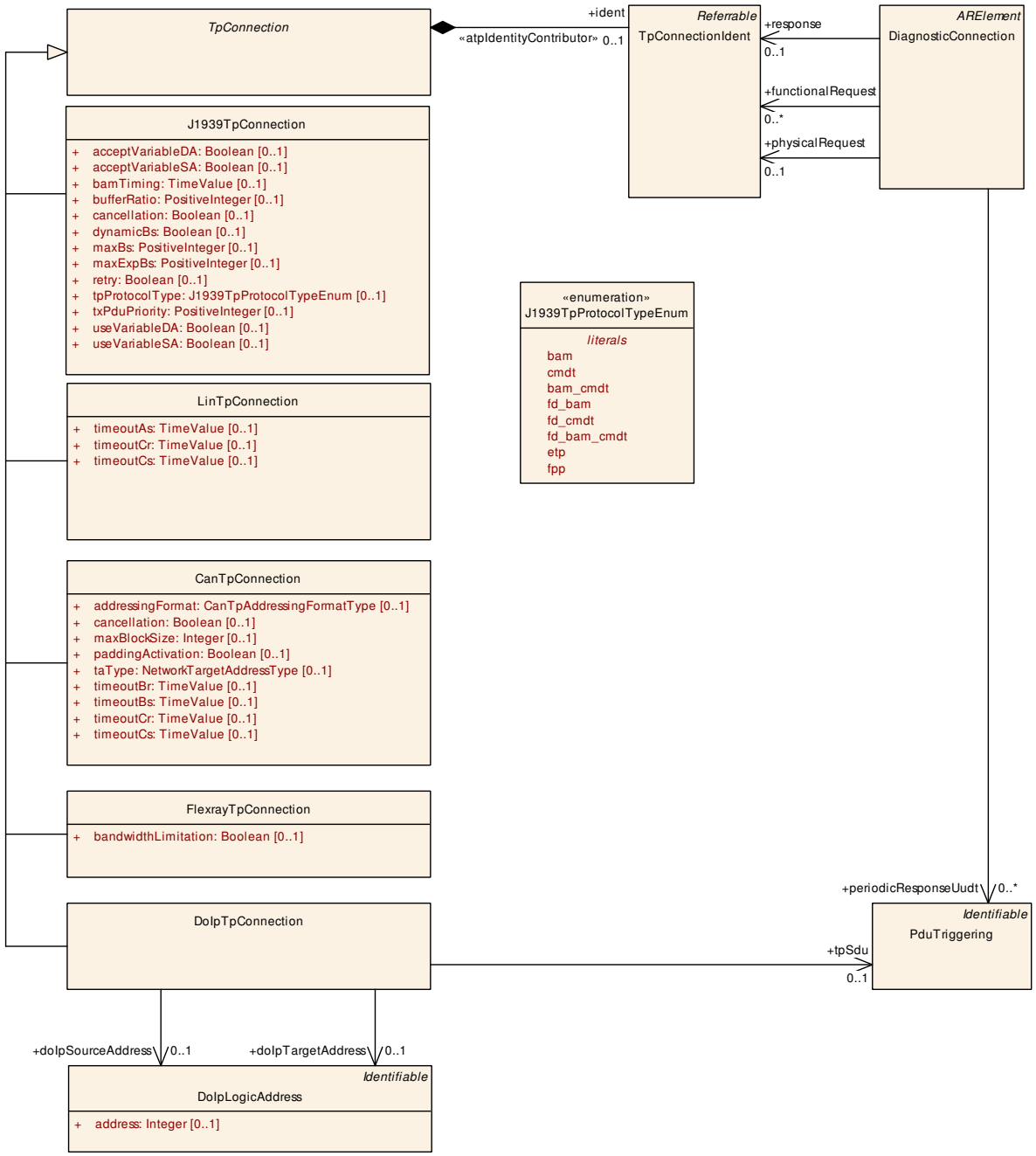

Figure 6.86: Modeling of `DiagnosticConnection`

[TPS_SYST_05003] Usage of `DiagnosticConnection` in combination with a TP

⌈`DiagnosticConnection` allows for the dedicated identification of TP connections used for the various diagnostic message sending use cases:

- `functionalRequest`
- `physicalRequest`
- `response`

⌋

[TPS_SYST_05004] Usage of DiagnosticConnection in combination with UUDT ⌈In addition to the usage of TP connections, the DiagnosticConnection foresees the transmission of UUDT message for **periodic response**. For this purpose, the role periodicResponseUudt is supported.⌋

[constr_1367] periodicResponseUudt.periodicResponseUudt shall only refer to a DcmIPdu

Imposition time: CP: IT_SysDesc

⌈If the role periodicResponseUudt exists then every PduTriggering referenced in the role periodicResponseUudt shall only refer to a DcmIPdu.⌋

Please note that the meta-class TpConnectionIdent (derived from Referrable) has been introduced for the purpose of allowing sub-classes of TpConnection to become the target of a reference while preserving full backwards-compatibility to the previous modeling.

This means in particular that the existence of a shortName is only required if the sub-class of TpConnection shall actually represent the target of a reference in the context of the definition of a DiagnosticConnection.

This, however, is kind of self-evident (because the reference would not work without the existence of a shortName at the reference target) and therefore it is not necessary to formulate an explicit constraint that clarifies this issue.

[constr_1368] Limitation of the target of references from DiagnosticConnection

Imposition time: CP: IT_SysDesc

⌈DiagnosticConnection shall only reference (via the indirection created by TpConnectionIdent) the following sub-classes of the meta-class TpConnection:

- CanTpConnection
- FlexrayTpConnection
- DoIpTpConnection

⌋

Class	**DiagnosticConnection**			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. **Tags:** atp.recommendedPackage=DiagnosticConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**DiagnosticConnection**			
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages. **Tags:** atp.Splitkey=functionalRequest
periodic ResponseUudt	PduTriggering	*	ref	Reference to UUDT responses. **Tags:** atp.Splitkey=periodicResponseUudt
physical Request	TpConnectionIdent	0..1	ref	Reference to a physical request message. **Tags:** atp.Splitkey=physicalRequest
response	TpConnectionIdent	0..1	ref	In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed. **Tags:** atp.Splitkey=response

Table 6.275: DiagnosticConnection

Class	***TpConnection*** (abstract)			
Note	TpConnection Base Class.			
Base	*ARObject*			
Subclasses	CanTpConnection, DoIpTpConnection, EthTpConnection, FlexrayTpConnection, J1939TpConnection, LinTpConnection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ident	TpConnectionIdent	0..1	aggr	This adds the ability to become referrable to Tp Connection. **Stereotypes:** atpIdentityContributor

Table 6.276: TpConnection

Class	**TpConnectionIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable Tp Connection.			
Base	*ARObject*, *Referrable*			
Aggregated by	*TpConnection*.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.277: TpConnectionIdent

6.8.11 IEEE1722 Transport Layer

`IEEE1722Tp` is a transport protocol for Time-Sensitive Applications [41].

On the Ethernet `IEEE1722Tp` is represented by an own `AVTP Ethertype` [41]. From an AUTOSAR layered architecture perspective the `IEEE1722Tp` messages are handled by the `IEEE1722Tp` module.

In AUTOSAR the Audio Video Transport Protocol (AVTP) supports two major use-cases:

- clock reference, audio, and video transport

- control message transport (e.g. CAN frame transport).

The elaborate modelling of the IEEE1722Tp subtype messages (e.g. for audio, video, or control data) provides details about the kind of messages transported. Thus producer and consumer have a common understanding of the payload.

This elaborate modelling of the IEEE1722Tp subtype messages allows also for a detailed definition of the actually required bandwidth on the ethernet channel. This supports the calculation of the busload.

The common parts to define an IEEE1722Tp message are described using the IEEE1722TpConnection.

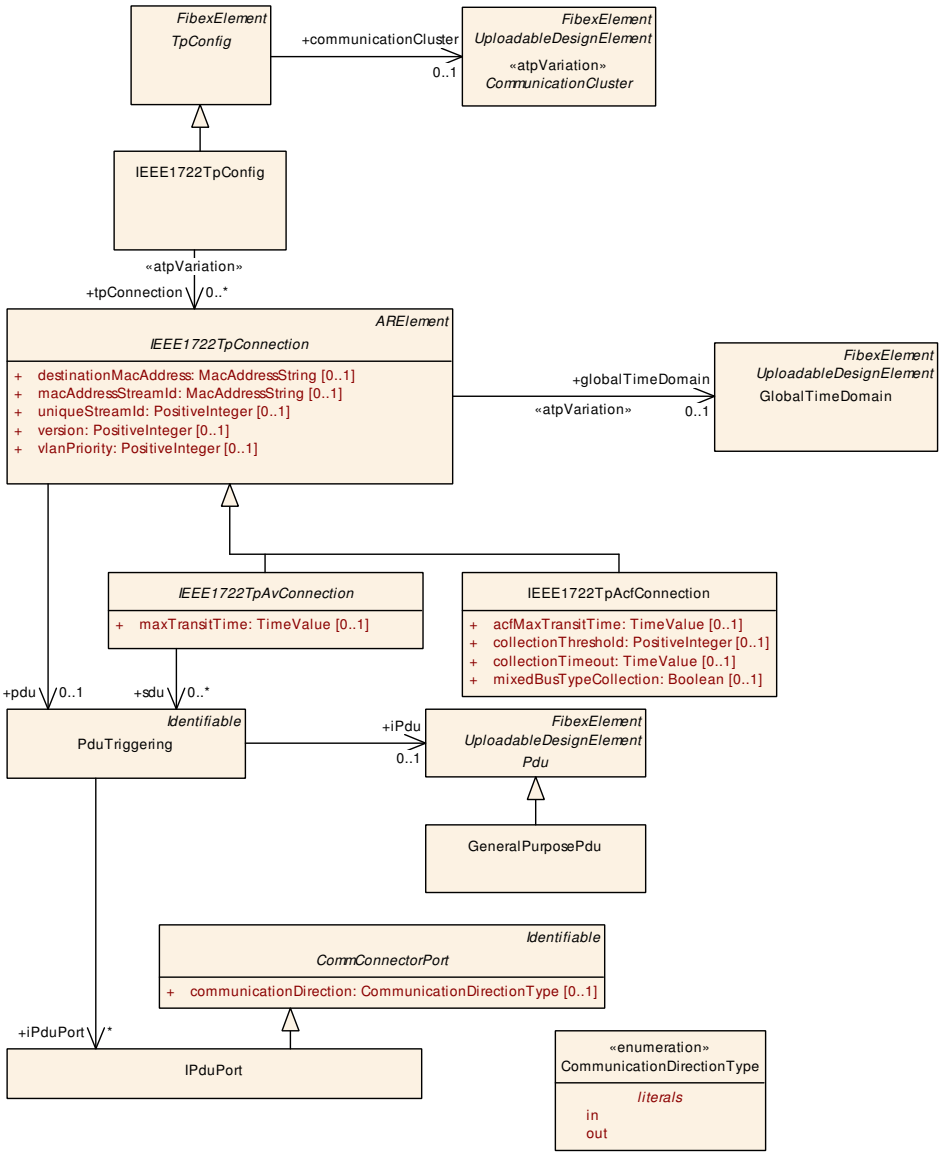

Figure 6.87: Modeling of IEEE1722TpConnection

The packet identification (Stream Id) is defined by the two attributes macAddressStreamId and uniqueStreamId.

[TPS_SYST_03098] IEEE1722Tp Stream Id definition

Status: DRAFT

⌈The attributes IEEE1722TpConnection.macAddressStreamId and IEEE1722TpConnection.uniqueStreamId are concatenated to form the stream_id [41] value for an IEEE1722Tp message.⌋

[constr_3743] Allowed values for IEEE1722TpConnection.uniqueStreamId

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The value for IEEE1722TpConnection.uniqueStreamId shall be in the range between 0 and 65535.⌋

[constr_3747] Existence of attribute IEEE1722TpConnection.uniqueStreamId

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpConnection, the attribute uniqueStreamId shall exist.⌋

[constr_3748] Existence of attribute IEEE1722TpConnection.macAddressStreamId

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpConnection, the attribute macAddressStreamId shall exist.⌋

[TPS_SYST_03100] IEEE1722Tp destination MAC address definition

Status: DRAFT

⌈In case the IEEE1722Tp message is sent, then the destination MAC address is determined in one of two ways:

- explicitly using the IEEE1722TpConnection.destinationMacAddress definition,
- or, if destinationMacAddress is not defined, implicitly using the IEEE1722TpConnection.macAddressStreamId definition.

⌋

[TPS_SYST_03099] IEEE1722Tp Stream Version definition

Status: DRAFT

⌈The attribute version defines the version [41] value for an IEEE1722Tp message.⌋

[constr_3744] Allowed values for IEEE1722TpConnection.version

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The value for IEEE1722TpConnection.version shall be in the range between 0 and 7.⌋

[constr_3749] Existence of attribute IEEE1722TpConnection.version

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpConnection, the attribute version shall exist.⌋

The reference IEEE1722TpConnection.pdu refers to a PduTriggering which in turn refers to a GeneralPurposePdu of category IEEE1722TP, This GeneralPurposePdu is used for the transport of all the IEEE1722Tp messages on a specific EthernetPhysicalChannel in one IPduPort.communicationDirection (either CommunicationDirectionType.in or CommunicationDirectionType.out) as defined by the PduTriggering. As the lower layers (below IEEE1722Tp) are not able to differentiate between different IEEE1722Tp messages (based on the Stream Id or other IEEE1722Tp specific criteria) only one GeneralPurposePdu of category IEEE1722TP per EthernetPhysicalChannel and communicationDirection is required and supported.

[constr_3745] category of GeneralPurposePdu referenced in the role IEEE1722TpConnection.pdu

Status: DRAFT
Imposition time: CP: IT_EcuExt

⌈The GeneralPurposePdu referenced by the PduTriggering which in turn is referenced in the role pdu by the IEEE1722TpConnection shall have the category set to IEEE1722TP.⌋

[constr_3750] Existence of attribute IEEE1722TpConnection.pdu

Status: DRAFT
Imposition time: CP: IT_EcuExt

⌈For each IEEE1722TpConnection, the reference pdu shall exist.⌋

[TPS_SYST_03109] Transmission of an IEEE1722TpConnection

Status: DRAFT

⌈If the IEEE1722TpConnection references a PduTriggering in role pdu and that PduTriggering refers to an IPduPort (in the context of an EcuInstance) in the role iPduPort with communicationDirection set to CommunicationDirectionType.out,

then the IEEE1722TpConnection is transmitted by that EcuInstance.⌋

In the IEEE1722Tp context [41] a transmitting EcuInstance (according to [TPS_SYST_03109]) is also called a talker or producer.

[TPS_SYST_03110] Reception of an IEEE1722TpConnection

Status: DRAFT

⌈If the IEEE1722TpConnection references a PduTriggering in role pdu and that PduTriggering refers to an IPduPort (in the context of an EcuInstance) in the role iPduPort with communicationDirection set to CommunicationDirectionType.in,

then the IEEE1722TpConnection is received by that EcuInstance.⌋

In the IEEE1722Tp context [41] a receiving EcuInstance (according to [TPS_SYST_03110]) is also called a listener or consumer.

Class	**IEEE1722TpConfig**			
Note	Definition of the IEEE1722Tp protocol. **Tags:** atp.Status=candidate atp.recommendedPackage=TpConfigs			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tpConnection	IEEE1722TpConnection	*	ref	Collection of IEEE1722Tp connections. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=tpConnection, tpConnection.ieeE1722Tp Connection, tpConnection.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild

Table 6.278: IEEE1722TpConfig

Class	***IEEE1722TpConnection*** (abstract)			
Note	Definition of the IEEE1722Tp protocol. **Tags:** atp.Status=candidate			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	IEEE1722TpAcfConnection, *IEEE1722TpAvConnection*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationMac Address	MacAddressString	0..1	attr	Optional definition of the destination MAC address for this stream. If no given then macAddressStreamId is used as destination MAC address. **Tags:** atp.Status=candidate
globalTime Domain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this IEEE1722Tp Connection shall be synchronized with. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTime Domain.globalTimeDomain, globalTimeDomain.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

▽

△

Class	*IEEE1722TpConnection* (abstract)			
macAddress StreamId	MacAddressString	0..1	attr	MAC Address part of the Stream Id. **Tags:** atp.Status=candidate
pdu	PduTriggering	0..1	ref	Reference to the lower layer Pdu used for the IEEE1722 Tp protocol transport. **Tags:** atp.Splitkey=pdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.
uniqueStreamId	PositiveInteger	0..1	attr	Unique Id part of the Stream Id. **Tags:** atp.Status=candidate
version	PositiveInteger	0..1	attr	Version of the IEEE1722TP stream. **Tags:** atp.Status=candidate
vlanPriority	PositiveInteger	0..1	attr	Optional definition of the VLAN priority for this stream.

Table 6.279: IEEE1722TpConnection

6.8.11.1 Audio Video Transport

For the definition of actual audio/video transport the `IEEE1722TpAvConnection` is used.

The reference `IEEE1722TpAvConnection.sdu` refers to a `PduTriggering` which in turn refers to a `GeneralPurposePdu` of `category` `IEEE1722TP_STREAM` which is used for the transport of the stream specific payload from or to software where the payload is produced or consumed. In figure 6.88 the relation of the `Pdu category` is shown.

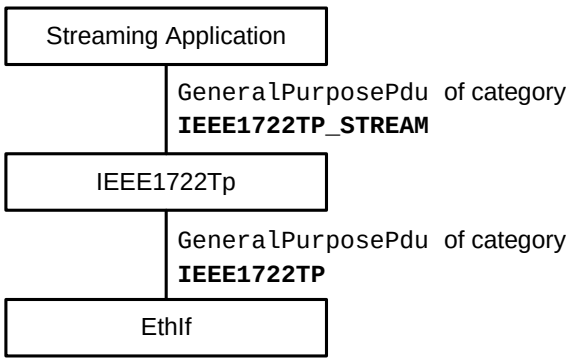

Figure 6.88: Usage of `Pdu category` for `IEEE1722Tp`

[constr_3746] `category` of `GeneralPurposePdu` referenced in the role `IEEE1722TpAvConnection.sdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈The `GeneralPurposePdu` referenced by the `PduTriggering` which in turn is referenced in the role `IEEE1722TpAvConnection.sdu` by the `IEEE1722TpAvConnection` shall have the `category` set to `IEEE1722TP_STREAM`.⌋

[constr_3761] Identical `EthernetPhysicalChannel` owning `PduTriggering`s referenced by `IEEE1722TpConnection.pdu` and `IEEE1722TpAvConnection.sdu`

Status: DRAFT
Imposition time: CP: `IT_EcuExt`

⌈The `PduTriggering`s referenced in the roles `IEEE1722TpConnection.pdu` and `IEEE1722TpAvConnection.sdu` shall be owned by the same `EthernetPhysicalChannel`.⌋

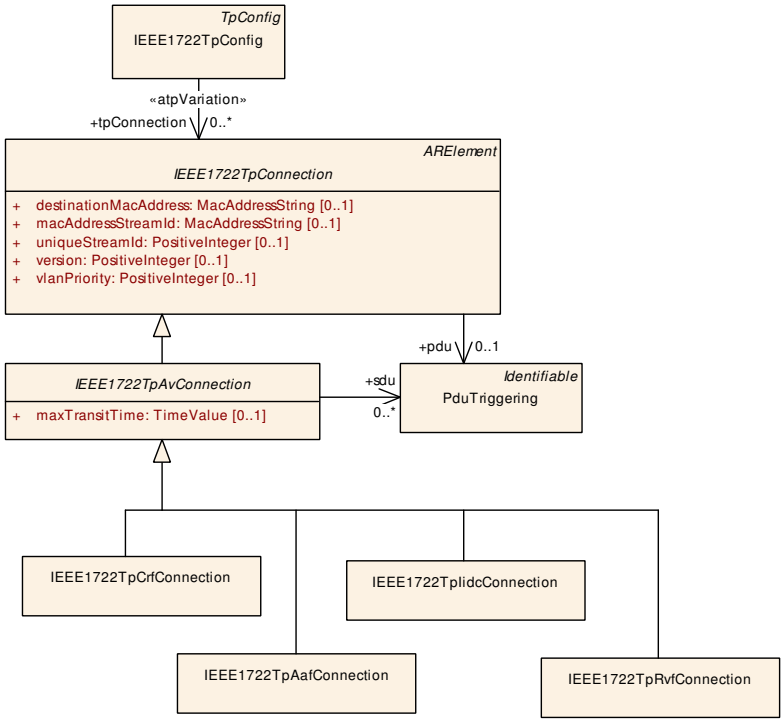

Figure 6.89: Modeling of `IEEE1722TpAvConnection`

Class	***IEEE1722TpAvConnection*** (abstract)
Note	AV IEEE1722Tp connection. **Tags:** atp.Status=candidate
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	***IEEE1722TpAvConnection*** (abstract)			
Subclasses	IEEE1722TpAafConnection, IEEE1722TpCrfConnection, IEEE1722TplidcConnection, IEEE1722TpRvfConnection			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxTransitTime	TimeValue	0..1	attr	Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.
sdu	PduTriggering	*	ref	Reference to the upper layer Sdu used for the transport of the payload of the IEEE1722Tp. **Tags:** atp.Splitkey=sdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.280: IEEE1722TpAvConnection

6.8.11.2 Audio Video Transport CRF

The Clock Reference Format (CRF) supports the distribution of a common clock in an audio/video/control system is used to synchronize events within a system.

The configuration details are shown in figure 6.90 and defined using `IEEE1722Tp-CrfConnection`.

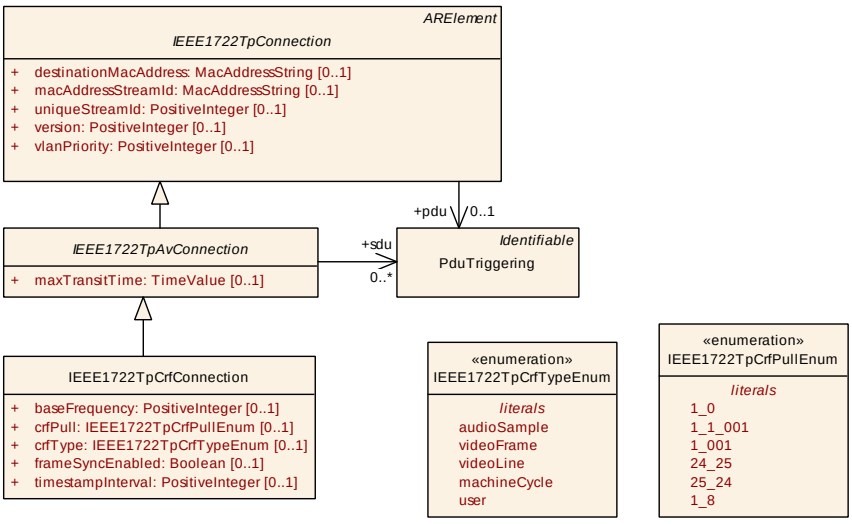

Figure 6.90: Modeling of `IEEE1722TpCrfConnection`

Class	**IEEE1722TpCrfConnection**			
Note	AV IEEE1722Tp CRF connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpAvConnection*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseFrequency	PositiveInteger	0..1	attr	CRF base frequency in Hz. **Tags:** atp.Status=candidate
crfPull	IEEE1722TpCrfPull Enum	0..1	attr	Definition of the CRF stream pull value.
crfType	IEEE1722TpCrfType Enum	0..1	attr	Definition of the CRF stream type.
frameSync Enabled	Boolean	0..1	attr	Defines whether the "fs" (frame sync) shall be enabled. **Tags:** atp.Status=candidate
timestamp Interval	PositiveInteger	0..1	attr	CRF timestamp interval as multiple of the baseFrequency. **Tags:** atp.Status=candidate

Table 6.281: IEEE1722TpCrfConnection

Enumeration	**IEEE1722TpCrfTypeEnum**
Note	Definition of the CRF stream type. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpCrfConnection.crfType
Literal	***Description***
audioSample	CRF_AUDIO_SAMPLE, Audio sample timestamp **Tags:** atp.EnumerationLiteralIndex=0
machineCycle	CRF_MACHINE_CYCLE, Machine cycle timestamp **Tags:** atp.EnumerationLiteralIndex=3
user	CRF_USER, User specified **Tags:** atp.EnumerationLiteralIndex=4
videoFrame	CRF_VIDEO_FRAME, Video frame sync timestamp **Tags:** atp.EnumerationLiteralIndex=1
videoLine	CRF_VIDEO_LINE, Video line sync timestamp **Tags:** atp.EnumerationLiteralIndex=2

Table 6.282: IEEE1722TpCrfTypeEnum

Enumeration	**IEEE1722TpCrfPullEnum**
Note	Definition of the CRF stream pull value. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpCrfConnection.crfPull
Literal	***Description***

▽

△

Enumeration	IEEE1722TpCrfPullEnum
_1_0	Multiply base_frequency field by 1.0 **Tags:** atp.EnumerationLiteralIndex=0 xml.name=1-0
_1_001	Multiply base_frequency field by 1.001 **Tags:** atp.EnumerationLiteralIndex=2 xml.name=1-001
_1_1_001	Multiply base_frequency field by 1/1.001 **Tags:** atp.EnumerationLiteralIndex=1 xml.name=1-1-001
_1_8	Multiply base_frequency field by 1/8 **Tags:** atp.EnumerationLiteralIndex=5 xml.name=1-8
_24_25	Multiply base_frequency field by 24/25 **Tags:** atp.EnumerationLiteralIndex=3 xml.name=24-25
_25_24	Multiply base_frequency field by 25/24 **Tags:** atp.EnumerationLiteralIndex=4 xml.name=25-24

Table 6.283: IEEE1722TpCrfPullEnum

6.8.11.3 Audio Video Transport AAF

The AVTP Audio Format (AAF) provides a mechanism for transporting audio data over an AVTP network.

The configuration details are shown in figure 6.91 and defined using `IEEE1722Tp-AafConnection`.

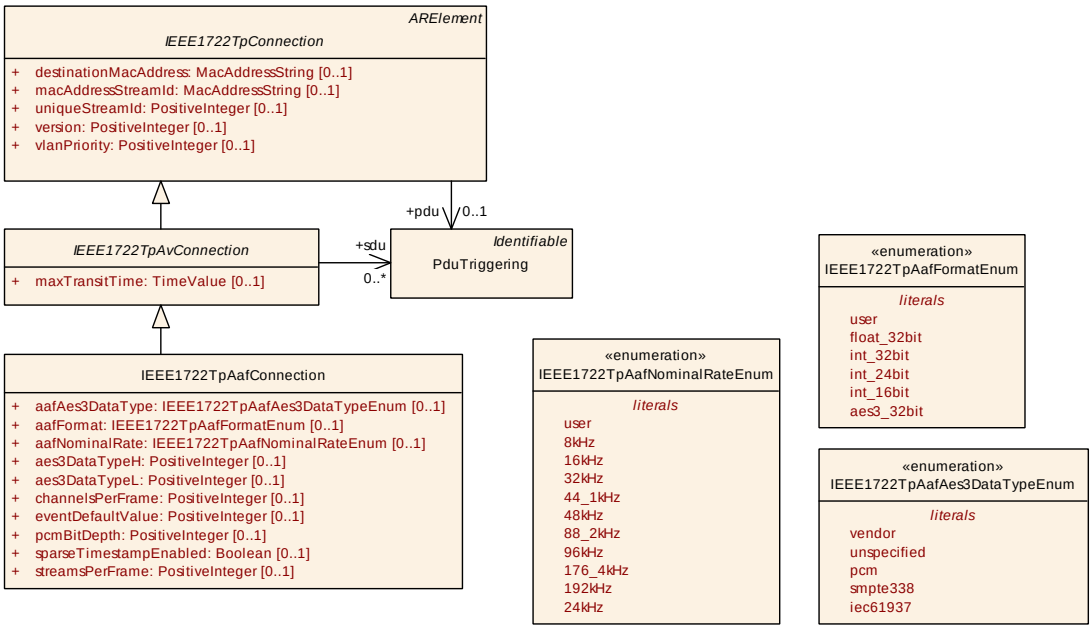

Figure 6.91: Modeling of `IEEE1722TpAafConnection`

Class	**IEEE1722TpAafConnection**			
Note	AV IEEE1722Tp AAF connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpAvConnection*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aafAes3Data Type	IEEE1722TpAafAes3 DataTypeEnum	0..1	attr	Definition of the AAF AES3 stream aes3_data_type reference.
aafFormat	IEEE1722TpAafFormat Enum	0..1	attr	Definition of the AAF stream format.
aafNominalRate	IEEE1722TpAaf NominalRateEnum	0..1	attr	Definition of the AAF stream nominal sample / frame rate. For an AAF PCM stream this is the nominal sample rate. For an AAF AES3 stream this is the nominal frame rate.
aes3DataTypeH	PositiveInteger	0..1	attr	Definition of the AAF AES3 aes3_data_type_h default value. **Tags:** atp.Status=candidate
aes3DataTypeL	PositiveInteger	0..1	attr	Definition of the AAF AES3 aes3_data_type_l default value. **Tags:** atp.Status=candidate
channelsPer Frame	PositiveInteger	0..1	attr	Definition of the AAF PCM stream channels_per_frame. e.g. 1: mono, 2: stereo, 8: 7.1 multicannel **Tags:** atp.Status=candidate
eventDefault Value	PositiveInteger	0..1	attr	Definition of a value to be used for the 4-bit "evt" field. **Tags:** atp.Status=candidate

△

Class	**IEEE1722TpAafConnection**			
pcmBitDepth	PositiveInteger	0..1	attr	Definition of the AAF PCM stream bit_depth. e.g. 16, 24, 32. **Tags:** atp.Status=candidate
sparse Timestamp Enabled	Boolean	0..1	attr	Defines whether the "sp" (sparse timestamp) shall be enabled. false: Normal operation, timestamp in every AAF AVTPDU true: Sparse mode, timestamp in every eighth AAF AVTPDU **Tags:** atp.Status=candidate
streamsPer Frame	PositiveInteger	0..1	attr	AAF AES3 stream streams_per_frame. **Tags:** atp.Status=candidate

Table 6.284: IEEE1722TpAafConnection

Enumeration	**IEEE1722TpAafNominalRateEnum**
Note	Definition of the AAF nominal sample / frame rate. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpAafConnection.aafNominalRate
Literal	*Description*
_16kHz	16 kHz **Tags:** atp.EnumerationLiteralIndex=2 xml.name=16-KHZ
_176_4kHz	176.4 kHz **Tags:** atp.EnumerationLiteralIndex=8 xml.name=176-4-KHZ
_192kHz	192 kHz **Tags:** atp.EnumerationLiteralIndex=9 xml.name=192-KHZ
_24kHz	24 kHz **Tags:** atp.EnumerationLiteralIndex=10 xml.name=24-KHZ
_32kHz	32 kHz **Tags:** atp.EnumerationLiteralIndex=3 xml.name=32-KHZ
_44_1kHz	44.1 kHz **Tags:** atp.EnumerationLiteralIndex=4 xml.name=44-1-KHZ
_48kHz	48 kHz **Tags:** atp.EnumerationLiteralIndex=5 xml.name=48-KHZ

▽

△

Enumeration	**IEEE1722TpAafNominalRateEnum**
_88_2kHz	88.2 kHz **Tags:** atp.EnumerationLiteralIndex=6 xml.name=88-2-KHZ
_8kHz	8 kHz **Tags:** atp.EnumerationLiteralIndex=1 xml.name=8-KHZ
_96kHz	96 kHz **Tags:** atp.EnumerationLiteralIndex=7 xml.name=96-KHZ
user	User specified **Tags:** atp.EnumerationLiteralIndex=0

Table 6.285: IEEE1722TpAafNominalRateEnum

Enumeration	**IEEE1722TpAafFormatEnum**
Note	Definition of the AAF stream format. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpAafConnection.aafFormat
Literal	***Description***
aes3_32bit	AES3_32BIT, 32-bit AES3 format, AES3 **Tags:** atp.EnumerationLiteralIndex=5 xml.name=AES-3-32-BIT
float_32bit	FLOAT_32BIT, 32bit floating, PCM **Tags:** atp.EnumerationLiteralIndex=1 xml.name=FLOAT-32-BIT
int_16bit	INT_16BIT, 16 bit integer, PCM **Tags:** atp.EnumerationLiteralIndex=4 xml.name=INT-16-BIT
int_24bit	INT_24BIT, 24 bit integer, PCM **Tags:** atp.EnumerationLiteralIndex=3 xml.name=INT-24-BIT
int_32bit	INT_32BIT, 32bit integer, PCM **Tags:** atp.EnumerationLiteralIndex=2 xml.name=INT-32-BIT
user	USER, user specific, PCM **Tags:** atp.EnumerationLiteralIndex=0

Table 6.286: IEEE1722TpAafFormatEnum

Enumeration	**IEEE1722TpAafAes3DataTypeEnum**
Note	Definition of the AAF AES3 stream aes3_data_type reference. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpAafConnection.aafAes3DataType
Literal	***Description***
iec61937	Data type reference is IEC 61937-2 **Tags:** atp.EnumerationLiteralIndex=4
pcm	Data type is PCM **Tags:** atp.EnumerationLiteralIndex=2
smpte338	Data type reference is SMPTE ST 338 **Tags:** atp.EnumerationLiteralIndex=3
unspecified	Data type not specified **Tags:** atp.EnumerationLiteralIndex=1
vendor	Data type reference is defined by vendor **Tags:** atp.EnumerationLiteralIndex=0

Table 6.287: IEEE1722TpAafAes3DataTypeEnum

An example of an `IEEE1722TpAafConnection` is illustrated in figure 6.92.

The `IEEE1722TpAafConnection` defines the audio stream credentials for this AAF stream.

In figure 6.92 the producer part of the `IEEE1722TpAafConnection` is shown. This is recognizable because the `IEEE1722TpAafConnection` refers to an `PduTriggering` *ProducerTransportPduTriggering* in the role `pdu` and that `PduTriggering` refers to an `IPduPort` *ProducerPort* that has the `communicationDirection` defined as `CommunicationDirectionType.out`.

During runtime the producer software will create the content of the *StreamPdu* and pass it to the `IEEE1722Tp` module. The `IEEE1722Tp` is able to match the `GeneralPurposePdu` *StreamPdu* with this `IEEE1722TpAafConnection` and will create the header of the `IEEE1722Tp` message according to the configuration details given in *AafConnection*. The `IEEE1722Tp` message is then handed to the Ethernet stack as the `GeneralPurposePdu` *GPP_1722TpTransportPdu* for sending on the network.

Whether the actual stream data is copied or handled via DMA during this process is up to the hardware capabilities and the configuration of the `IEEE1722Tp` module.

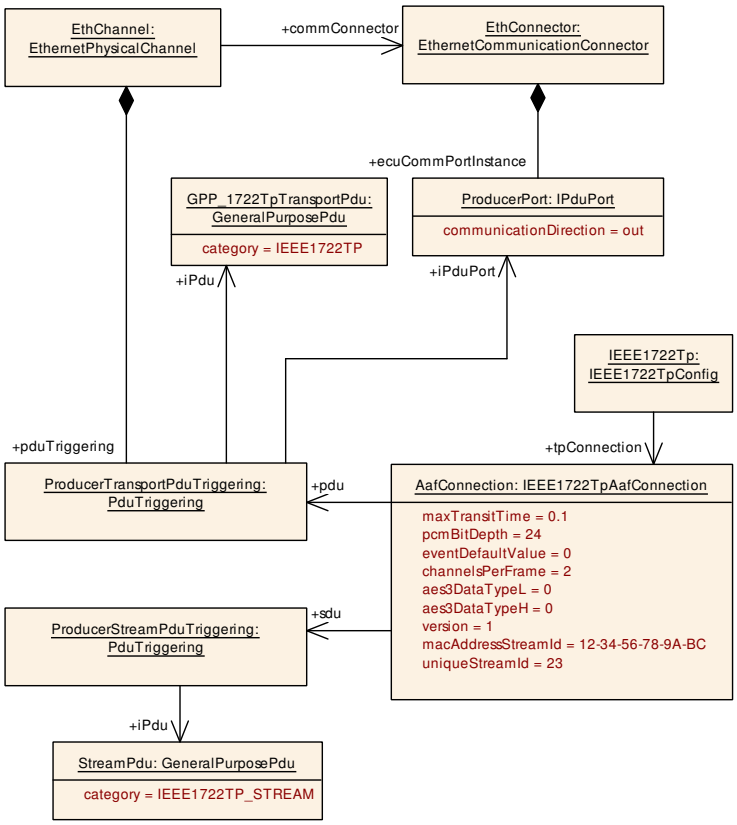

Figure 6.92: Example model of `IEEE1722TpAafConnection` producer

In figure 6.93 the same `IEEE1722TpAafConnection` is shown for the consumer part. Here the `IPduPort` *ConsumerPort* that has the `communicationDirection` defined as `CommunicationDirectionType.in`.

During runtime the `IEEE1722Tp` message is received and (due to the `Ethertype` for `IEEE1722Tp`) handed to the `IEEE1722Tp` module as `GeneralPurposePdu` *GPP_1722TpTransportPdu*. Based on the used stream id ([TPS_SYST_03098]) the `IEEE1722Tp` message is associated with this `IEEE1722TpAafConnection` and is indicated to the actual consumer software as `GeneralPurposePdu` *StreamPdu*.

Whether the actual stream data is copied or handled via DMA during this process is up to the hardware capabilities and the configuration of the `IEEE1722Tp` module.

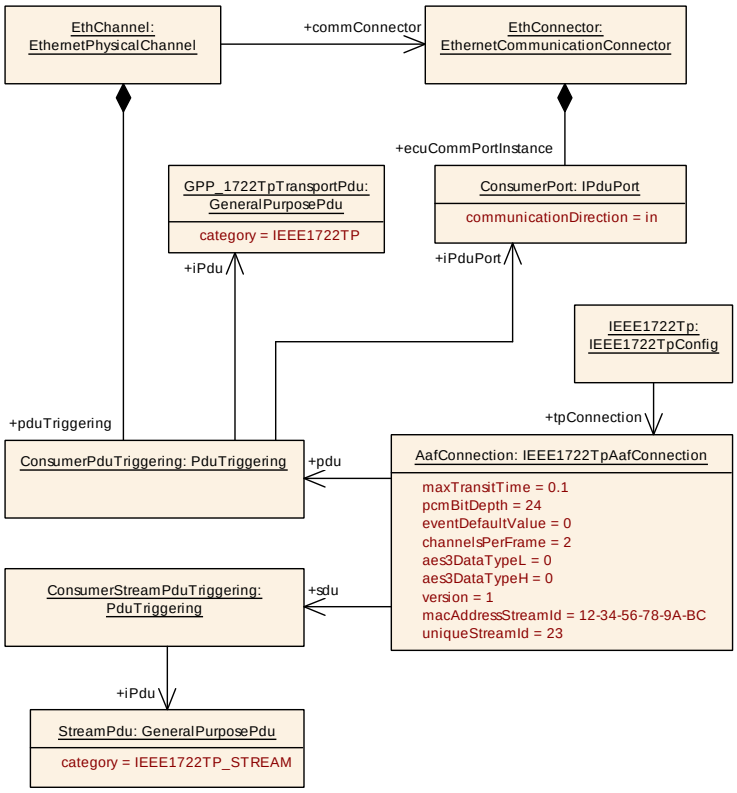

Figure 6.93: Example model of `IEEE1722TpAafConnection` consumer

6.8.11.4 Audio Video Transport using IIDC and IEC61883

The AVTP IIDC and IEC61883 format provides a mechanism for transporting audio and video data over an AVTP network.

The configuration details are shown in figure 6.94 and defined using `IEEE1722Tp-IidcConnection`.

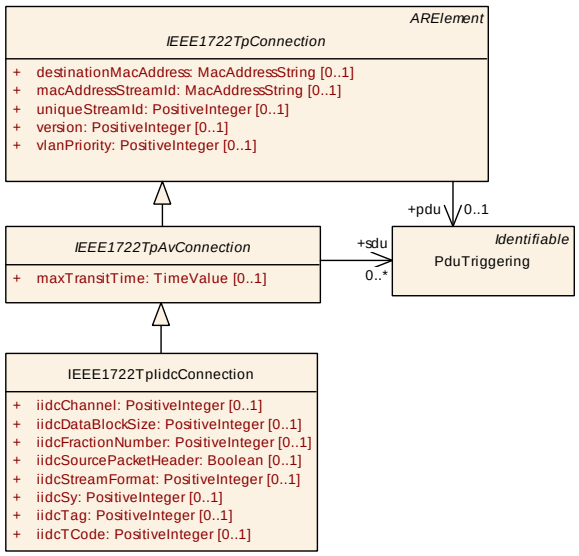

Figure 6.94: Modeling of `IEEE1722TpIidcConnection`

Class	IEEE1722TplidcConnection			
Note	AV IEEE1722Tp IIDC and IEC61883 connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpAvConnection*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iidcChannel	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 channel. **Tags:** atp.Status=candidate
iidcDataBlock Size	PositiveInteger	0..1	attr	Definition of the IEC61883 data block size (DBS). **Tags:** atp.Status=candidate
iidcFraction Number	PositiveInteger	0..1	attr	Definition of the IEC61883 fractionNumber (FN). **Tags:** atp.Status=candidate
iidcSource PacketHeader	Boolean	0..1	attr	Defines the IEC61883 source packet header (SPH) existence. **Tags:** atp.Status=candidate
iidcStream Format	PositiveInteger	0..1	attr	Definition of the IEC61883 stream format (FMT). **Tags:** atp.Status=candidate
iidcSy	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 sy. **Tags:** atp.Status=candidate
iidcTag	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 tag. **Tags:** atp.Status=candidate
iidcTCode	PositiveInteger	0..1	attr	Definition of the IIDC and IEC61883 tcode. **Tags:** atp.Status=candidate

Table 6.288: IEEE1722TplidcConnection

[TPS_SYST_03132] `category` of `IEEE1722TpIidcConnection`

Status: DRAFT

⌈The `category` of `IEEE1722TpIidcConnection` defines which kind of protocol shall be used for the transport of the `IEEE1722TpIidcConnection`

- if the `IEEE1722TpIidcConnection` shall be transported using IIDC, then the `category` of the `IEEE1722TpIidcConnection` shall be set to `IIDC`
- if the `IEEE1722TpIidcConnection` shall be transported using IEC61883, then the `category` of the `IEEE1722TpIidcConnection` shall be set to `IEC61883`.

⌋

[constr_3796] Applicable attributes of `IEEE1722TpIidcConnection` for category `IIDC`

Status: DRAFT
Imposition time: CP: `IT_SysDesc`

⌈If the `category` of an `IEEE1722TpIidcConnection` is set to `IIDC`, then the collection of attributes shall be restricted to the following list:

- iidcChannel
- iidcSy
- iidcTag
- iidcTCode

⌋

[constr_3797] Applicable attributes of IEEE1722TpIidcConnection for category IEC61883

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈If the category of an IEEE1722TpIidcConnection is set to IEC61883, then the collection of attributes shall be restricted to the following list:

- iidcChannel
- iidcDataBlockSize
- iidcFractionNumber
- iidcSourcePacketHeader
- iidcStreamFormat
- iidcSy
- iidcTag
- iidcTCode

⌋

6.8.11.5 Audio Video Transport RVF

The AVTP Raw Video Format (RVF) provides a mechanism for transporting raw video data over an AVTP network.

The configuration details are shown in figure 6.95 and defined using IEEE1722TpRvfConnection.

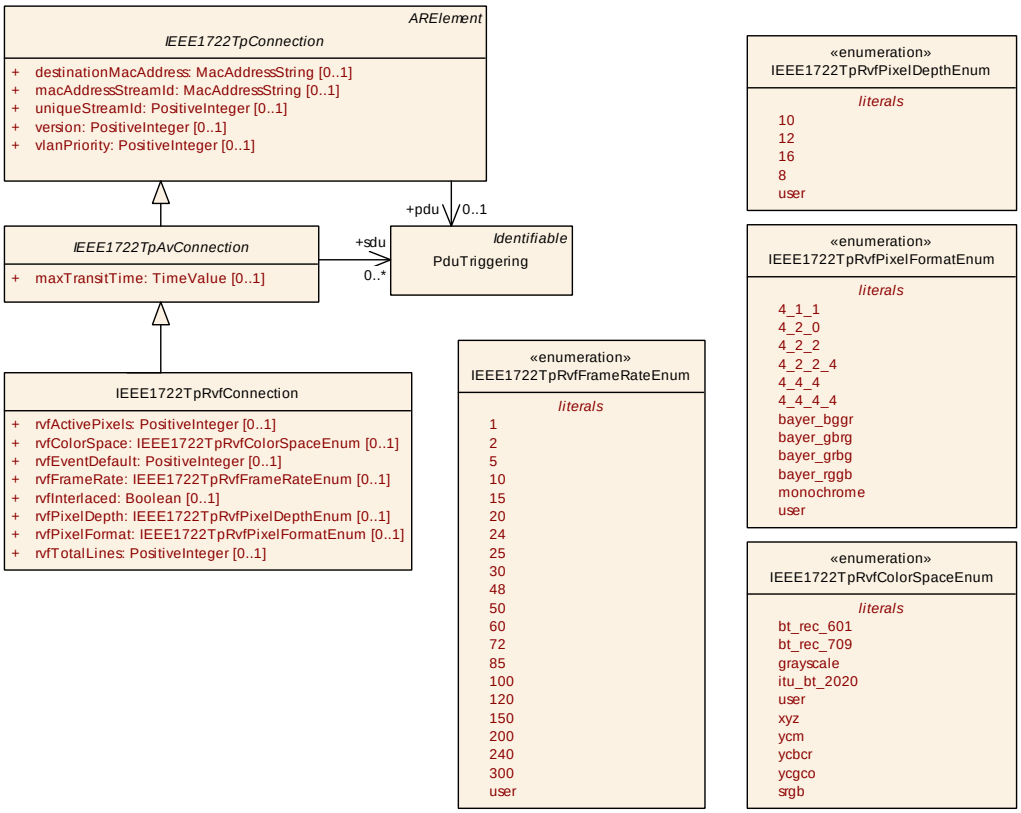

Figure 6.95: Modeling of `IEEE1722TpRvfConnection`

Class	**IEEE1722TpRvfConnection**			
Note	AV IEEE1722Tp RVF connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpAvConnection*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
rvfActivePixels	PositiveInteger	0..1	attr	Definition of the RVF stream active_pixels. **Tags:** atp.Status=candidate
rvfColorSpace	IEEE1722TpRvfColorSpaceEnum	0..1	attr	Definition of the RVF stream colorspace. **Tags:** atp.Status=candidate
rvfEventDefault	PositiveInteger	0..1	attr	Definition of the RVF stream event (evt) default value. **Tags:** atp.Status=candidate
rvfFrameRate	IEEE1722TpRvfFrameRateEnum	0..1	attr	Definition of the RVF stream frame_rate. **Tags:** atp.Status=candidate
rvfInterlaced	Boolean	0..1	attr	Defines the RVF stream interlaced (i). **Tags:** atp.Status=candidate
rvfPixelDepth	IEEE1722TpRvfPixelDepthEnum	0..1	attr	Definition of the RVF stream pixel_depth. **Tags:** atp.Status=candidate

△

Class	**IEEE1722TpRvfConnection**			
rvfPixelFormat	IEEE1722TpRvfPixelFormatEnum	0..1	attr	Definition of the RVF stream pixel_format. **Tags:** atp.Status=candidate
rvfTotalLines	PositiveInteger	0..1	attr	Definition of the RVF stream total_lines. **Tags:** atp.Status=candidate

Table 6.289: IEEE1722TpRvfConnection

Enumeration	**IEEE1722TpRvfPixelDepthEnum**
Note	Definition of the RVF Pixel Depth. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpRvfConnection.rvfPixelDepth
Literal	***Description***
_10	pixel depth 10 **Tags:** atp.EnumerationLiteralIndex=0 xml.name=10
_12	pixel depth 12 **Tags:** atp.EnumerationLiteralIndex=1 xml.name=12
_16	pixel depth 16 **Tags:** atp.EnumerationLiteralIndex=2 xml.name=16
_8	pixel depth 8 **Tags:** atp.EnumerationLiteralIndex=3 xml.name=8
user	pixel depth user defined **Tags:** atp.EnumerationLiteralIndex=4

Table 6.290: IEEE1722TpRvfPixelDepthEnum

Enumeration	**IEEE1722TpRvfPixelFormatEnum**
Note	Definition of the RVF Pixel Format. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpRvfConnection.rvfPixelFormat
Literal	***Description***
_4_1_1	pixel format 4:1:1 **Tags:** atp.EnumerationLiteralIndex=0 xml.name=4-1-1
_4_2_0	pixel format 4:2:0 **Tags:** atp.EnumerationLiteralIndex=1 xml.name=4-2-0

▽

△

Enumeration	**IEEE1722TpRvfPixelFormatEnum**
_4_2_2	pixel format 4:2:2 **Tags:** atp.EnumerationLiteralIndex=2 xml.name=4-2-2
_4_2_2_4	pixel format 4:2:2:4 **Tags:** atp.EnumerationLiteralIndex=3 xml.name=4-2-2-4
_4_4_4	pixel format 4:4:4 **Tags:** atp.EnumerationLiteralIndex=4 xml.name=4-4-4
_4_4_4_4	pixel format 4:4:4:4 **Tags:** atp.EnumerationLiteralIndex=5 xml.name=4-4-4-4
bayer_bggr	pixel format Bayer bggr **Tags:** atp.EnumerationLiteralIndex=6 xml.name=BAYER-BGGR
bayer_gbrg	pixel format Bayer gbrg **Tags:** atp.EnumerationLiteralIndex=7 xml.name=BAYER-GBRG
bayer_grbg	pixel format Bayer grbg **Tags:** atp.EnumerationLiteralIndex=8 xml.name=BAYER-GRBG
bayer_rggb	pixel format Bayer rggb **Tags:** atp.EnumerationLiteralIndex=9 xml.name=BAYER-RGGB
monochrome	pixel format Monochrome **Tags:** atp.EnumerationLiteralIndex=10
user	pixel format User defined **Tags:** atp.EnumerationLiteralIndex=11

Table 6.291: IEEE1722TpRvfPixelFormatEnum

Enumeration	**IEEE1722TpRvfColorSpaceEnum**
Note	Definition of the RVF stream colorspace. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpRvfConnection.rvfColorSpace
Literal	***Description***
bt_rec_601	BT Rec.601 **Tags:** atp.EnumerationLiteralIndex=0 xml.name=BT-REC-601

▽

△

Enumeration	**IEEE1722TpRvfColorSpaceEnum**
bt_rec_709	BT Rec.709 **Tags:** atp.EnumerationLiteralIndex=1 xml.name=BT-REC-709
grayscale	Grayscale **Tags:** atp.EnumerationLiteralIndex=2
itu_bt_2020	ITU BT 2020 **Tags:** atp.EnumerationLiteralIndex=3 xml.name=ITU-BT-2020
srgb	sRGB **Tags:** atp.EnumerationLiteralIndex=9
user	User defined **Tags:** atp.EnumerationLiteralIndex=4
xyz	XYZ **Tags:** atp.EnumerationLiteralIndex=5
ycbcr	YCbCr **Tags:** atp.EnumerationLiteralIndex=7
ycgco	YCgCo **Tags:** atp.EnumerationLiteralIndex=8
ycm	YCM **Tags:** atp.EnumerationLiteralIndex=6

Table 6.292: IEEE1722TpRvfColorSpaceEnum

Enumeration	**IEEE1722TpRvfFrameRateEnum**
Note	Definition of the RVF stream frame_rate. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpRvfConnection.rvfFrameRate
Literal	***Description***
_1	frame rate 1 **Tags:** atp.EnumerationLiteralIndex=0 xml.name=1
_10	frame rate 10 **Tags:** atp.EnumerationLiteralIndex=3 xml.name=10
_100	frame rate 100 **Tags:** atp.EnumerationLiteralIndex=14 xml.name=100
_120	frame rate 120 **Tags:** atp.EnumerationLiteralIndex=15 xml.name=120

▽

△

Enumeration	**IEEE1722TpRvfFrameRateEnum**
_15	frame rate 15 **Tags:** atp.EnumerationLiteralIndex=4 xml.name=15
_150	frame rate 150 **Tags:** atp.EnumerationLiteralIndex=16 xml.name=150
_2	frame rate 2 **Tags:** atp.EnumerationLiteralIndex=1 xml.name=2
_20	frame rate 20 **Tags:** atp.EnumerationLiteralIndex=5 xml.name=20
_200	frame rate 200 **Tags:** atp.EnumerationLiteralIndex=17 xml.name=200
_24	frame rate 24 **Tags:** atp.EnumerationLiteralIndex=6 xml.name=24
_240	frame rate 240 **Tags:** atp.EnumerationLiteralIndex=18 xml.name=240
_25	frame rate 25 **Tags:** atp.EnumerationLiteralIndex=7 xml.name=25
_30	frame rate 30 **Tags:** atp.EnumerationLiteralIndex=8 xml.name=30
_300	frame rate 300 **Tags:** atp.EnumerationLiteralIndex=19 xml.name=300
_48	frame rate 48 **Tags:** atp.EnumerationLiteralIndex=9 xml.name=48
_5	frame rate 5 **Tags:** atp.EnumerationLiteralIndex=2 xml.name=5
_50	frame rate 50 **Tags:** atp.EnumerationLiteralIndex=10 xml.name=50

▽

△

Enumeration	IEEE1722TpRvfFrameRateEnum
_60	frame rate 60 **Tags:** atp.EnumerationLiteralIndex=11 xml.name=60
_72	frame rate 72 **Tags:** atp.EnumerationLiteralIndex=12 xml.name=72
_85	frame rate 85 **Tags:** atp.EnumerationLiteralIndex=13 xml.name=85
user	frame rate User defined **Tags:** atp.EnumerationLiteralIndex=20

Table 6.293: IEEE1722TpRvfFrameRateEnum

6.8.11.6 Control Data Transport

The AVTP Control Format (ACF) can be used to transport control messages (e.g. CAN or LIN messages) using an AVTP stream.

Note that in AUTOSAR `R23-11` this ACF transport feature is not fully supported yet, as parts of the AUTOSAR Basic software are not yet adapted for this transport.

Figure 6.96 shows an example of connecting two CAN nodes (*Node3* and *Node4*) via an *Ethernet Network*. CAN communication originating at *Node3* is sent on *CanCluster A* and received in *Node1*. *Node1* is also connected to the *Ethernet Network*. The `IEEE1722Tp` stream configuration in *Node1* can be set up to take the CAN messages of *CanCluster A* and put the CAN messages into an `IEEE1722Tp` ACF stream. *Node1* is the producer of that `IEEE1722Tp` ACF stream.

Node2 is the consumer of the `IEEE1722Tp` ACF stream. *Node2* takes the CAN messages from the stream and sends them on *CanCluster B* where the CAN messages can eventually be received by CAN *Node4*.

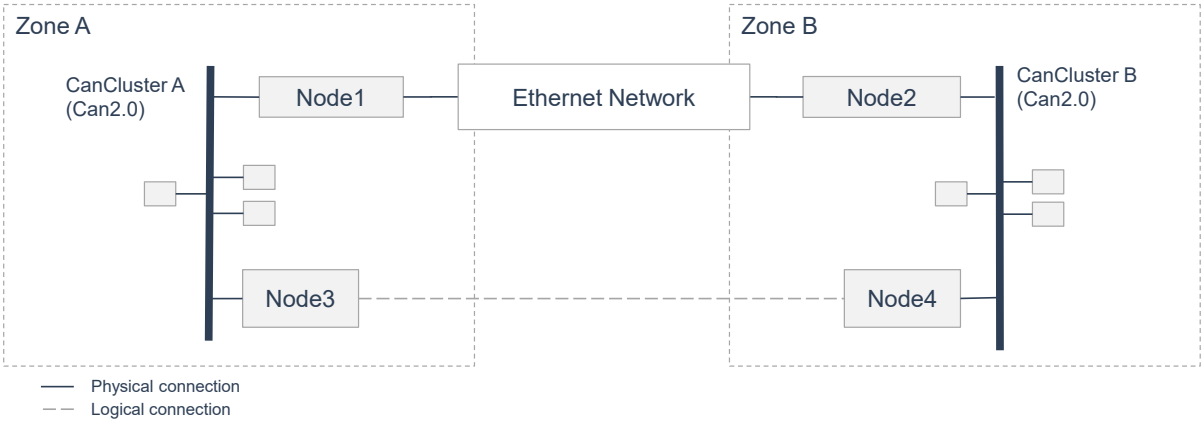

Figure 6.96: Example of `IEEE1722Tp` ACF stream transporting CAN messages

The IEEE1722TpAcfConnection may be configured to use either *Time-Synchronous Control Format* or *Non-Time-Synchronous Control Format*. This is defined by the existence of the attribute IEEE1722TpAcfConnection.acfMaxTransitTime.

[TPS_SYST_03116] Definition of Time-Synchronous Control Format

Status: DRAFT

⌈If the attribute IEEE1722TpAcfConnection.acfMaxTransitTime is defined, then the IEEE1722TpAcfConnection is transported using the *Time-Synchronous Control Format (TSCF)*.

If the attribute IEEE1722TpAcfConnection.acfMaxTransitTime is not defined, then the IEEE1722TpAcfConnection is transported using the *Non-Time-Synchronous Control Format (NTSCF)*.⌋

The IEEE1722TpAcfConnection defines an IEEE1722Tp stream which can transport control messages from one or multiple busses. Each transported bus is defined as an instance of IEEE1722TpAcfBus and is aggregated at the IEEE1722TpAcfConnection.acfTransportedBus.

Each transported bus defines a IEEE1722TpAcfBus.busId which is used to distinguish the potential multiple messages originating from different busses transported inside the IEEE1722Tp stream.

[constr_3751] Allowed values for IEEE1722TpAcfBus.busId

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The value for IEEE1722TpAcfBus.busId shall be in the range between 0 and 31.⌋

[constr_3752] Existence of attribute IEEE1722TpAcfBus.busId

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpAcfBus, the attribute busId shall exist.⌋

[TPS_SYST_03102] Values of IEEE1722TpAcfBus.busId per bus system

Status: DRAFT

⌈The values for IEEE1722TpAcfBus.busId are defined per bus system in the scope of one IEEE1722TpAcfConnection.

It is valid to define a transported IEEE1722TpAcfCan CAN bus with the identical busId value as a transported IEEE1722TpAcfLin LIN bus in the same IEEE1722TpAcfConnection. The bus kind in combination with the busId is used to distinguish the messages.⌋

The control messages that are put into one IEEE1722TpAcfConnection are typically much smaller than the size of one IEEE1722Tp ACF message. Thus a collection of several control messages into one IEEE1722Tp ACF message is supported. This

may introduce a slight delay in the transport of individual control messages as the IEEE1722Tp ACF message is waiting to be filled with several control messages. The time for collection and the fill rate are defined according to [TPS_SYST_03101].

[TPS_SYST_03101] Collection of several control messages in one IEEE1722Tp-AcfConnection

Status: DRAFT

⌈The attributes IEEE1722TpAcfConnection.collectionTimeout and IEEE1722TpAcfConnection.collectionThreshold control the collection behavior of the IEEE1722TpAcfConnection.⌋

If an individual control massage is put into the IEEE1722Tp ACF message it can be defined whether that specific control massage (defined as IEEE1722TpAcfBusPart) triggers the IEEE1722Tp ACF message to be sent immediately.

[TPS_SYST_03104] Triggering the immediate transmission of an IEEE1722Tp ACF message

Status: DRAFT

⌈The attribute IEEE1722TpAcfBusPart.collectionTrigger defines whether that IEEE1722TpAcfBusPart triggers immediate sending of the IEEE1722Tp ACF message.⌋

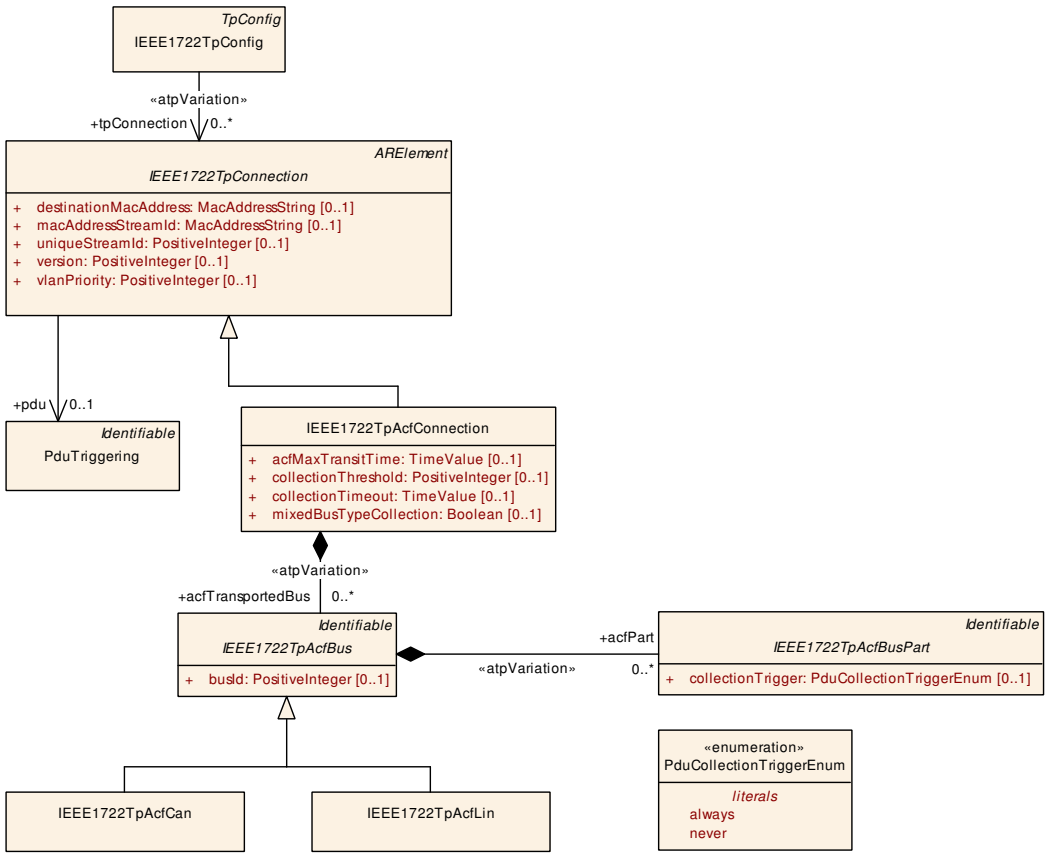

Figure 6.97: Modeling of IEEE1722TpAcfConnection

Class	IEEE1722TpAcfConnection			
Note	ACF IEEE1722Tp connection. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *IEEE1722TpConnection*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
acfMaxTransitTime	TimeValue	0..1	attr	Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.
acfTransportedBus	IEEE1722TpAcfBus	*	aggr	Definition of the transported busses over this ACF connection. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=acfTransportedBus, acfTransportedBus.shortName, acfTransportedBus.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
collectionThreshold	PositiveInteger	0..1	attr	Defines the size threshold in bytes which, when exceeded, triggers the sending of the IEEE1722Tp ACF message, even when the maxium IEEE1722Tp ACF message size has not been reached yet.
collectionTimeout	TimeValue	0..1	attr	When this timeout expires the IEEE1722Tp ACF message is triggered for sending. The respective timer is started when the first Pdu is put into the IEEE1722Tp ACF message. Defined in seconds.
mixedBusTypeCollection	Boolean	0..1	attr	Defines if this ACF-stream is allowed to collect ACF-messages of different bus kinds (i.e. whether it is allowed to collect CAN and LIN ACF-messages in one ACF-stream message).

Table 6.294: IEEE1722TpAcfConnection

Class	***IEEE1722TpAcfBus*** (abstract)			
Note	Abstract class to define various busses to be transported over a IEEE1722TP ACF connection. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	IEEE1722TpAcfCan, IEEE1722TpAcfLin			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
acfPart	IEEE1722TpAcfBusPart	*	aggr	One part transported over IEEE1722Tp channel. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=acfPart, acfPart.shortName, acfPart.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=postBuild
busId	PositiveInteger	0..1	attr	Id of the transported bus over the ACF connection.

Table 6.295: IEEE1722TpAcfBus

Class	***IEEE1722TpAcfBusPart*** (abstract)			
Note	Definition of one IEEE1722Tp ACF part transported over the IEEE1722Tp channel. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	IEEE1722TpAcfCanPart, IEEE1722TpAcfLinPart			
Aggregated by	*IEEE1722TpAcfBus*.acfPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
collectionTrigger	PduCollectionTrigger Enum	0..1	attr	Defines whether putting this AcfPart to the IEEE1722Tp ACF message triggers immediate sending of the IEEE1722Tp ACF message.

Table 6.296: IEEE1722TpAcfBusPart

6.8.11.7 Control Data Transport CAN

For the transport of CAN messages over an IEEE1722Tp ACF stream the definition of the IEEE1722TpAcfCan is used.

[TPS_SYST_03103] Abbreviated CAN message transport

Status: DRAFT

⌈The attribute IEEE1722TpAcfCan.messageType defines whether the CAN messages shall be transported in an abbreviated format (IEEE1722TpAcfCanMessageTypeEnum.canBrief) or in an non-abbreviated format (IEEE1722TpAcfCanMessageTypeEnum.can) in the IEEE1722Tp ACF stream.⌋

[constr_3754] Existence of attribute IEEE1722TpAcfCan.messageType

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpAcfCan, the attribute messageType shall exist.⌋

Which CAN messages are transported on an IEEE1722Tp ACF stream in the context of a dedicated IEEE1722TpAcfCan bus is defined using the IEEE1722TpAcfBus.acfPart aggregation.

[constr_3755] Consistend aggregation of IEEE1722TpAcfCanPart

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈An IEEE1722TpAcfCan shall only aggregate IEEE1722TpAcfCanParts in the role IEEE1722TpAcfCan.acfPart.⌋

[TPS_SYST_03107] Definition of CAN message transport in IEEE1722Tp ACF stream

Status: DRAFT

⌈Each CAN message to be transported on an IEEE1722Tp ACF stream is defined by having a dedicated IEEE1722TpAcfCanPart and referring from that IEEE1722TpAcfCanPart to a PduTriggering in the role sdu. This PduTriggering defines which Pdu shall be transported via the iPdu reference.⌋

[TPS_SYST_03108] Definition of CAN Id in IEEE1722Tp ACF stream

Status: DRAFT

⌈The CAN Id to be used when transporting a CAN message on an IEEE1722Tp ACF stream is defined using the IEEE1722TpAcfCanPart.canIdentifier.⌋

[constr_3758] Allowed values for IEEE1722TpAcfCanPart.canIdentifier

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The value for IEEE1722TpAcfCanPart.canIdentifier shall be in the range between 0 and 536870911.⌋

The CAN basic software supports the definition if CAN Id range reception (CanFrameTriggering.rxIdentifierRange). Thus it is not required to define an individual Pdu in the ECU that handles the transport of the CAN messages in the IEEE1722Tp ACF stream (both transmission and reception) for every Pdu that is transported in the IEEE1722Tp ACF stream.

If the ECU is not interested in the content of a set of CAN messages (because this message is neither produced nor consumed at this ECU), then a whole range of CAN messages (based on their CAN Id) can be handled as one GeneralPurposePdu with category set to IEEE1722TP_ID_RANGE.

In case the CAN messages are received on a CAN bus at the ECU and transmitted in an IEEE1722Tp ACF stream, the range reception may be configured at the CanFrameTriggering. All CAN messages out of the defined rxIdentifierRange will be transported in an anonymous way as the GeneralPurposePdu of category IEEE1722TP_ID_RANGE. The actually CAN Id is then transported along the GeneralPurposePdu in its MetaData.

There is however a special case for the range based reception of CAN messages: If for a specific CAN Id which is in a defined CAN Id reception range rxIdentifierRange there exists also a dedicated reception defined with the CanFrameTriggering.identifier attribute, then this CAN message will only be handled in the specific Pdu. Such a CAN message will not be handled additionally as the GeneralPurposePdu of category IEEE1722TP_ID_RANGE. This behavior needs to be considered when defining which Pdus shall be put into an IEEE1722Tp ACF stream.

[constr_3759] Existence of attribute IEEE1722TpAcfCanPart.canIdentifier for IEEE1722Tp ACF stream transmission

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈If an IEEE1722TpAcfCanPart is part of an IEEE1722TpAcfConnection which, according to [TPS_SYST_03109], is transmitted in that IEEE1722TpAcfConnection and

the IEEE1722TpAcfCanPart refers to a PduTriggering in the role sdu which in turn refers to a Pdu that is NOT a GeneralPurposePdu of category IEEE1722TP_ID_RANGE,

then the attribute IEEE1722TpAcfCanPart.canIdentifier shall exist.⌋

In case the CAN messages are received as part of an IEEE1722Tp ACF stream at the ECU and shall be sent on a CAN bus, the range reception may be configured at the IEEE1722TpAcfCanPart. All CAN messages out of the defined canIdentifierRange will be transported in an anonymous way as the GeneralPurposePdu of category IEEE1722TP_ID_RANGE. The actually CAN Id is then transported along the GeneralPurposePdu in its MetaData.

[constr_3760] Existence of attribute IEEE1722TpAcfCanPart.canIdentifierRange or canIdentifierMask for IEEE1722Tp ACF stream reception

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈If an IEEE1722TpAcfCanPart is part of an IEEE1722TpAcfConnection which, according to [TPS_SYST_03110], is received in that IEEE1722TpAcfConnection and

the IEEE1722TpAcfCanPart refers to a PduTriggering in the role sdu which in turn refers to a Pdu that is NOT a GeneralPurposePdu of category IEEE1722TP_ID_RANGE,

then the attribute IEEE1722TpAcfCanPart.canIdentifierRange or canIdentifierMask shall exist.⌋

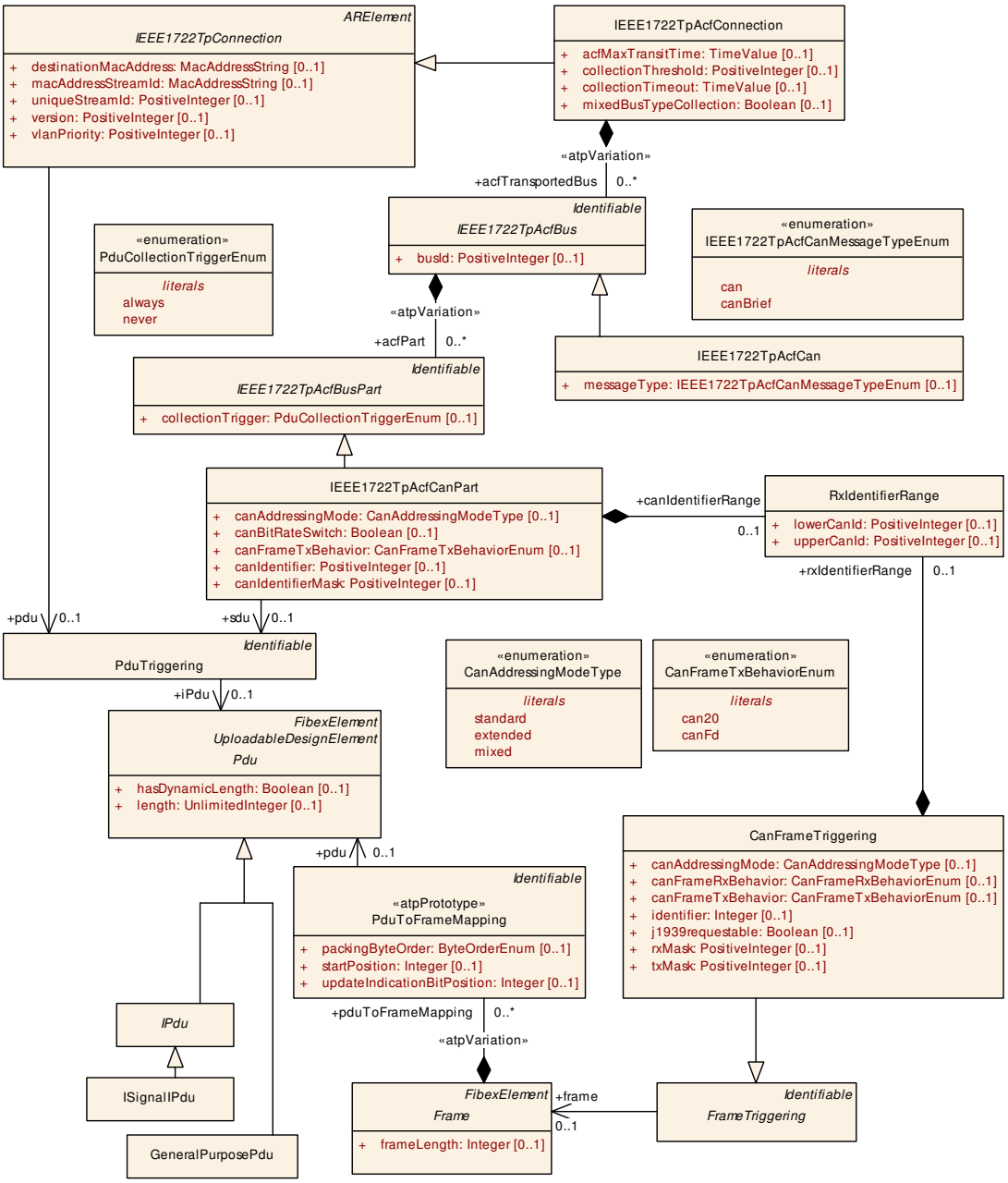

Figure 6.98: Modeling of `IEEE1722TpAcfCan`

Class	**IEEE1722TpAcfCan**			
Note	ACF IEEE1722Tp bus used for CAN transport. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARObject*, *IEEE1722TpAcfBus*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
messageType	IEEE1722TpAcfCan MessageTypeEnum	0..1	attr	Definition of the ACF CAN stream message type.

Table 6.297: IEEE1722TpAcfCan

Class	**IEEE1722TpAcfCanPart**			
Note	Definition of one CAN part (frame or frame range) transported over the IEEE1722Tp channel. **Tags:** atp.Status=candidate			
Base	*ARObject*, *IEEE1722TpAcfBusPart*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*IEEE1722TpAcfBus*.acfPart			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canAddressing Mode	CanAddressingMode Type	0..1	attr	Defines whether standard or extended address format shall be used.
canBitRate Switch	Boolean	0..1	attr	Defines whether the bit rate switch bit shall be set.
canFrameTx Behavior	CanFrameTxBehavior Enum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
canIdentifier	PositiveInteger	0..1	attr	Optional Can Id defined in case the Can Id can not be determined during runtime.
canIdentifier Mask	PositiveInteger	0..1	attr	CAN identifier mask which denotes relevant bits in the CAN Identifier. This attribute defines a CAN Identifier range in an alternative way to canIdentifierRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier.
canIdentifier Range	RxIdentifierRange	0..1	aggr	Definition of the identifier range for IEEE1722Tp ACF Can messages. **Tags:** atp.Splitkey=canIdentifierRange atp.Status=candidate
sdu	PduTriggering	0..1	ref	Reference to the Pdu transported in the IEEE1722Tp channel. **Tags:** atp.Splitkey=sdu atp.Status=candidate

Table 6.298: IEEE1722TpAcfCanPart

Enumeration	**IEEE1722TpAcfCanMessageTypeEnum**
Note	Definition of the ACF CAN stream message type. **Tags:** atp.Status=candidate
Aggregated by	IEEE1722TpAcfCan.messageType
Literal	***Description***
can	Defines the ACF CAN stream to use the ACF_CAN message type. **Tags:** atp.EnumerationLiteralIndex=0
canBrief	Defines the ACF CAN stream to use the ACF_CAN_BRIEF message type. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.299: IEEE1722TpAcfCanMessageTypeEnum

In the example figure 6.99 two use cases are shown:

- local reception of a dedicated CAN message from a `CanPhysicalChannel` and the transport of that CAN message to an `IEEE1722Tp` ACF stream
- transport of a CAN Id message range from a `CanPhysicalChannel` to an `IEEE1722Tp` ACF stream.

The local reception is defined to be a CAN message with the CAN identifier *23* and being configured to be handled as an ISignalIPdu *IndividualReceptionPdu* for further processing in this ECU (e.g. handling in the COM module).

Additionally that received ISignalIPdu is configured to be transmitted in the IEEE1722TpAcfConnection *AcfConnection* using the IEEE1722TpAcfCanPart.sdu reference. The message in the IEEE1722Tp ACF stream will be qualified with the busId *4* and the canIdentifier *23*. The transport in the IEEE1722Tp ACF stream will use the IEEE1722TpAcfCanMessageTypeEnum.canBrief format.

The transport of a CAN Id message range can be configured by first defining a CAN range reception using the CanFrameTriggering.rxIdentifierRange. This setup will put all CAN messages in the defined CAN Id range of *0 - 3000* into the GeneralPurposePdu *RangeReceptionPdu*. Additionally it is expected that the actual CAN identifier for a specific received CAN message will be put into the MetaData of that GeneralPurposePdu.

The GeneralPurposePdu *RangeReceptionPdu* is configured to be transmitted in the IEEE1722TpAcfConnection *AcfConnection* using the IEEE1722TpAcfCanPart.sdu reference. The message in the IEEE1722Tp ACF stream will be qualified with the busId *4* and will use the IEEE1722TpAcfCanMessageTypeEnum.canBrief format.

The actual CAN Id to be used when putting the CAN message into the IEEE1722Tp ACF stream is then taken from the MetaData of the GeneralPurposePdu. Therefore it is not required to define a canIdentifier at the IEEE1722TpAcfCanPart *RangeIdPart* of the IEEE1722TpAcfCan *AcfCan*.

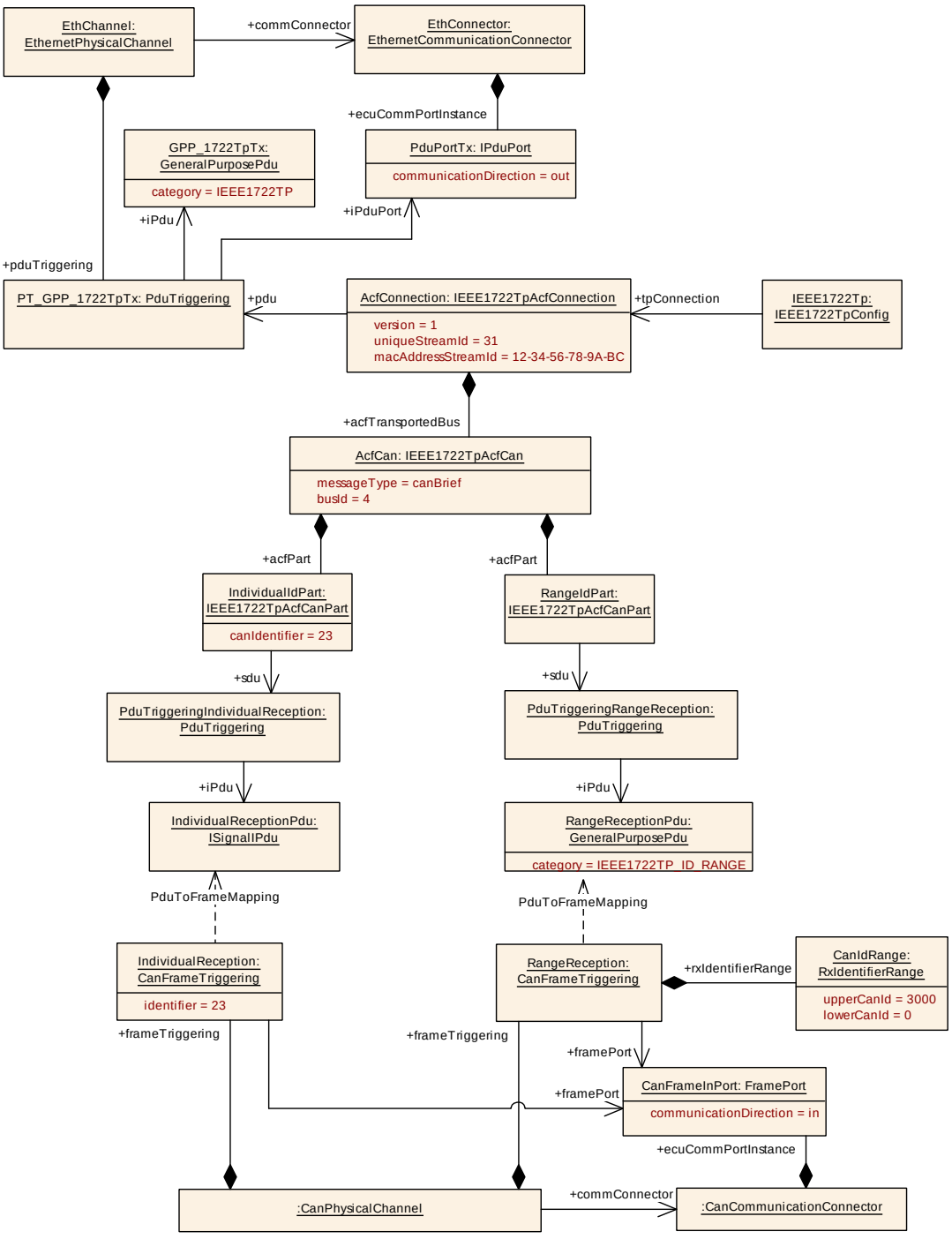

Figure 6.99: Example model of `IEEE1722TpAcfCan` transport

In the example figure 6.100 an `ISignalIPdu` is produced locally in this ECU and is sent to a CAN bus. This is configured in the lower part of the figure where the `ISignalIPdu` *LocalProducedPdu* is eventually mapped to a `CanFrameTriggering` (the mapping details are skipped here as they are not relevant for this use case). The `ISignalIPdu` is sent on the `CanPhysicalChannel` using the CAN `identifier` *7*.

Additionally to this transport on the CanPhysicalChannel the same ISignal-IPdu *LocalProducedPdu* is configured to be transmitted in the IEEE1722TpAcf-Connection *AcfConnection* using the IEEE1722TpAcfCanPart.sdu reference. The message in the IEEE1722Tp ACF stream will be qualified with the busId *4* and the canIdentifier *7*. The transport in the IEEE1722Tp ACF stream will use the IEEE1722TpAcfCanMessageTypeEnum.canBrief format.

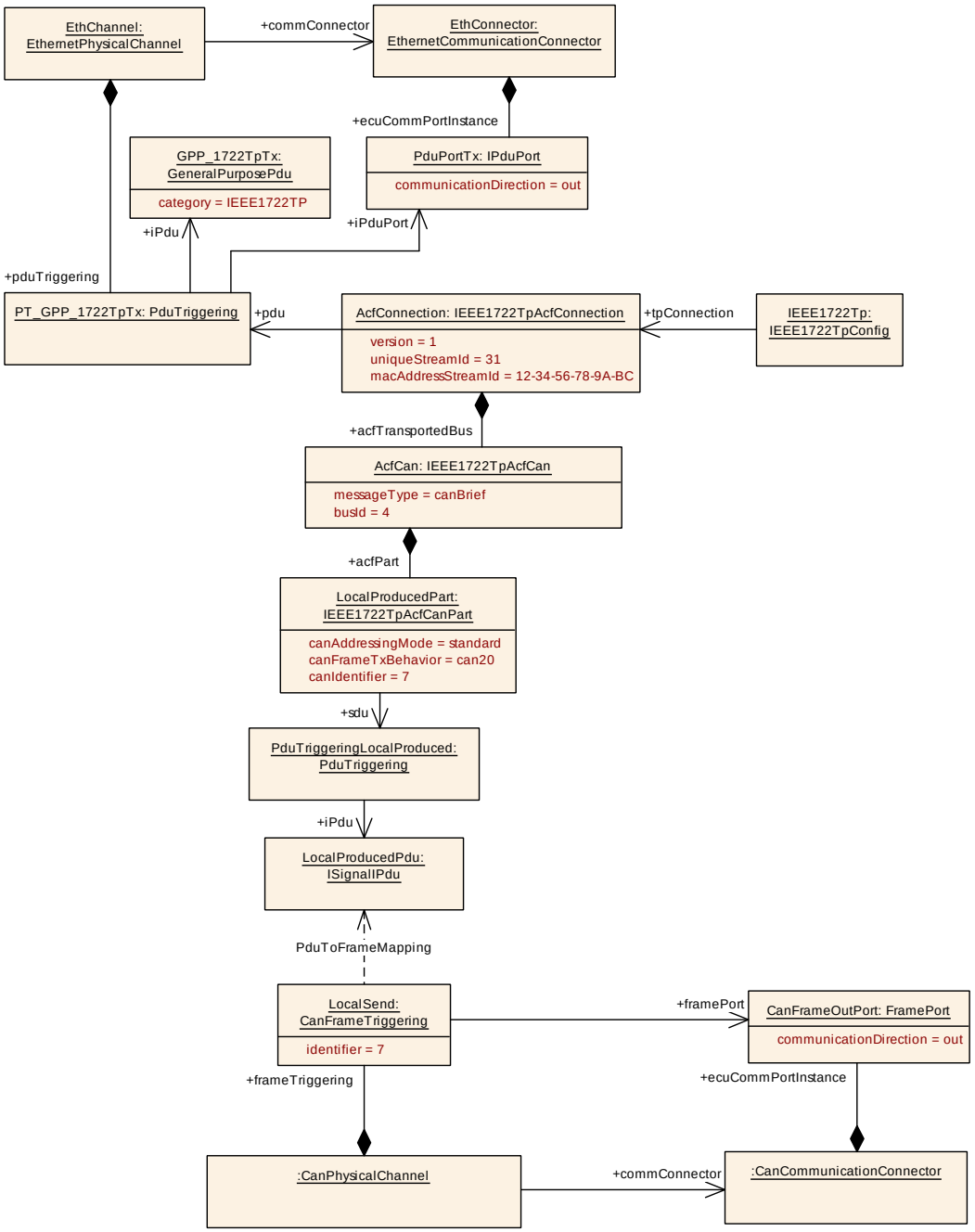

Figure 6.100: Example model of IEEE1722TpAcfCan transport of locally produced Pdu

6.8.11.8 Control Data Transport LIN

For the transport of LIN messages over an IEEE1722Tp ACF stream the definition of the IEEE1722TpAcfLin is used.

Which LIN messages are transported on an IEEE1722Tp ACF stream in the context of a dedicated IEEE1722TpAcfLin bus is defined using the IEEE1722TpAcfBus.acfPart aggregation.

[constr_3756] Consistend aggregation of IEEE1722TpAcfLinPart

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈An IEEE1722TpAcfLin shall only aggregate IEEE1722TpAcfLinParts in the role IEEE1722TpAcfLin.acfPart.⌋

[TPS_SYST_03105] Definition of LIN message transport in IEEE1722Tp ACF stream

Status: DRAFT

⌈Each LIN message to be transported on an IEEE1722Tp ACF stream is defined by having a dedicated IEEE1722TpAcfLinPart and referring from that IEEE1722TpAcfLinPart to a PduTriggering in the role sdu. This PduTriggering defines which Pdu shall be transported via the iPdu reference.⌋

[TPS_SYST_03106] Definition of LIN Id in IEEE1722Tp ACF stream

Status: DRAFT

⌈The LIN Id to be used when transporting a LIN message on an IEEE1722Tp ACF stream is defined using the IEEE1722TpAcfLinPart.linIdentifier.⌋

[constr_3757] Allowed values for IEEE1722TpAcfLinPart.linIdentifier

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The value for IEEE1722TpAcfLinPart.linIdentifier shall be in the range between 0 and 63.⌋

The LIN basic software does not support id range reception, thus every Pdu handled by the IEEE1722Tp module will be associated with a dedicated linIdentifier (no support for GeneralPurposePdu of category IEEE1722TP_ID_RANGE).

[constr_3753] Existence of attribute IEEE1722TpAcfLinPart.linIdentifier

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈For each IEEE1722TpAcfLinPart, the attribute linIdentifier shall exist.⌋

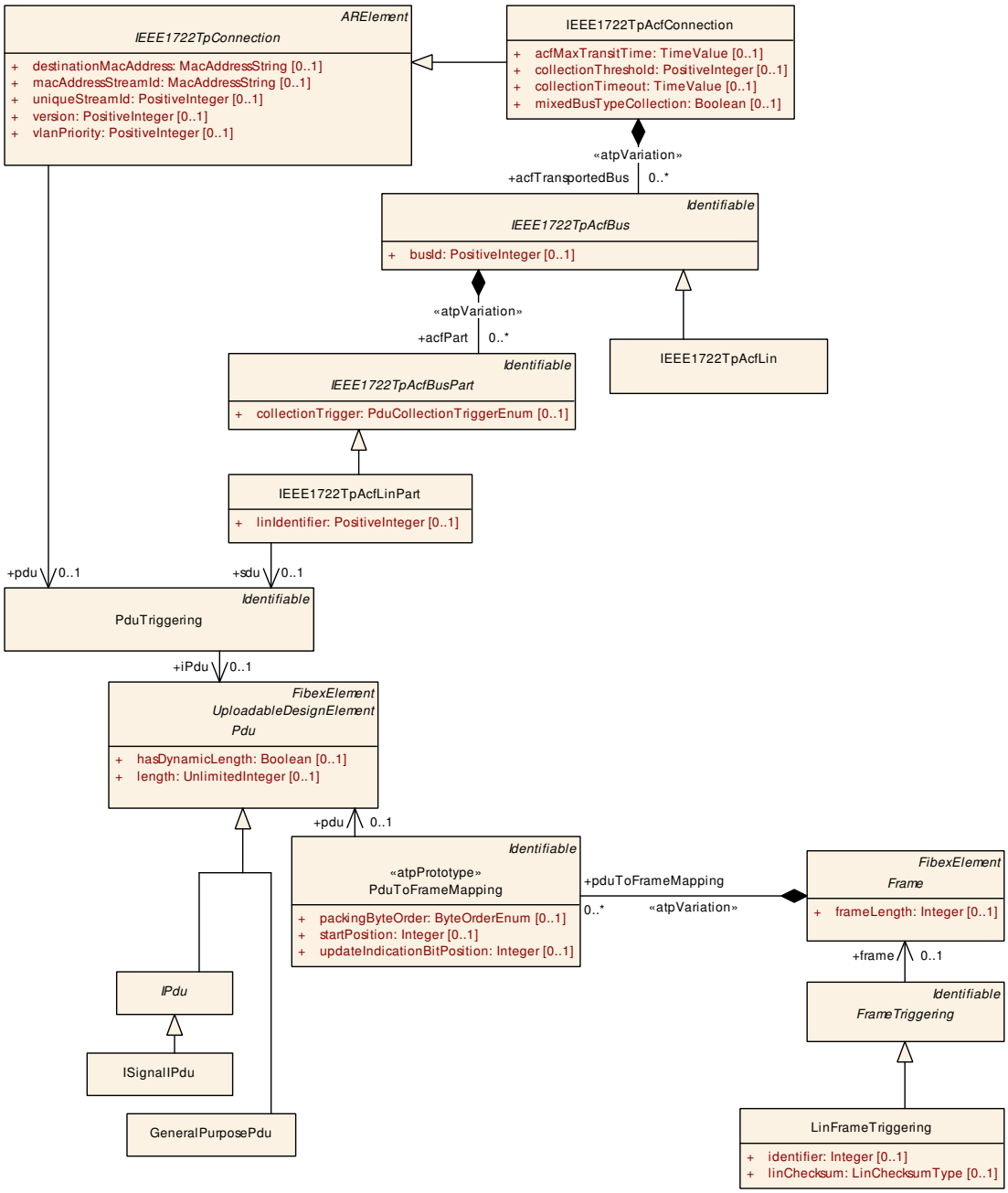

Figure 6.101: Modeling of `IEEE1722TpAcfLin`

Class	**IEEE1722TpAcfLin**			
Note	ACF IEEE1722Tp bus used for LIN transport. **Tags:** atp.Status=candidate atp.recommendedPackage=IEEE1722TpConnections			
Base	*ARObject*, *IEEE1722TpAcfBus*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IEEE1722TpAcfConnection.acfTransportedBus			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	IEEE1722TpAcfLin			
baseFrequency	PositiveInteger	0..1	attr	CRF base frequency in Hz.
frameSync Enabled	Boolean	0..1	attr	Defines whether the "fs" (frame sync) shall be enabled.
timestamp Interval	PositiveInteger	0..1	attr	CRF timestamp interval as multiple of the baseFrequency.

Table 6.300: IEEE1722TpAcfLin

Class	IEEE1722TpAcfLinPart			
Note	Definition of one LIN part transported over the IEEE1722Tp channel. **Tags:** atp.Status=candidate			
Base	*ARObject*, *IEEE1722TpAcfBusPart*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*IEEE1722TpAcfBus*.acfPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
linIdentifier	PositiveInteger	0..1	attr	Optional Lin Id defined in case the Lin Id can not be determined during runtime. **Tags:** atp.Status=candidate
sdu	PduTriggering	0..1	ref	Reference to the Pdu transported in the IEEE1722Tp channel. **Tags:** atp.Splitkey=sdu atp.Status=candidate This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.301: IEEE1722TpAcfLinPart

6.9 Network Management

The NM specification of AUTOSAR consist of a Generic Network Management Interface Module and of bus specific Network management adaptation layers (CanNm, FrNm, UdpNm, J1939Nm). The AUTOSAR Generic NM Interface module acts as a bus-independent adaptation layer between the bus-specific Network Management modules and the AUTOSAR basic software module Communication Manager. The AUTOSAR Generic NM Interface module is represented by `NmCluster`, `NmEcu`, `NmCoordinator` and `NmNode`. The bus-specific Network Management attributes are represented by `BusspecificNmEcu`. See also figure 6.102.

[constr_5032] Maximal one `NmConfig` per `System` is allowed to be defined

Imposition time: CP: `IT_SysDesc`

⌈Each `System` element is allowed to reference at most one `NmConfig` element with the `fibexElement` reference.⌋

[constr_3057] Maximal one `BusspecificNmEcu` per `NmEcu` and bus system is allowed to be defined

Imposition time: CP: `IT_SysDesc`

⌈For each `NmEcu`, at most one `BusspecificNmEcu` per bus system (FlexRay/-Can/Udp/J1939) is allowed to be defined.⌋

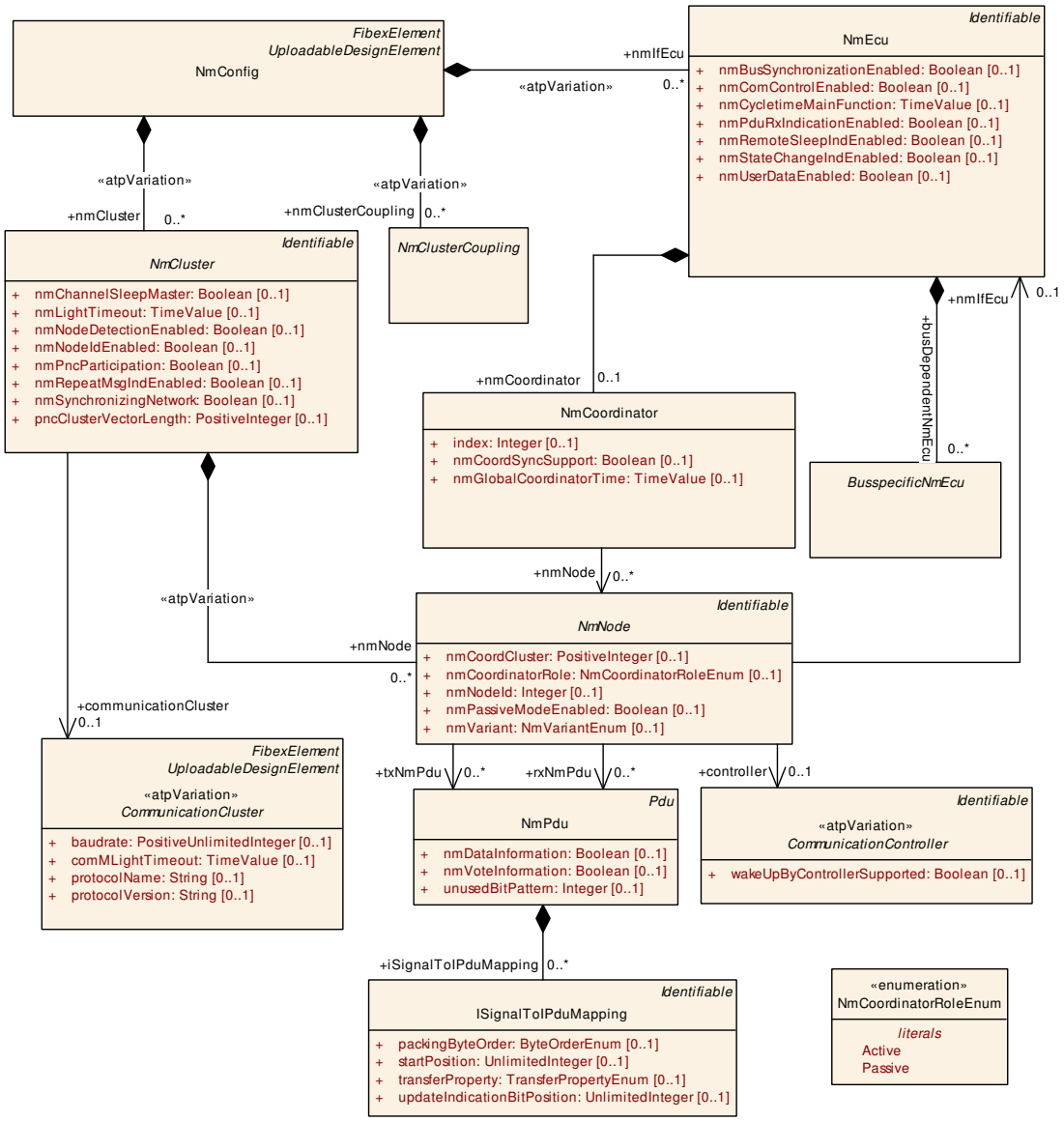

Figure 6.102: Generic Nm elements

The NmCluster contains a set of NmNodes.

The NmNodes are associated with the CommunicationController in the topology and belong to exactly one NmEcu. The reception and transmission of NmPdus is specified with the rxNmPdu and txNmPdu associations to NmPdus.

[TPS_SYST_01107] Definition of NmCoordinator ⌈An nmCoordinator is connected to two or more CommunicationClusters (via NmNodes) out of which at least two contain the requirement to shutdown synchronously.⌋

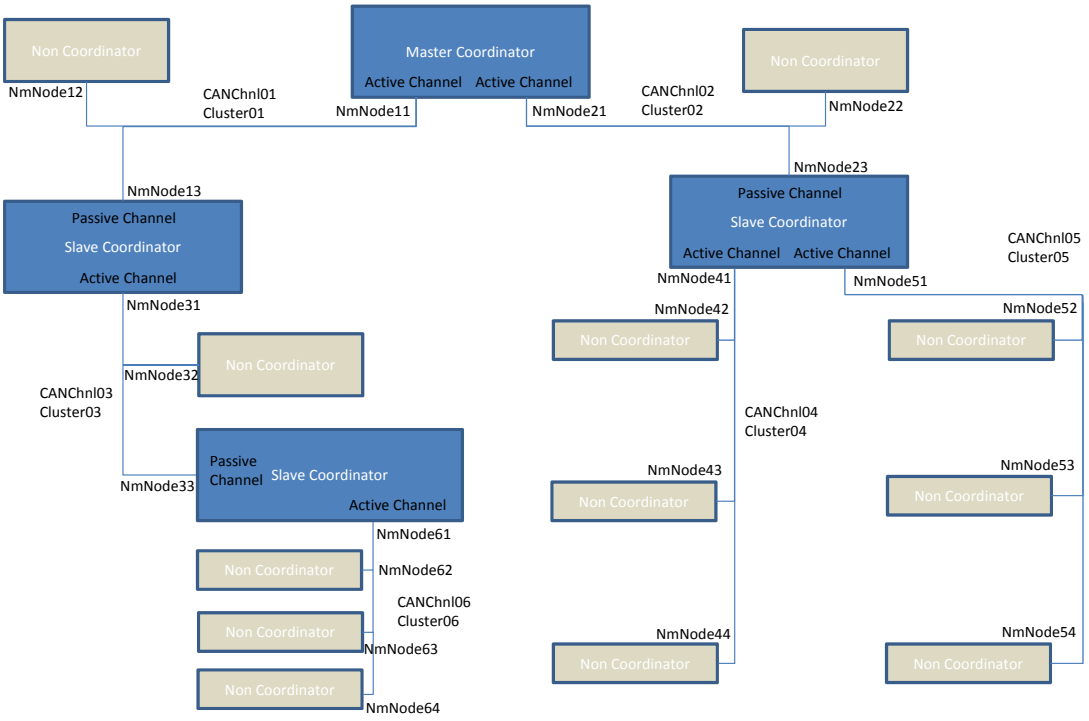

Figure 6.103: Nm Example

Figure 6.103 shows an example and the following section shows how the model shall be used:

NmCluster: Cluster01

- NmNodes:
  - NmNode11 (NmEcu1)
  - NmNode12 (NmEcu2)
  - NmNode13 (NmEcu3)

NmCluster: Cluster02

- NmNodes:
  - NmNode21 (NmEcu1)
  - NmNode22 (NmEcu4)
  - NmNode23 (NmEcu5)

NmCluster: Cluster03

- NmNodes:
  - NmNode31 (NmEcu3)
  - NmNode32 (NmEcu6)

  - NmNode33 (NmEcu7)

...

NmEcu1: NmCoordinator (MasterCoordinator)

- NmNode11 (nmCoordinatorRole: Active)
- NmNode21 (nmCoordinatorRole: Active)

NmEcu3: NmCoordinator (SlaveCoordinator)

- NmNode13 (nmCoordinatorRole: Passive)
- NmNode31 (nmCoordinatorRole: Active)

...

Class	**NmConfig**			
Note	Contains the all configuration elements for AUTOSAR Nm. **Tags:** atp.recommendedPackage=NmConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmCluster	NmCluster	*	aggr	Collection of NM Clusters atpVariation: Derived, because cluster can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmCluster, nmCluster.shortName, nm Cluster.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmCluster Coupling	NmClusterCoupling	*	aggr	Collection of NmClusterCouplings atpVariation: Derived, because NmCluster can vary. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmClusterCoupling, nmCluster Coupling.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.
nmIfEcu	NmEcu	*	aggr	Collection of NM ECUs atpVariation: Derived, because EcuInstance can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmIfEcu, nmIfEcu.shortName, nmIf Ecu.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.302: NmConfig

Class	***NmCluster*** (abstract)			
Note	Set of NM nodes coordinated with use of the NM algorithm.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmCluster, FlexrayNmCluster, J1939NmCluster, UdpNmCluster			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Cluster	CommunicationCluster	0..1	ref	Association to a CommunicationCluster in the topology description. **Tags:** atp.Splitkey=communicationCluster
nmChannel SleepMaster	Boolean	0..1	attr	This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision. This Attribute is only used by the AUTOSAR Classic Platform.
nmLightTimeout	TimeValue	0..1	attr	Defines the timeout (in seconds) after COMM_FULL_ COMMUNICATION sub-state COMM_FULL_COM_ READY_SLEEP is left.

△

Class	*NmCluster* (abstract)			
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nmNode, nmNode.shortName, nm Node.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmNode Detection Enabled	Boolean	0..1	attr	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true. This Attribute is only used by the AUTOSAR Classic Platform.
nmNodeId Enabled	Boolean	0..1	attr	Enables the source node identifier. This Attribute is only used by the AUTOSAR Classic Platform.
nmPnc Participation	Boolean	0..1	attr	Defines whether this NmCluster contributes to the partial network mechanism.
nmRepeatMsg IndEnabled	Boolean	0..1	attr	Switch for enabling the Repeat Message Bit Indication. This Attribute is only used by the AUTOSAR Classic Platform.
nm Synchronizing Network	Boolean	0..1	attr	If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_ SynchronizationPoint() at regular intervals. This Attribute is only used by the AUTOSAR Classic Platform.
pncCluster VectorLength	PositiveInteger	0..1	attr	Optionally defines the length of the PNC Request Bit Vector per CommunicationCluster (and VLAN in case of UdpNm). If not defined then System.pncVectorLength applies. Should only make the PNC Request Bit Vector shorter (or same length as defined in System.pncVectorLength).

Table 6.303: NmCluster

[constr_3771] Range of `NmCluster.nmLightTimeout`

Imposition time: CP: `IT_SysDesc`

⌈The value given for `NmCluster.nmLightTimeout` shall be in the range from 0 to 255.⌋

For the placement of user data in an `NmPdu` several boundaries and constraints have to be respected.

[TPS_SYST_03069] User data shall be defined within empty space of the `NmPdu`

⌈If user data is defined it shall be located within the not already used bounds of the `NmPdu`, specifically user data shall:

- not exceed `NmPdu.length`
- not be defined in the location where CBV and/or Nid are defined (`nmCbvPosition` and `nmNidPosition` are configured
- not be defined where the PNC Request Bit Vector is located.

⌋

[TPS_SYST_03070] User data shall be before the PNC Request Bit Vector or after the PNC Request Bit Vector ⌈If NmCluster.nmPncParticipation is set to true then user data shall be placed either:

- between CBV (control bit vector) and PNC Request Bit Vector or
- after the PNC Request Bit Vector.

There shall not be two sections with user data in one NmPdu if PNC Request Bit Vector (one before the PNC Request Bit Vector and one after the PNC Request Bit Vector).⌋

[TPS_SYST_03071] Available space of user data with PNC Request Bit Vector ⌈If NmCluster.nmPncParticipation is set to true (i.e. partial network is enabled on that NmCluster) then user data may be defined in the range:

- if the user data is mapped between the NM system bytes (CBV, Nid) and the PNC Request Bit Vector, then the size of user data range is determined by the difference between System.pncVectorOffset and the length of the system bytes (Nid, CBV).
- if the user data is mapped after the PNC Request Bit Vector, then the size of user data range is determined by the difference between NmPdu.length and the last byte position of the PNC Request Bit Vector.

⌋

[TPS_SYST_03072] Available space of user data without PNC Request Bit Vector ⌈If NmCluster.nmPncParticipation is set to false (i.e. no partial network is enabled on that NmCluster) then user data may be defined in the range between the NM system bytes (CBV, Nid) and the end of the NmPdu. The size of user data range is determined by the difference between NmPdu.length and the length of the system bytes (Nid, CBV).⌋

[constr_3044] CBV configuration in case partial network is used

Imposition time: CP: IT_SysDesc

⌈In case a partial network is used the control bit vector (CBV) shall be defined in Byte 0 of the NmPdu (nmCbvPosition = 0).⌋

[constr_3227] NmNode.nmPassiveModeEnabled setting

Imposition time: CP: IT_SysDesc

⌈NmNode.nmPassiveModeEnabled shall be set to the same value in all NmClusters with the same bus protocol in the scope of one NmEcu.⌋

[constr_3767] `CommunicationConnector.nmVariant` setting to `slavePassive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.slavePassive` if an `EthernetCommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `slaveActAsPassiveCommunicationSlave` in the referenced `EthernetCommunicationController` is set to true.⌋

[constr_3768] `CommunicationConnector.nmVariant` setting to `slaveActive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.slaveActive` if a `LinSlave` is referenced in the role `EcuInstance.commController`.⌋

[constr_3769] `CommunicationConnector.nmVariant` setting to `full`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.full` if a `CommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `nmPassiveModeEnabled` in the `NmEcu` that is referencing this `CommunicationController` is not present or is set to false.⌋

[constr_3770] `CommunicationConnector.nmVariant` setting to `passive`

Imposition time: CP: `IT_SysDesc`

⌈`CommunicationConnector.nmVariant` shall only be set to `NmVariantEnum.passive` if a `CommunicationController` is referenced in the role `CommunicationConnector.commController` and the attribute `nmPassiveModeEnabled` in the `NmEcu` that is referencing this `CommunicationController` is set to true.⌋

Class	**NmEcu**			
Note	ECU on which NM is running. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	NmConfig.nmIfEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
busDependent NmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes **Tags:** atp.Splitkey=busDependentNmEcu
ecuInstance	EcuInstance	0..1	ref	Association to an ECUInstance in the topology description. **Tags:** atp.Splitkey=ecuInstance

▽

△

Class	**NmEcu**			
nmBus Synchronization Enabled	Boolean	0..1	attr	Enables bus synchronization support.
nmComControl Enabled	Boolean	0..1	attr	Enables the Communication Control support.
nmCoordinator	NmCoordinator	0..1	aggr	Nm ECU may coordinate different clusters. **Tags:** atp.Splitkey=nmCoordinator
nmCycletime MainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the NM Interface in seconds.
nmPduRx Indication Enabled	Boolean	0..1	attr	Switch for enabling the PDU Rx Indication.
nmRemote SleepInd Enabled	Boolean	0..1	attr	Switch for enabling remote sleep indication support.
nmStateChange IndEnabled	Boolean	0..1	attr	Enables the CAN Network Management state change notification.
nmUserData Enabled	Boolean	0..1	attr	Switch for enabling user data support.

Table 6.304: NmEcu

[constr_9150] Existence of NmEcu.ecuInstance

Imposition time: CP: IT_SysDesc

⌈For each NmEcu, the reference to EcuInstance in the role ecuInstance shall exist.⌋

Class	***BusspecificNmEcu*** (abstract)			
Note	Busspecific NmEcu attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	CanNmEcu, FlexrayNmEcu, J1939NmEcu, UdpNmEcu			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.305: BusspecificNmEcu

Class	**NmCoordinator**			
Note	A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	NmEcu.nmCoordinator			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
index	Integer	0..1	attr	Identification of the NMCoordinator.

▽

△

Class	NmCoordinator			
nmCoordSync Support	Boolean	0..1	attr	Switch for enabling NmCoordinatorSync (coordination of nested busses) support.
nmGlobal Coordinator Time	TimeValue	0..1	attr	This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.
nmNode	NmNode	*	ref	reference to busses (via NmNodes) that are coordinated by the NmCoordinator. **Tags:** atp.Splitkey=nmNode

Table 6.306: NmCoordinator

Class	*NmNode* (abstract)			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmNode, FlexrayNmNode, J1939NmNode, UdpNmNode			
Aggregated by	*NmCluster*.nmNode			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
controller	Communication Controller	0..1	ref	Association to an CommunicationController in the topology description. **Tags:** atp.Splitkey=controller This Attribute is only used by the AUTOSAR Classic Platform.
nmCoord Cluster	PositiveInteger	0..1	attr	NmCoordinationCluster identification number. This Attribute is only used by the AUTOSAR Classic Platform.
nmCoordinator Role	NmCoordinatorRole Enum	0..1	attr	This attribute indicates the role the NM Coordinator will have on this channel. This Attribute is only used by the AUTOSAR Classic Platform.
nmIfEcu	NmEcu	0..1	ref	Reference to the NmEcu that contains this NmNode. (CommunicationController that is referenced by the Nm Node shall be contained in the EcuInstance that is referenced by the NmEcu). **Tags:** atp.Splitkey=nmIfEcu This Attribute is only used by the AUTOSAR Classic Platform.
nmNodeId	Integer	0..1	attr	Node identifier of local NmNode. Shall be unique in the NmCluster.
nmPassive ModeEnabled	Boolean	0..1	attr	Enables support of the Passive Mode. The passive mode is configurable per channel. This Attribute is only used by the AUTOSAR Classic Platform.
nmVariant	NmVariantEnum	0..1	attr	Defines the functionality of Network Management. **Tags:** atp.Status=obsolete
rxNmPdu	NmPdu	*	ref	receive NM Pdu. **Tags:** atp.Splitkey=rxNmPdu This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	***NmNode*** (abstract)			
txNmPdu	NmPdu	*	ref	transmit NM Pdu **Tags:** atp.Splitkey=txNmPdu This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.307: NmNode

Enumeration	**NmCoordinatorRoleEnum**
Note	Supported NmCoordinator roles.
Aggregated by	*NmNode*.nmCoordinatorRole
Literal	***Description***
Active	Coordinator which "actively" performs NmCoordinator functionality at this channel **Tags:** atp.EnumerationLiteralIndex=0
Passive	Coordinator which "passively" performs NmCoordinator functionality at this channel - used at Nm CoordinatorSync use case. **Tags:** atp.EnumerationLiteralIndex=1

Table 6.308: NmCoordinatorRoleEnum

Class	***NmClusterCoupling*** (abstract)			
Note	Attributes that are valid for each of the referenced (coupled) clusters. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	CanNmClusterCoupling, FlexrayNmClusterCoupling, UdpNmClusterCoupling			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 6.309: NmClusterCoupling

6.9.1 FlexRay Network Management

The following class tables specify the configuration parameters of FlexRay Nm.

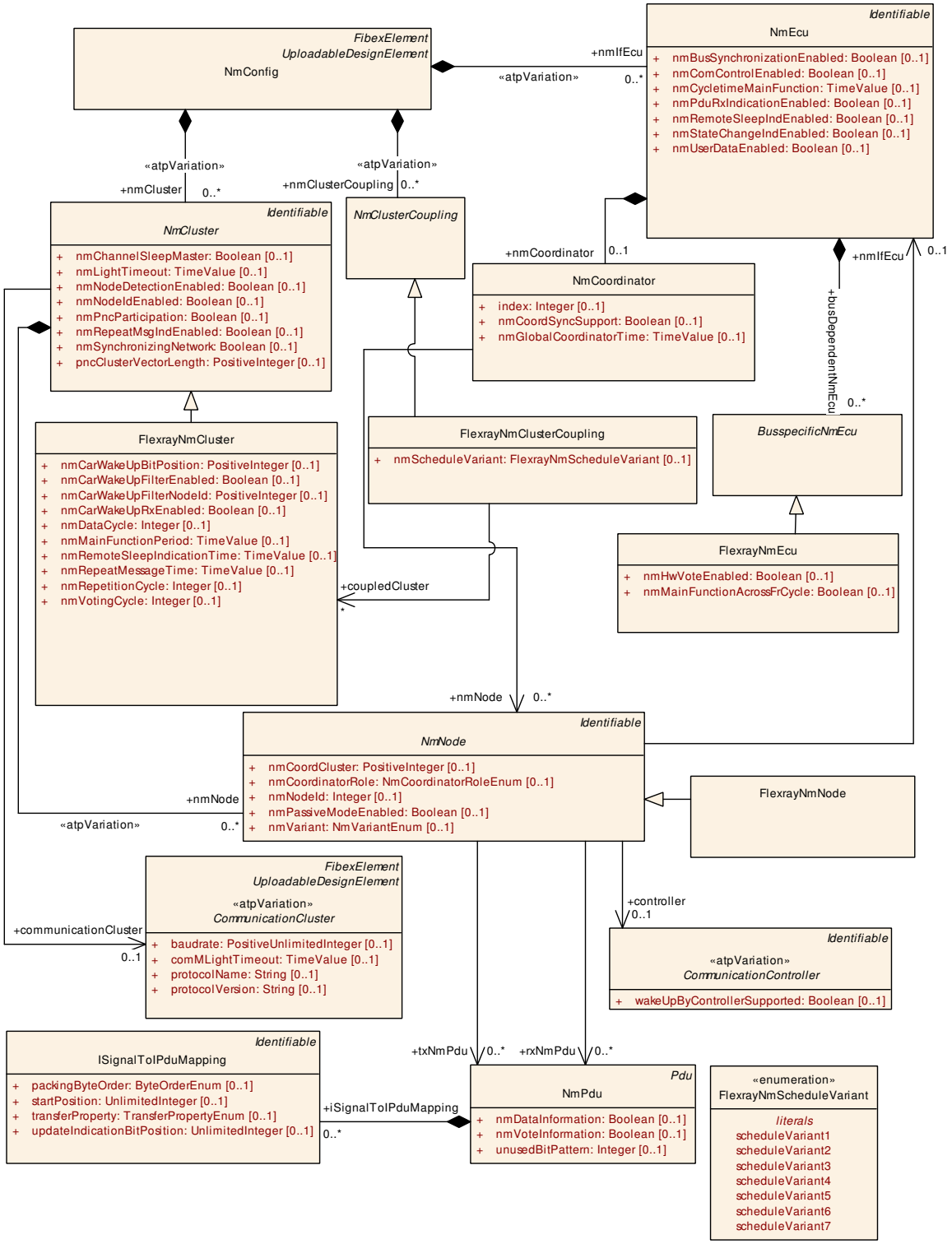

Figure 6.104: FlexRay Network Management Configuration (TransportProtocols: NmFlexRayConfiguration)

Class	**FlexrayNmCluster**			
Note	FlexRay specific NM cluster attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilter NodeId is considered as CarWakeUp request.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If Car WakeUp filtering is supported (nmCarWakeUpFilter Enabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUp RxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWake Up bit evaluation in received NmPdus.
nmDataCycle	Integer	0..1	attr	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.
nmMain FunctionPeriod	TimeValue	0..1	attr	Defines the processing cycle of the main function of FrNm module.
nmRemote SleepIndication Time	TimeValue	0..1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmRepetition Cycle	Integer	0..1	attr	Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all Flex Ray NmEcus of this FlexRayNmCluster. This value shall be an integral multiple of nmVotingCycle.
nmVotingCycle	Integer	0..1	attr	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.

Table 6.310: FlexrayNmCluster

[constr_9151] Existence of nmDataCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmDataCycle shall exist.⌋

[constr_9152] Existence of nmRemoteSleepIndicationTime

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRemoteSleepIndicationTime shall exist.⌋

[constr_9153] Existence of nmRepeatMessageTime

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRepeatMessageTime shall exist.⌋

[constr_9154] Existence of nmRepetitionCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmRepetitionCycle shall exist.⌋

[constr_9155] Existence of nmVotingCycle

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmCluster, the attribute nmVotingCycle shall exist.⌋

Class	**FlexrayNmEcu**			
Note	FlexRay specific attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusspecificNmEcu*			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmHwVote Enabled	Boolean	0..1	attr	Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.
nmMain FunctionAcross FrCycle	Boolean	0..1	attr	Parameter describing if the execution of the FrNm_Main function crosses theFlexRay cycle boundary or not.

Table 6.311: FlexrayNmEcu

Class	**FlexrayNmClusterCoupling**			
Note	FlexRay attributes that are valid for each of the referenced (coupled) FlexRay clusters. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *NmClusterCoupling*			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
coupledCluster	FlexrayNmCluster	*	ref	Reference to coupled FlexRay Clusters. **Tags:** atp.Splitkey=coupledCluster
nmSchedule Variant	FlexrayNmSchedule Variant	0..1	attr	FrNm schedule variant according to FrNm SWS.

Table 6.312: FlexrayNmClusterCoupling

[constr_9156] Existence of nmScheduleVariant

Imposition time: CP: IT_SysDesc

⌈For each FlexrayNmClusterCoupling, the attribute nmScheduleVariant shall exist.⌋

Class	**FlexrayNmNode**			
Note	FlexRay specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.313: FlexrayNmNode

[constr_9391] `FlexrayNmNode.txNmPdu` reference restriction

Imposition time: CP: `IT_SysDesc`

⌈Each `FlexrayNmNode` shall reference at most four `NmPdu`s in the role `txNmPdu` at the time when the System Description is complete.⌋

Enumeration	**FlexrayNmScheduleVariant**
Note	FrNm schedule variant according to FrNm SWS.
Aggregated by	FlexrayNmClusterCoupling.nmScheduleVariant
Literal	***Description***
scheduleVariant1	NM-Vote and NM Data transmitted within one PDU in static segment. The NM-Vote has to be realized as separate bit within the PDU. **Tags:** atp.EnumerationLiteralIndex=0
scheduleVariant2	NM-Vote and NM-Data transmitted within one PDU in dynamic segment. The presence (or non-presence) of the PDU corresponds to the NM-Vote **Tags:** atp.EnumerationLiteralIndex=1
scheduleVariant3	NM-Vote and NM-Data are transmitted in the static segment in separate PDUs. This alternative is not recommended => Alternative 1 should be used instead. **Tags:** atp.EnumerationLiteralIndex=2
scheduleVariant4	NM-Vote transmitted in static and NM-Data transmitted in dynamic segment. **Tags:** atp.EnumerationLiteralIndex=3
scheduleVariant5	NM-Vote is transmitted in dynamic and NM-Data is transmitted in static segment. This alternative is not recommended => Variants 2 or 6 should be used instead. **Tags:** atp.EnumerationLiteralIndex=4
scheduleVariant6	NM-Vote and NM-Data are transmitted in dynamic segment in separate PDUs. **Tags:** atp.EnumerationLiteralIndex=5
scheduleVariant7	NM-Vote and a copy of the CBV are transmitted in the static segment (using the FlexRay NM Vector support) and NM-Data is transmitted in the dynamic segment **Tags:** atp.EnumerationLiteralIndex=6

Table 6.314: FlexrayNmScheduleVariant

6.9.2 CAN Network Management

The following class tables specify the configuration parameters of CAN Nm.

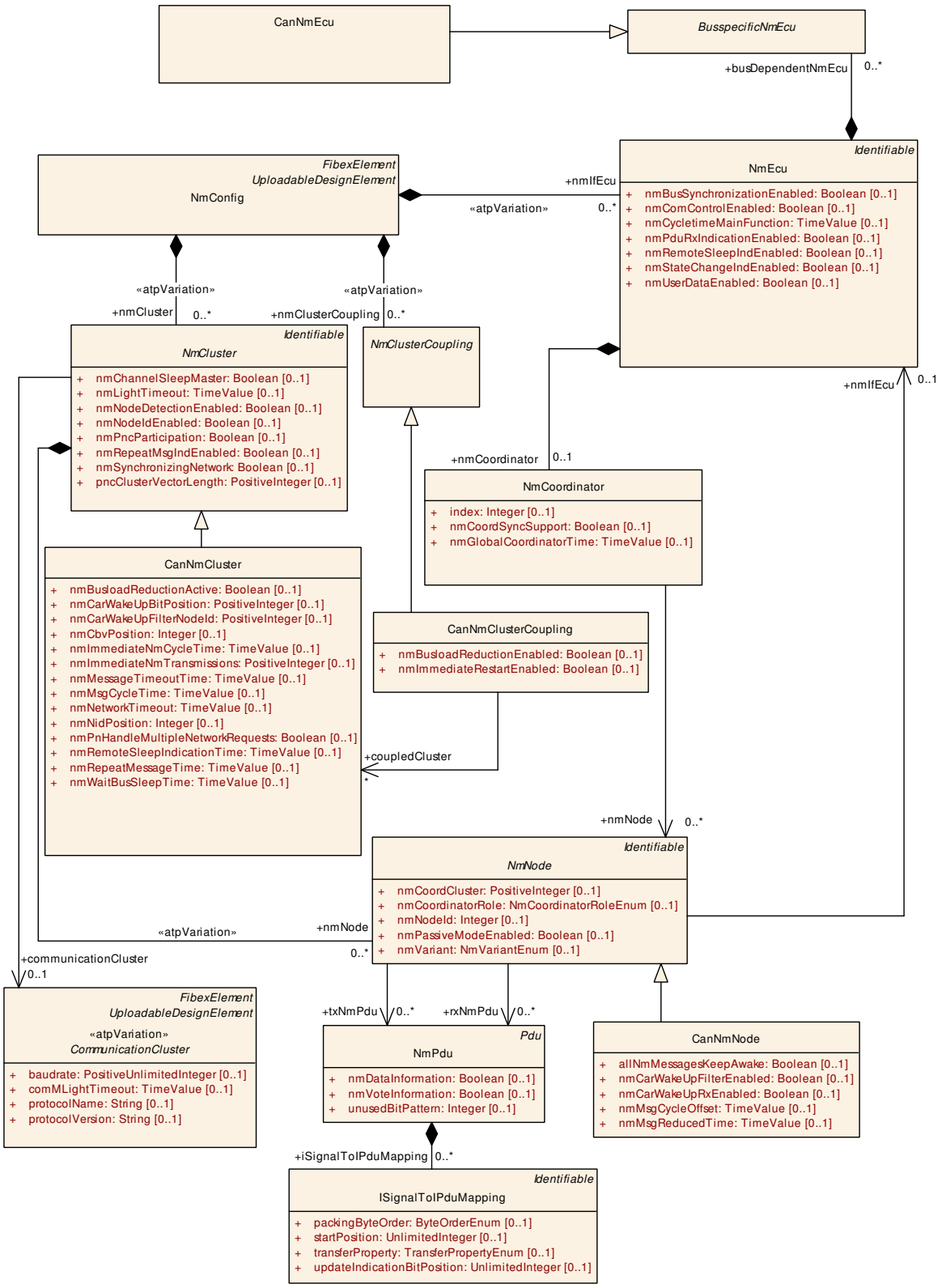

Figure 6.105: CAN Network Management Configuration (TransportProtocols: NmCanConfiguration)

Class	**CanNmCluster**			
Note	Can specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmBusload ReductionActive	Boolean	0..1	attr	It determines if bus load reduction for the respective Can Nm channel is active or not.
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering.
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
nmImmediate NmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This parameter is only valid if CanNm ImmediateNmTransmissions is greater one.
nmImmediate Nm Transmissions	PositiveInteger	0..1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessage TimeoutTime	TimeValue	0..1	attr	Timeout of an NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
nmMsgCycle Time	TimeValue	0..1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetwork Timeout	TimeValue	0..1	attr	Network Timeout for NmPdus in seconds It denotes the time how long the CanNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmPnHandle MultipleNetwork Requests	Boolean	0..1	attr	Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).
nmRemote SleepIndication Time	TimeValue	0..1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	0..1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.

Table 6.315: CanNmCluster

[constr_3069] Allowed `CanNmCluster.nmNidPosition` values

Imposition time: CP: `IT_SysDesc`

⌈If defined, the value of `CanNmCluster.nmNidPosition` shall only be set to either 0 or 1.⌋

[constr_3070] Allowed CanNmCluster.nmCbvPosition values

Imposition time: CP: IT_SysDesc

⌈If defined, the value of CanNmCluster.nmCbvPosition shall only be set to either 0 or 1.⌋

[constr_3071] CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value

Imposition time: CP: IT_SysDesc

⌈CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value.⌋

[constr_9157] Existence of nmBusloadReductionActive

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmBusloadReductionActive shall exist.⌋

[constr_9158] Existence of nmImmediateNmTransmissions

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmImmediateNmTransmissions shall exist.⌋

[constr_9159] Existence of nmMessageTimeoutTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmMessageTimeoutTime shall exist.⌋

[constr_9160] Existence of nmMsgCycleTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster the attribute nmMsgCycleTime shall exist.⌋

[constr_9161] Existence of nmNetworkTimeout

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmNetworkTimeout shall exist.⌋

[constr_9162] Existence of nmRemoteSleepIndicationTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmRemoteSleepIndicationTime shall exist.⌋

[constr_9163] Existence of nmRepeatMessageTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmRepeatMessageTime shall exist.⌋

[constr_9164] Existence of nmWaitBusSleepTime

Imposition time: CP: IT_SysDesc

⌈For each CanNmCluster, the attribute nmWaitBusSleepTime shall exist.⌋

Class	**CanNmEcu**			
Note	CAN specific attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusspecificNmEcu*			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.316: CanNmEcu

Class	**CanNmClusterCoupling**			
Note	CAN attributes that are valid for each of the referenced (coupled) CAN clusters. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *NmClusterCoupling*			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
coupledCluster	CanNmCluster	*	ref	Reference to coupled CAN Clusters. **Tags:** atp.Splitkey=coupledCluster
nmBusload Reduction Enabled	Boolean	0..1	attr	Enables busload reduction support
nmImmediate RestartEnabled	Boolean	0..1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.317: CanNmClusterCoupling

[constr_9165] Existence of nmBusloadReductionEnabled

Imposition time: CP: IT_SysDesc

⌈For each CanNmClusterCoupling, the attribute nmBusloadReductionEnabled shall exist.⌋

[constr_9166] Existence of nmImmediateRestartEnabled

Imposition time: CP: IT_SysDesc

⌈For each CanNmClusterCoupling, the attribute nmImmediateRestartEnabled shall exist.⌋

Class	**CanNmNode**
Note	CAN specific NM Node attributes.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*
Aggregated by	*NmCluster*.nmNode

▽

△

Class	**CanNmNode**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allNmMessages KeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake
nmCarWakeUp FilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported.
nmCarWakeUp RxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWake Up bit evaluation in received NmPdus.
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
nmMsg ReducedTime	TimeValue	0..1	attr	Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.

Table 6.318: CanNmNode

[constr_9388] `CanNmNode.txNmPdu` reference restriction

Imposition time: CP: `IT_SysDesc`

⌈Each `CanNmNode` shall reference at most one `NmPdu` in the role `txNmPdu` at the time when the System Description is complete.⌋

6.9.3 UDP Network Management

The UPD Nm model is similar to the Nm models of the other communication buses but there are some specific characteristics due to the modeling of EthernetCluster and EthernetPhysicalChannel (see also chapter 3.3.5).

The UdpNmCluster corresponds to one EthernetPhysicalChannel (VLAN). Therefore it is required that for each EthernetPhysicalChannel on one EthernetCluster a respective UdpNmCluster with a reference to the Ethernet-PhysicalChannel is created. All of these UdpNmClusters point to the same EthernetCluster which the EthernetPhysicalChannels are contained in.

Thus, additionally to the reference from NmCluster to the CommunicationCluster (which applies to all Nm models), there is need for an Ethernet specific reference from the UdpNmCluster to the EthernetPhysicalChannel. This allows to specify for which VLAN this UdpNmCluster applies.

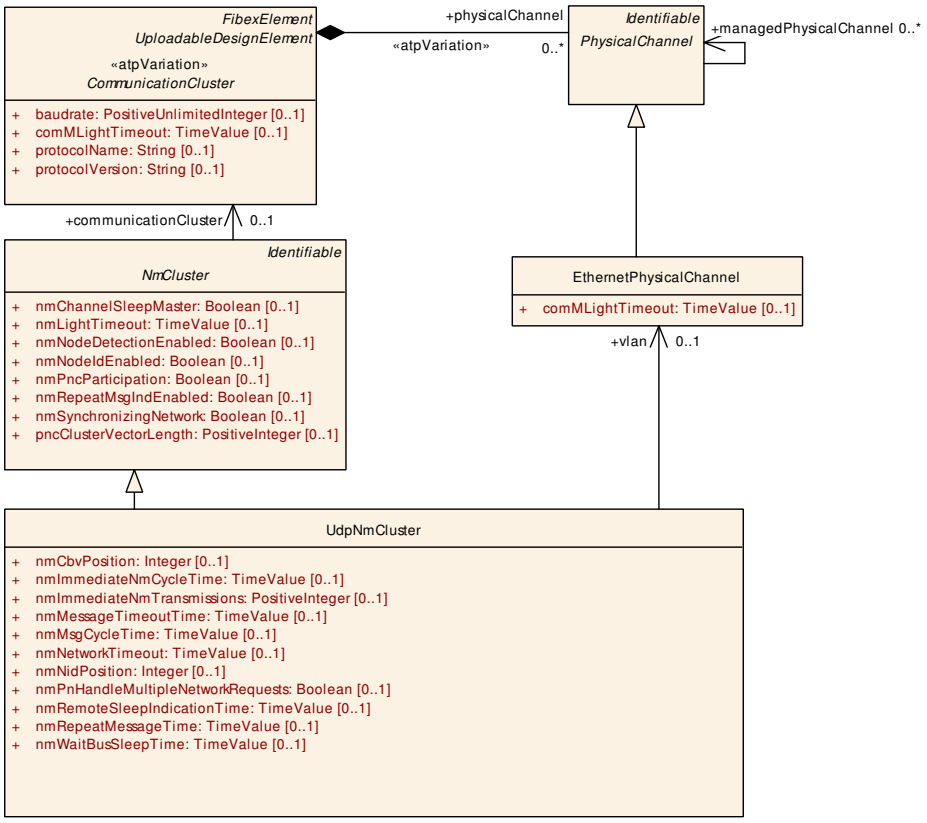

Figure 6.106: UdpNmCluster structure

The following class tables specify the configuration parameters of UDP Nm.

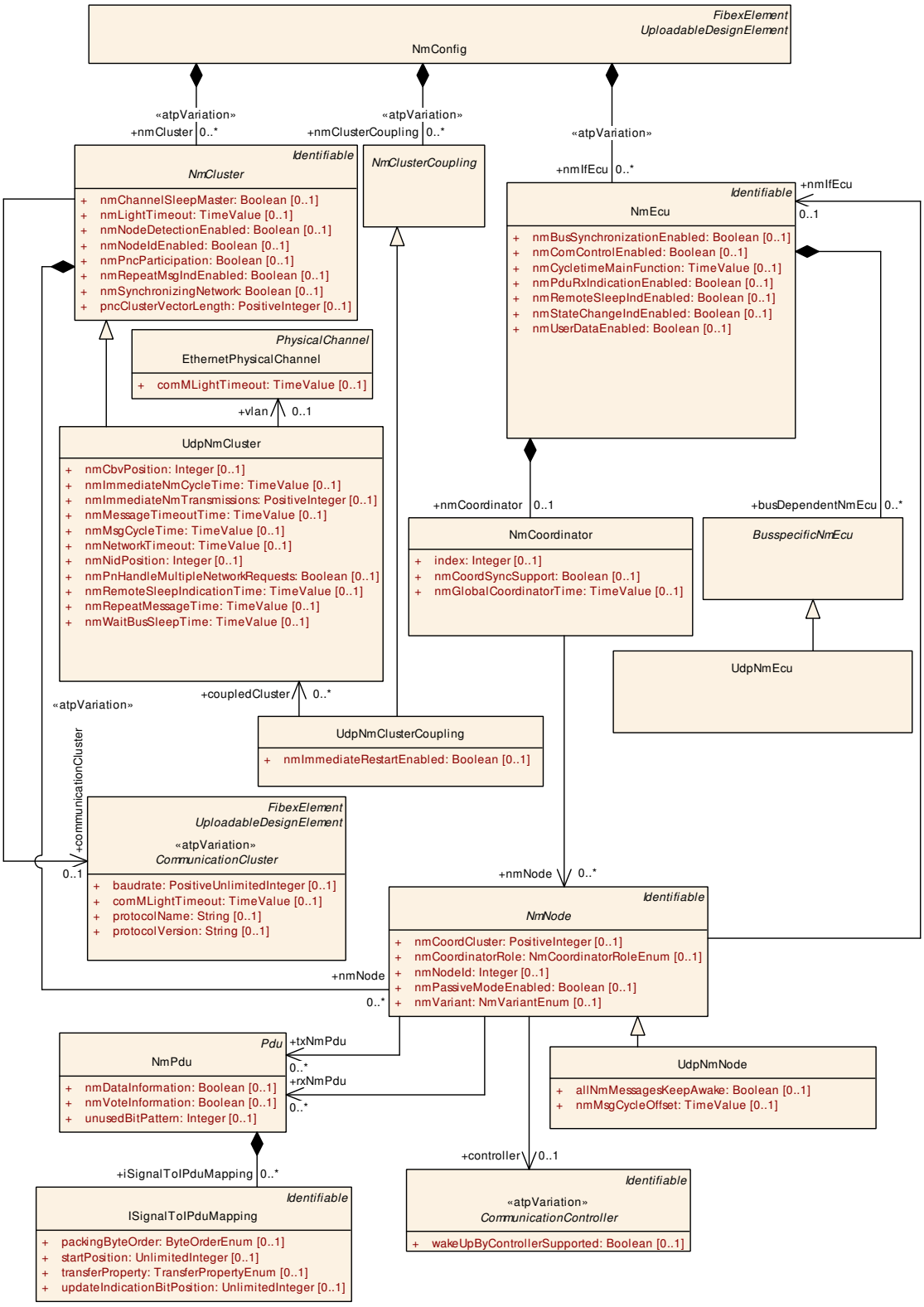

Figure 6.107: UDP Network Management Configuration (TransportProtocols: NmUdp-Configuration)

Class	**UdpNmCluster**			
Note	Udp specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
nmImmediate NmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmImmediate NmTransmissions is greater one.
nmImmediate Nm Transmissions	PositiveInteger	0..1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessage TimeoutTime	TimeValue	0..1	attr	Timeout of a NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus. This Attribute is only used by the AUTOSAR Classic Platform.
nmMsgCycle Time	TimeValue	0..1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetwork Timeout	TimeValue	0..1	attr	Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmPnHandle MultipleNetwork Requests	Boolean	0..1	attr	Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).
nmRemote SleepIndication Time	TimeValue	0..1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep. This Attribute is only used by the AUTOSAR Classic Platform.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	0..1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.
vlan	EthernetPhysical Channel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to. **Tags:** atp.Splitkey=vlan

Table 6.319: UdpNmCluster

[constr_3078] Allowed `UdpNmCluster.nmNidPosition` values

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `UdpNmCluster.nmNidPosition` shall only be set to either 0 or 1.⌋

[constr_3079] Allowed `UdpNmCluster.nmCbvPosition` values

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `UdpNmCluster.nmCbvPosition` shall only be set to either 0 or 1.⌋

[constr_3080] `UdpNmCluster.nmCbvPosition` and `UdpNmCluster.nmNidPosition` shall never have the same value

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈`UdpNmCluster.nmCbvPosition` and `UdpNmCluster.nmNidPosition` shall never have the same value.⌋

[constr_5222] Mandatory elements of `UdpNmCluster`

Imposition time: CP: `IT_SysDesc`

⌈The following attributes shall always be defined for the `UdpNmCluster`:

- `nmMsgCycleTime`
- `nmMessageTimeoutTime`
- `nmNetworkTimeout`
- `nmRemoteSleepIndicationTime`
- `nmRepeatMessageTime`
- `nmWaitBusSleepTime`
- `communicationCluster`

⌋

Class	**UdpNmEcu**			
Note	Udp NM specific ECU attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusspecificNmEcu*			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 6.320: UdpNmEcu

Class	**UdpNmClusterCoupling**			
Note	Udp attributes that are valid for each of the referenced (coupled) UdpNm clusters. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *NmClusterCoupling*			
Aggregated by	NmConfig.nmClusterCoupling			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**UdpNmClusterCoupling**			
coupledCluster	UdpNmCluster	*	ref	Reference to coupled UdpNm Clusters. **Tags:** atp.Splitkey=coupledCluster
nmImmediate RestartEnabled	Boolean	0..1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.321: UdpNmClusterCoupling

Class	**UdpNmNode**			
Note	Udp specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allNmMessages KeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.

Table 6.322: UdpNmNode

[constr_9389] UdpNmNode.txNmPdu reference restriction

Imposition time: CP: IT_SysDesc

⌈Each UdpNmNode shall reference at most one NmPdu in the role txNmPdu at the time when the System Description is complete.⌋

[constr_5223] Mandatory elements of UdpNmNode

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The following attributes shall always be defined for the UdpNmNode:

- nmMsgCycleOffset

⌋

[constr_5224] UdpNmNode.nmMsgCycleOffset < UdpNmCluster.nmMsgCycleTime

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmNode.nmMsgCycleOffset shall be smaller than the value of UdpNmCluster.nmMsgCycleTime.⌋

[constr_5225] **UdpNmCluster.nmNetworkTimeout multiple of UdpNmCluster.nmMsgCycleTime**

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmCluster.nmNetworkTimeout shall be n * UdpNmCluster.nmMsgCycleTime with n > 1.⌋

[constr_5226] **UdpNmCluster.nmRepeatMessageTime multiple of UdpNmCluster.nmMsgCycleTime**

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The value of UdpNmCluster.nmRepeatMessageTime shall be n * UdpNmCluster.nmMsgCycleTime.⌋

6.9.4 J1939 Network Management

The following class tables specify the configuration parameters of J1939 Nm.

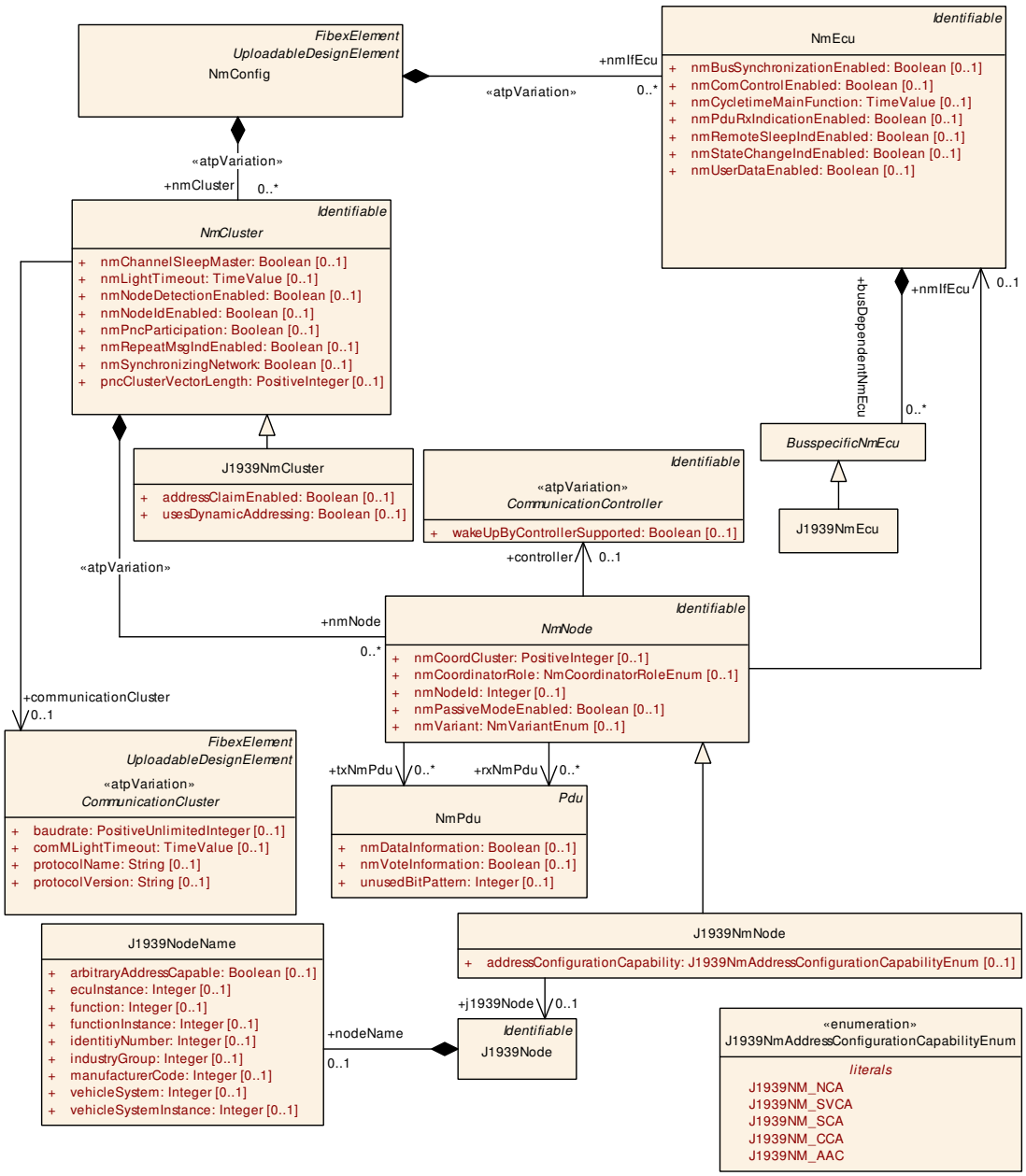

Figure 6.108: J1939 Network Management Configuration (TransportProtocols: NmJ1939Configuration)

Class	**J1939NmCluster**			
Note	J1939 specific NmCluster attributes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
addressClaim Enabled	Boolean	0..1	attr	This attribute specifies whether the J1939Nm Bsw module is used or not. If this attribute is set to false then the J1939Nm configuration shall not be derived from the system description. But even in this case the nmNodeId might still be necessary for the J1939Rm and J1939Tp.
usesDynamic Addressing	Boolean	0..1	attr	Defines whether fully dynamic address resolution according to SAE J1939-81 shall be supported on this J1939NmCluster. • True: The dynamically allocated addresses on the bus are matched at runtime to the configured addresses. • False: The addresses on the bus resemble the configured addresses.

Table 6.323: J1939NmCluster

Class	**J1939NmNode**			
Note	J1939 specific NM Node attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Aggregated by	*NmCluster*.nmNode			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
address Configuration Capability	J1939NmAddress ConfigurationCapability Enum	0..1	attr	Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).
j1939Node	J1939Node	0..1	ref	Reference to J1939Node that has a defined J1939Node Name and is assocated with a J1939Controller Application. **Tags:** atp.Splitkey=j1939Node

Table 6.324: J1939NmNode

Enumeration	**J1939NmAddressConfigurationCapabilityEnum**
Note	Defines the Address Configuration Capability options for the J1939NmNode.
Aggregated by	J1939NmNode.addressConfigurationCapability
Literal	***Description***
J1939NM_AAC	Arbitrary Address Capable CA **Tags:** atp.EnumerationLiteralIndex=4 xml.name=J-1939-NM--AAC
J1939NM_CCA	Command Configurable Address CA. **Tags:** atp.EnumerationLiteralIndex=3 xml.name=J-1939-NM--CCA

▽

△

Enumeration	**J1939NmAddressConfigurationCapabilityEnum**
J1939NM_NCA	Non-Configurable Address CA. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=J-1939-NM--NCA
J1939NM_SCA	Self-Configurable Address CA. **Tags:** atp.EnumerationLiteralIndex=2 xml.name=J-1939-NM--SCA
J1939NM_SVCA	Service Configurable Address CA. **Tags:** atp.EnumerationLiteralIndex=1 xml.name=J-1939-NM--SVCA

Table 6.325: J1939NmAddressConfigurationCapabilityEnum

[constr_9390] `J1939NmNode.txNmPdu` reference restriction

Imposition time: CP: `IT_SysDesc`

⌈Each `J1939NmNode` shall reference at most one `NmPdu` in the role `txNmPdu` at the time when the System Description is complete.⌋

Class	**J1939NmEcu**			
Note	J1939 NmEcu specific attributes. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusspecificNmEcu*			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 6.326: J1939NmEcu

6.9.4.1 J1939SharedAddressCluster

There are two ways of identifying source and target nodes in routing relations in J1939 networks (see [TPS_SYST_02107] and [TPS_SYST_02108]).

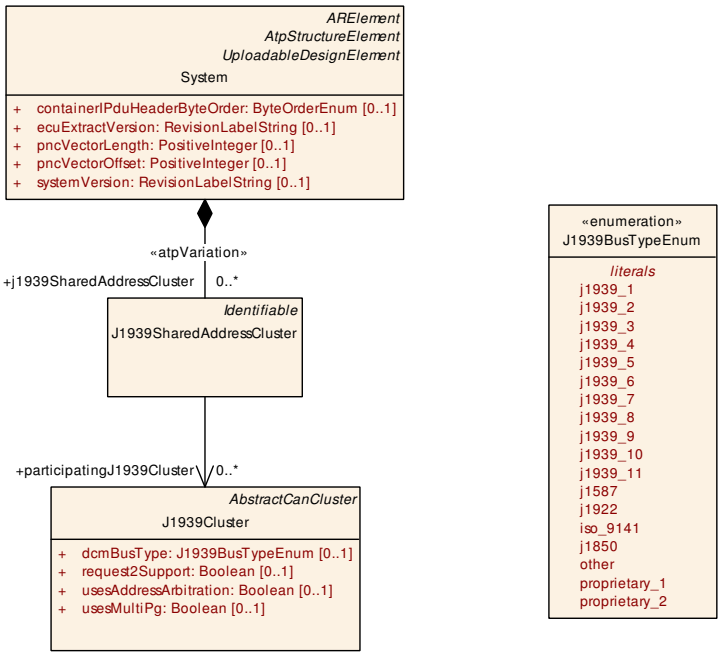

Figure 6.109: J1939SharedAddressCluster

Class	**J1939SharedAddressCluster**			
Note	This meta-class represents the ability to identify several J1939Clusters that share a common address space for the routing of messages This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	System.j1939SharedAddressCluster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
participating J1939Cluster	J1939Cluster	*	ref	This identifies the J1939Clusters that share a common address space **Tags:** atp.Splitkey=participatingJ1939Cluster

Table 6.327: J1939SharedAddressCluster

[TPS_SYST_02107] Shared address space for J1939 routing relations ⌈Address claims are routed between several CommunicationClusters independent of whether there are actual routings between individual nodes on respective CommunicationClusters. This means that the overall number of nodes in the shared CommunicationCluster cannot exceed 254, independently of the routing relations.⌋

[TPS_SYST_02108] Address proxying for J1939 routing relations ⌈The gateway claims all addresses used in routed messages on those CommunicationClusters to which the actual nodes are not connected. Thereby the address spaces are separate and only the nodes participating in a routing appear on more than one CommunicationCluster. The total number of nodes in the participating CommunicationClusters can be higher than 254, and the address arbitration is faster with less conflicts.⌋

[TPS_SYST_02109] Absence of participatingJ1939Cluster to a J1939Cluster ⌈If J1939Clusters exist that participate in a routing relation but are not referenced in the role J1939SharedAddressCluster.participating-J1939Cluster by the same J1939SharedAddressCluster then gateway shall apply the address proxying according to [TPS_SYST_02108].⌋

6.9.5 Managed Channels

6.9.5.1 Ethernet VLANs

There is the use case to transmit NM frames on one VLAN (EthernetPhysical-Channel) and the application data on different VLANs. At the same time it shall be possible to indicate that a VLAN uses Network Management although no NmPdus are defined on this channel.

A reference between PhysicalChannels is used to express such a setting: The managing PhysicalChannel that contains configured NmPdus references PhysicalChannels in the role managedPhysicalChannel.

Since the reference managedPhysicalChannel is available on the abstract PhysicalChannel element it is not only usable in case of UdpNm and VLANs but also for similar cases on other bus networks.

[constr_3479] PhysicalChannel is not allowed to be a managedPhysical-Channel and a managing PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈If a PhysicalChannel is referenced in role managedPhysicalChannel, then it shall not be the source of another managedPhysicalChannel relation.⌋

[constr_3480] PhysicalChannel shall be referenced in the role managed-PhysicalChannel only once

Imposition time: CP: IT_SysDesc

⌈A PhysicalChannel shall be referenced in the role managedPhysicalChannel only up to once.⌋

[constr_3481] UdpNmCluster is not allowed to reference a managedPhysical-Channel in the role vlan

Imposition time: CP: IT_SysDesc

⌈If an EthernetPhysicalChannel is target of a managedPhysicalChannel reference, then no UdpNmCluster shall reference this managedPhysicalChannel in the role vlan.⌋

[constr_3482] `NmCluster` is not allowed to reference a `CommunicationCluster` that aggregates a `managedPhysicalChannel`

Imposition time: CP: `IT_SysDesc`

⌈If a `PhysicalChannel`, except `EthernetPhysicalChannel`, is target of a `managedPhysicalChannel`, then the aggregating `CommunicationCluster` shall not be referenced by any `NmCluster` in the role `communicationCluster`.⌋

6.9.5.2 Ethernet tunneling through CAN XL

For the use case of tunneling Ethernet frames through CAN XL (see Ch. 3.3.1.2.2), the following specifics regarding network management configuration need to be observed:

[constr_3708] No UDP network management in case of Ethernet tunneling through CAN XL

Imposition time: CP: `IT_SysDesc`

⌈For an `EthernetPhysicalChannel` that is connected to an `EthernetCommunicationController` of category `CAN_XL` (i.e. an `EthernetPhysicalChannel` tunneled through CAN XL), no UDP network management shall be configured.⌋

Instead, the tunneled `EthernetPhysicalChannel` is managed by the CAN network management of the tunneling CAN XL network. This needs to be defined in the managed channel relationship:

[TPS_SYST_03077] Managed channel in case of Ethernet tunneling through CAN XL ⌈If a `CanPhysicalChannel` belonging to a CAN XL network is configured to tunnel Ethernet frames, then `managedPhysicalChannel`.`CanPhysicalChannel` shall refer to the `EthernetPhysicalChannel` that represents the tunneled Ethernet network.⌋

6.10 Bus Mirroring

Many communication buses in a vehicle are not directly accessible by a tester. To allow a tester to listen to the traffic on such internal communication buses the bus mirroring is introduced. The bus mirroring collects traffic from such an internal communication bus and forwards it to an intermediate destination bus or to a destination bus that is accessible by the tester.

Testers connected via CAN will receive unmodified CAN frames and LIN frames with special CAN IDs. Testers connected via Ethernet will receive a stream containing current time, identification, and content of CAN, LIN, and FlexRay frames.

On intermediate FlexRay buses, a set of PDUs is used to transport streams of mirrored frames with the same layout as on Ethernet.

[TPS_SYST_02202] Modeling of bus mirroring ⌈The `BusMirrorChannelMapping` defines the bus mirroring in which the communication traffic of the `sourceChannel` is forwarded to the `targetChannel`.⌋

Class	***BusMirrorChannelMapping*** (abstract)			
Note	This element defines a bus mirroring in which the traffic from one communication bus (sourceChannel) is forwarded to another one (targetChannel). This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	BusMirrorChannelMappingCan, BusMirrorChannelMappingFlexray, BusMirrorChannelMappingIp, BusMirrorChannelMappingUserDefined			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecuInstance	EcuInstance	0..1	ref	Ecu on which the BusMirroring is performed **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuInstance, ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
globalTimeDomain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this BusMirrorChannelMapping shall be synchonized with. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTimeDomain.globalTimeDomain, globalTimeDomain.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
mirroringProtocol	MirroringProtocolEnum	0..1	attr	This attribute defines the bus mirroring protocol that is used in the BusMirrorChannelMapping
sourceChannel	BusMirrorChannel	0..1	aggr	Defines the sourceChannel from which frames are received. **Tags:** atp.Splitkey=sourceChannel
targetChannel	BusMirrorChannel	0..1	aggr	Defines the targetChannel to which frames are forwarded. **Tags:** atp.Splitkey=targetChannel
targetPduTriggering	PduTriggering	*	ref	Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=targetPduTriggering, targetPduTriggering.pduTriggering, targetPduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
transmissionDeadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table 6.328: BusMirrorChannelMapping

Enumeration	**MirroringProtocolEnum**
Note	Eunumeration that defines the supported bus mirroring protocol options) with two literals. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*BusMirrorChannelMapping*.mirroringProtocol
Literal	***Description***
none	mirroringProtocol is not used **Tags:** atp.EnumerationLiteralIndex=1
version1	version1 of the mirroringProtocol is used **Tags:** atp.EnumerationLiteralIndex=0

Table 6.329: MirroringProtocolEnum

[constr_5384] Existence of BusMirrorChannelMapping.mirroringProtocol

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorChannelMapping, the attribute mirroringProtocol shall exist.⌋

[constr_3464] Allowed Pdu type on BusMirrorChannelMapping.targetChannel

Imposition time: CP: IT_SysDesc

⌈Each PduTriggering that is referenced by BusMirrorChannelMapping in the role targetPduTriggering is only allowed to reference a GeneralPurposeIPdu of category BUS_MIRRORING.⌋

[constr_9321] Same time base for all BusMirrorChannelMappings of one EcuInstance

Imposition time: CP: IT_SysDesc

⌈All BusMirrorChannelMappings that are referencing the same EcuInstance in the role ecuInstance shall reference the same GlobalTimeDomain in the role globalTimeDomain.⌋

Class	**BusMirrorChannel**			
Note	This element assigns a busMirrorNetworkId to the referenced channel.			
Base	*ARObject*			
Aggregated by	*BusMirrorChannelMapping*.sourceChannel, *BusMirrorChannelMapping*.targetChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
busMirror NetworkId	PositiveInteger	0..1	attr	This attribute defines the networkId of the communication channel.
channel	PhysicalChannel	0..1	ref	Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=channel, channel.physicalChannel, channel.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table 6.330: BusMirrorChannel

[constr_5494] Existence of BusMirrorChannel.busMirrorNetworkId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorChannel, the attribute busMirrorNetworkId shall exist.⌋

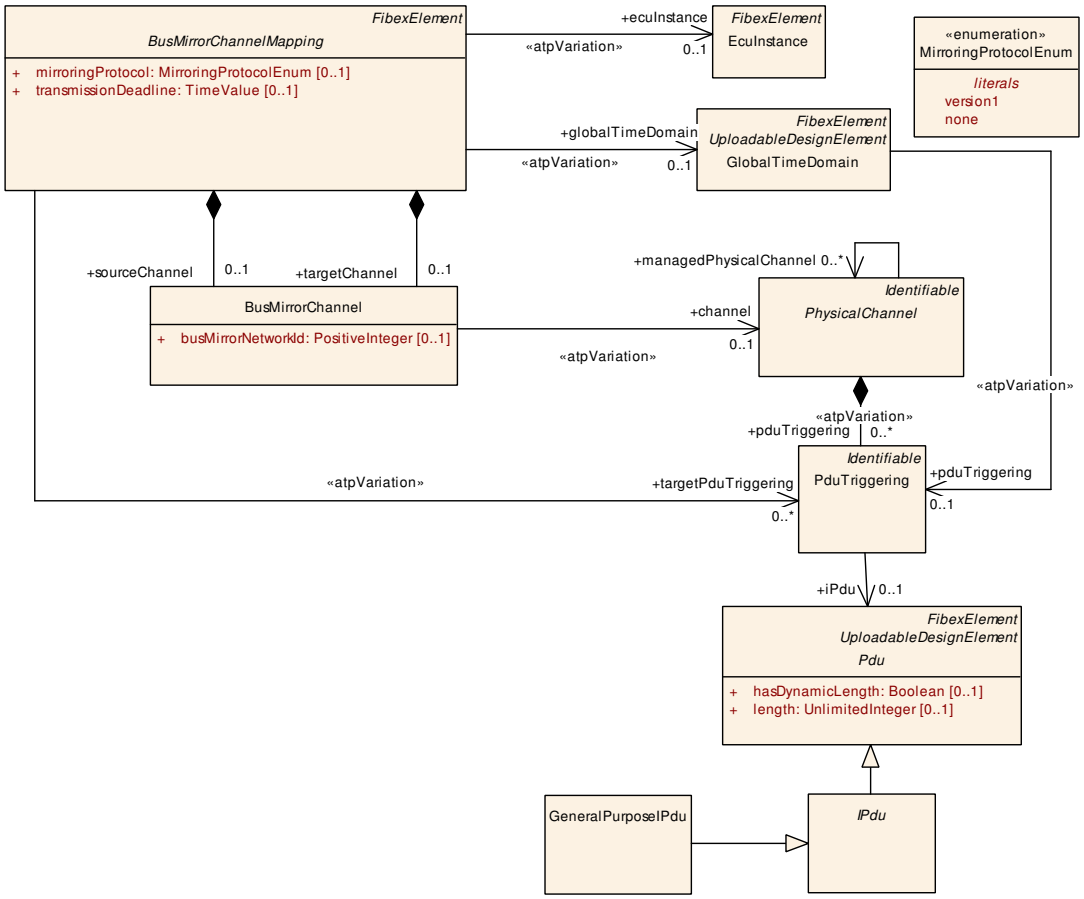

Figure 6.110: Bus mirroring

[constr_3465] Identical BusMirrorChannel.busMirrorNetworkId for BusMirrorChannels referencing the same PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈The attribute BusMirrorChannel.busMirrorNetworkId shall be identical in all BusMirrorChannels that are referencing the same PhysicalChannel in the scope of the System.⌋

[constr_3466] Unique BusMirrorChannel.busMirrorNetworkIds for each specialization of PhysicalChannel

Imposition time: CP: IT_SysDesc

⌈The attribute BusMirrorChannel.busMirrorNetworkId associated with PhysicalChannels that have the same specialization (e.g. all CanPhysicalChannels) shall have unique BusMirrorChannel.busMirrorNetworkIds within the scope of the System).⌋

[TPS_SYST_02404] Relevance of `BusMirrorChannelMapping.transmissionDeadline` ⌈The value of attribute `BusMirrorChannelMapping.transmissionDeadline` is only relevant if the value of attribute `mirroringProtocol` of the same `BusMirrorChannelMapping` is not set to `MirroringProtocolEnum.none`.⌋

6.10.1 CAN Destination Channel

[TPS_SYST_02203] BusMirroring to CAN destination channel ⌈In case of CAN to CAN and LIN to CAN the `BusMirrorChannelMappingCan` meta-class shall be used for the modeling of the bus mirroring.⌋

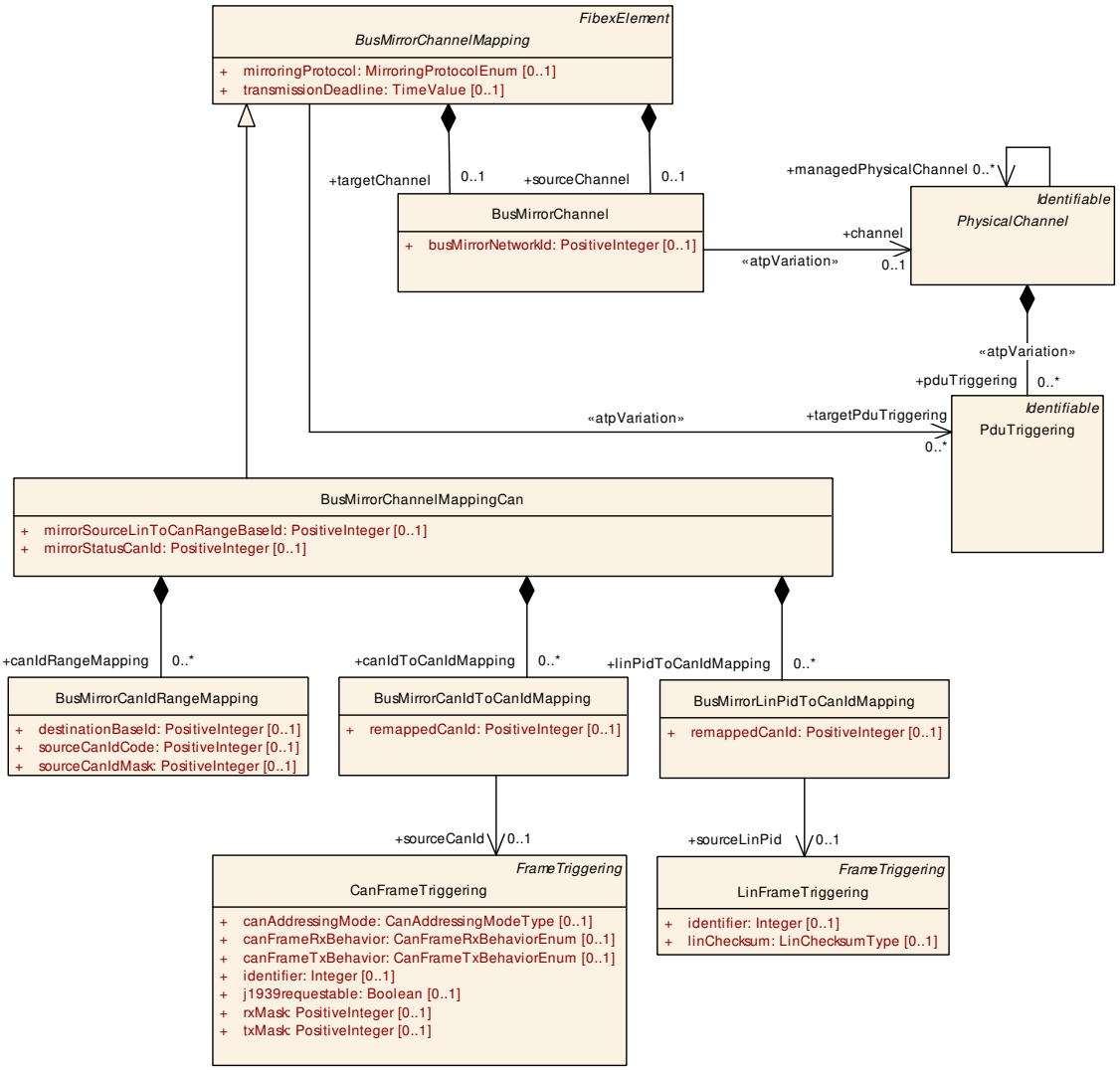

Figure 6.111: Bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel

Class	**BusMirrorChannelMappingCan**			
Note	This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
canIdRange Mapping	BusMirrorCanIdRange Mapping	*	aggr	Rules for remapping of a set of CAN IDs. **Tags:** atp.Splitkey=canIdRangeMapping
canIdToCanId Mapping	BusMirrorCanIdToCanId Mapping	*	aggr	Rules for remapping of single CanIds. **Tags:** atp.Splitkey=canIdToCanIdMapping
linPidToCanId Mapping	BusMirrorLinPidToCanIdMapping	*	aggr	Rules for remapping of single LIN Frames. **Tags:** atp.Splitkey=linPidToCanIdMapping
mirrorSourceLinToCanRange BaseId	PositiveInteger	0..1	attr	Base ID merged with the LIN frame ID to form the CAN ID. Only required when a BusMirrorChannel that refers to a LinPhysicalChannel in the role channel is referenced in the role sourceChannel.
mirrorStatus CanId	PositiveInteger	0..1	attr	CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.

Table 6.331: BusMirrorChannelMappingCan

[TPS_SYST_02399] `mirroringProtocol` for CAN destination channel with CAN 2.0 protocol ⌈The attribute `mirroringProtocol` shall always be set to none if the `BusMirrorChannelMappingCan` element references a CAN destination `PhysicalChannel` that in turn references a `CanCommunicationConnector` that in turn references a `CanCommunicationController` with aggregated `CanControllerConfiguration` or `CanControllerConfigurationRequirements` settings.⌋

In other words [TPS_SYST_02399] is only valid for CAN destination channels on which the CAN 2.0 protocol is used and not for CAN destination channels on which CanFD or CanXL are used.

[constr_3467] `CanPhysicalChannel` as destination channel of `BusMirrorChannelMappingCan`

Imposition time: CP: `IT_SysDesc`

⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingCan` shall only reference a `CanPhysicalChannel` in the role `targetChannel`.⌋

[constr_3468] `BusMirrorChannelMappingCan.targetPduTriggering` restriction

Imposition time: CP: `IT_SysDesc`

⌈`BusMirrorChannelMappingCan` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.⌋

[constr_3469] `CanFrameTriggering.txMask` setting for the destination frame

Imposition time: CP: `IT_SysDesc`

⌈The `CanFrameTriggering` of a `Frame` that contains a `Pdu` of which the `PduTriggering` is referenced by `BusMirrorChannelMappingCan` in the role `targetPduTriggering` shall set the `txMask` to 0.⌋

[constr_5051] Existence of `CanFrameTriggering.identifier` in case of bus mirror target

Imposition time: CP: `IT_SysDesc`

⌈The `CanFrameTriggering` of a `Frame` that contains a `Pdu` of which the `PduTriggering` is referenced by `BusMirrorChannelMappingCan` in the role `targetPduTriggering` shall not define an `identifier`.⌋

[constr_3470] PaddingValue used to transmit the Pdu on a Can-Fd destination bus

Imposition time: CP: `IT_SysDesc`

⌈In case that the `BusMirrorChannelMappingCan` references a `PduTriggering` in the role `targetPduTriggering` and

- the `CanFrameTriggering` of the `Frame` that contains this `targetPduTriggering` has the `canFrameTxBehavior` set to `canFd` and
- the `CanFrameTriggering` has a reference to an "out" `FramePort` (i.e. the Frame is transmitted by an Ecu on a Can-Fd destination bus) and
- the `CommunicationController` of the transmitting `EcuInstance` that is referenced via the `CommunicationConnector` by the `PhysicalChannel` on which the `targetPduTriggering` is located then the `CanControllerFdConfiguration.paddingValue` or `CanControllerFdConfigurationRequirements.paddingValue` shall have the value 0.

⌋

Class	**BusMirrorCanIdRangeMapping**			
Note	This element defines a rule for remapping a set of CAN IDs.			
Base	*ARObject*			
Aggregated by	BusMirrorChannelMappingCan.canIdRangeMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationBaseId	PositiveInteger	0..1	attr	Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.
sourceCanIdCode	PositiveInteger	0..1	attr	Value to match masked original CAN IDs.
sourceCanIdMask	PositiveInteger	0..1	attr	Mask applied to original CAN IDs before comparison.

Table 6.332: BusMirrorCanIdRangeMapping

Class	BusMirrorCanIdToCanIdMapping			
Note	This element defines a rule for remapping a single CAN ID.			
Base	ARObject			
Aggregated by	BusMirrorChannelMappingCan.canIdToCanIdMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
remappedCanId	PositiveInteger	0..1	attr	This attribute defines the CanId on the targetChannel.
sourceCanId	CanFrameTriggering	0..1	ref	This reference points to the sourceFrame with sourceCanId on the sourceChannel. **Tags:** atp.Splitkey=sourceCanId

Table 6.333: BusMirrorCanIdToCanIdMapping

Class	BusMirrorLinPidToCanIdMapping			
Note	This element defines a rule for remapping a single LIN Frame. This Class is only used by the AUTOSAR Classic Platform.			
Base	ARObject			
Aggregated by	BusMirrorChannelMappingCan.linPidToCanIdMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
remappedCanId	PositiveInteger	0..1	attr	This attribute defines the CanId on the targetChannel.
sourceLinPid	LinFrameTriggering	0..1	ref	This reference points to the sourceFrame with LIN identifier on the sourceChannel. **Tags:** atp.Splitkey=sourceLinPid

Table 6.334: BusMirrorLinPidToCanIdMapping

[constr_5495] Existence of BusMirrorCanIdRangeMapping.destinationBaseId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute destinationBaseId shall exist.⌋

[constr_5496] Existence of BusMirrorCanIdRangeMapping.sourceCanIdCode

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute sourceCanIdCode shall exist.⌋

[constr_5497] Existence of BusMirrorCanIdRangeMapping.sourceCanIdMask

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdRangeMapping, the attribute sourceCanIdMask shall exist.⌋

[constr_5498] Existence of BusMirrorCanIdToCanIdMapping.remappedCanId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorCanIdToCanIdMapping, the attribute remappedCanId shall exist.⌋

[constr_5499] Existence of BusMirrorLinPidToCanIdMapping.remappedCanId

Imposition time: CP: IT_SysDesc

⌈For each BusMirrorLinPidToCanIdMapping, the attribute remappedCanId shall exist.⌋

[constr_9319] Value of BusMirrorChannelMappingCan.mirroringProtocol

Imposition time: CP: IT_SysDesc

⌈Within the scope of a BusMirrorChannelMappingCan, if the (see [constr_3468]) PduTriggering referenced in the role BusMirrorChannelMappingCan.targetPduTriggering is in turn referenced in the role pduTriggering by a CanFrameTriggering where the aggregation in the role canXlFrameTriggeringProps exists, then the value of attribute mirroringProtocol shall only be set to MirroringProtocolEnum.version1.⌋

6.10.2 FlexRay Destination Channel

[TPS_SYST_02204] BusMirroring to FlexRay destination channel ⌈In case of CAN to FlexRay, LIN to FlexRay and FlexRay to FlexRay the BusMirrorChannelMappingFlexray meta-class shall be used for the modeling of the bus mirroring.⌋

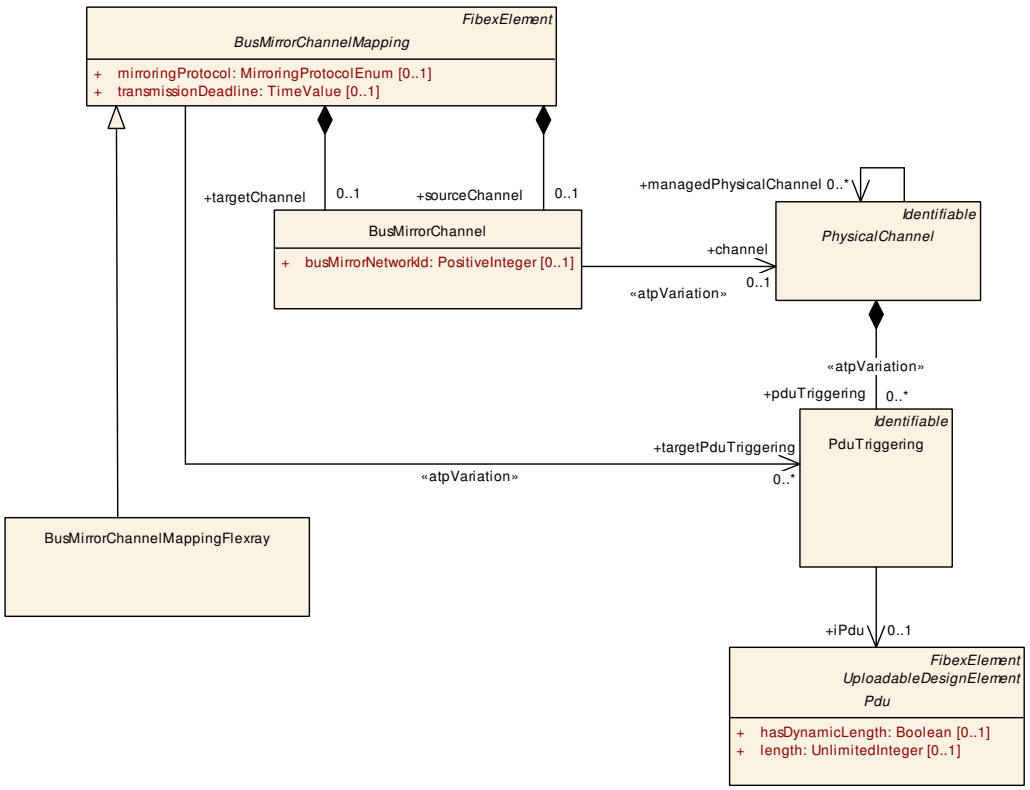

Figure 6.112: Bus mirroring between between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel

Class	BusMirrorChannelMappingFlexray			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 6.335: BusMirrorChannelMappingFlexray

[constr_3471] `FlexrayPhysicalChannel` as destination channel of `BusMirrorChannelMappingFlexray`

Imposition time: CP: `IT_SysDesc`

⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingFlexray` shall only reference a `FlexrayPhysicalChannel` in the role `targetChannel`.⌋

[constr_3472] Number of `BusMirrorChannel`s derived for one `FlexrayCluster`

Imposition time: CP: `IT_SysDesc`

⌈For each `FlexrayCluster`, only one `BusMirrorChannel` shall be derived. I.e. if both channels A and B are derived, only one of the two `FlexrayPhysicalChannel`s of one `FlexrayCluster` shall be referenced by a `BusMirrorChannel` in the `System`.⌋

[constr_3473] `BusMirrorChannelMappingFlexray.targetPduTriggering` restriction

Imposition time: CP: `IT_SysDesc`

⌈The `FlexrayFrameTriggering` of a `Frame` that contains a `Pdu` of which the `PduTriggering` is referenced by `BusMirrorChannelMappingFlexray` in the role `targetPduTriggering` shall have the `allowDynamicLSduLength` attribute set to true.⌋

[constr_9320] Value of `BusMirrorChannelMappingFlexray.mirroringProtocol`

Imposition time: CP: `IT_SysDesc`

⌈The value of attribute `BusMirrorChannelMappingFlexray.mirroringProtocol` shall only be set to `MirroringProtocolEnum.version1`.⌋

6.10.3 Ethernet Destination Channel

[TPS_SYST_02205] BusMirroring to Ethernet destination channel ⌈In case of CAN to Ethernet, LIN to Ethernet and FlexRay to Ethernet the BusMirrorChannelMappingIp meta-class shall be used for the modeling of the bus mirroring.⌋

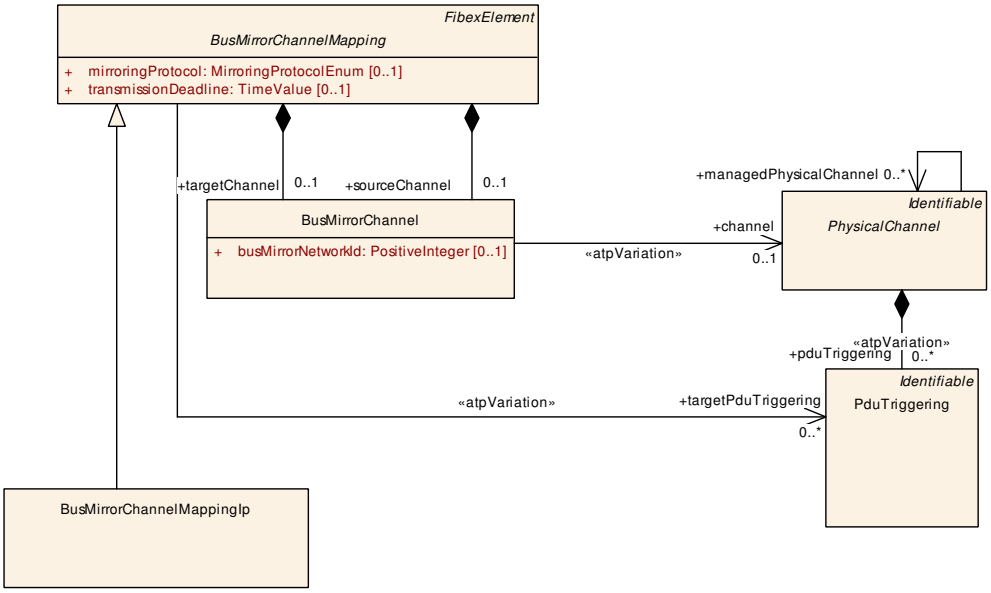

Figure 6.113: Bus mirroring between between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel

Class	**BusMirrorChannelMappingIp**			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.336: BusMirrorChannelMappingIp

[TPS_SYST_02400] mirroringProtocol for Ethernet destination channel ⌈The attribute BusMirrorChannelMappingIp.mirroringProtocol shall always be set to version1.⌋

[constr_3474] EthernetPhysicalChannel as destination channel of BusMirrorChannelMappingIp

Imposition time: CP: IT_SysDesc

⌈The BusMirrorChannel that is aggregated by BusMirrorChannelMappingIp shall only reference an EthernetPhysicalChannel in the role targetChannel.⌋

[constr_3475] `BusMirrorChannelMappingIp.targetPduTriggering` restriction

Imposition time: CP: `IT_SysDesc`

⌈`BusMirrorChannelMappingIp` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.⌋

6.10.4 User Defined Destination Channel

[TPS_SYST_02206] BusMirroring to UserDefined destination channel ⌈In case of CAN to `UserDefinedPhysicalChannel`, LIN to `UserDefinedPhysicalChannel` and FlexRay to `UserDefinedPhysicalChannel` the `BusMirrorChannelMappingUserDefined` meta-class shall be used for the modeling of the bus mirroring.⌋

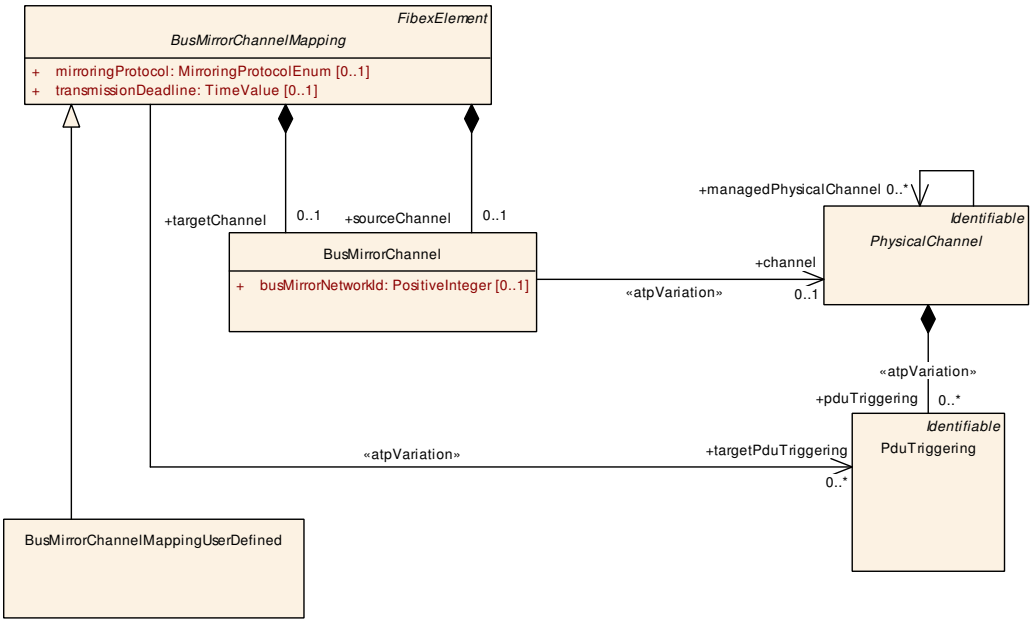

Figure 6.114: Bus mirroring between between a CAN, LIN or FlexRay sourceChannel and a UserDefined targetChannel

Class	**BusMirrorChannelMappingUserDefined**			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a User Defined targetChannel. **Tags:** atp.recommendedPackage=BusMirrorChannelMappings This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 6.337: BusMirrorChannelMappingUserDefined

[constr_3476] `UserDefinedPhysicalChannel` as destination channel of `BusMirrorChannelMappingUserDefined`

Imposition time: CP: `IT_SysDesc`

⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingUserDefined` shall only reference a `UserDefinedPhysicalChannel` in the role `targetChannel`.⌋

[constr_3477] `BusMirrorChannelMappingUserDefined.targetPduTriggering` restriction

Imposition time: CP: `IT_SysDesc`

⌈`BusMirrorChannelMappingUserDefined` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.⌋

6.11 Fan-out

AUTOSAR supports three different fan-outs:

- Signal fan-out
- Pdu fan-out
- Frame fan-out

Note that the specification in this section does not apply for Client/Server communication. The respective details are described in section 5.2.1.3.

6.11.1 Signal fan-out

A Signal fan-out can either be RTE signal fan-out or COM Signal Gateway fan-out. The details are explained in the following subchapters.

6.11.1.1 RTE signal fan-out

The RTE supports a “signal fan-out” where two or more `ISignal`s of the same `SystemSignal` are sent in multiple `ISignalIPdu`s to potentially multiple receivers.

[TPS_SYST_01109] RTE signal fan-out support for a `SystemSignal` ⌈The RTE signal fan-out from a `SystemSignal` to multiple `ISignal`s is enabled in the System Description if the following conditions apply:

- several `ISignal`s reference the `SystemSignal` in the role `systemSignal` and
- the `SystemSignal` is not referenced by a `SystemSignalGroup`
- a `DataMapping` references the `SystemSignal` and

- the DataMapping references an element in a PPortPrototype and
- the ISignals are transmitted by the EcuInstance on which the RTE is running, i.e. an ISignalTriggering exists that references the ISignal in the role iSignal and references a ISignalPort of the EcuInstance with communicationDirection out.

⌋

Figure 6.115 shows a scenario where the RTE does not need to perform a fan-out since each SystemSignal is referenced by exactly one ISignal.

Please note that in all example scenarios that are shown in this chapter the ISignalToIPduMappings shall be modeled for each ISignal or ISignalGroup. For simplicity reasons, the following diagrams leave the ISignalToIPduMappings out and just assume that they exist.

In addition for simplicity reasons the examples scenarios are showing always one OUT EcuPort and/or one IN EcuPort. But the different ISignalTriggerings are also allowed to reference own ISignalPorts that are defined on the EcuInstance.

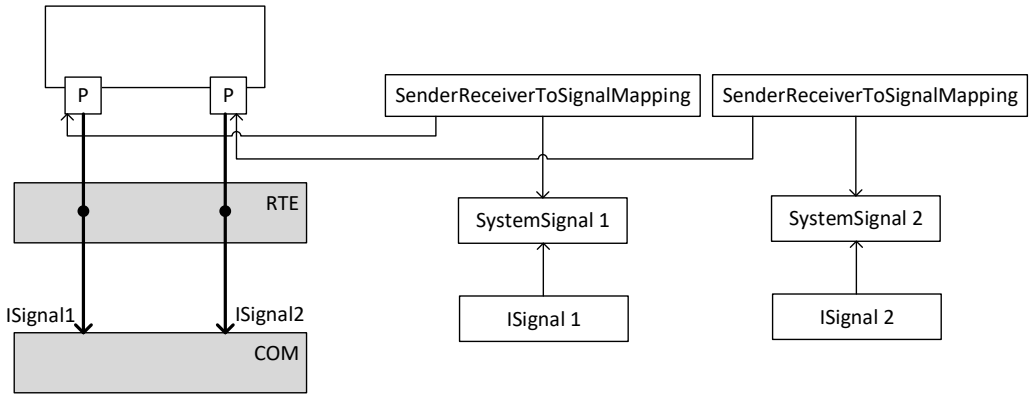

Figure 6.115: Scenario where RTE does not perform a fan-out

Figure 6.116 shows a scenario where the RTE needs to perform a fan-out since the SystemSignal is referenced by two ISignals, a SenderReceiverToSignalMapping is defined that maps a VariableDataPrototype in a SenderReceiverInterface of a PPortPrototype to the SystemSignal and the ISignals are both transmitted by the EcuInstance.

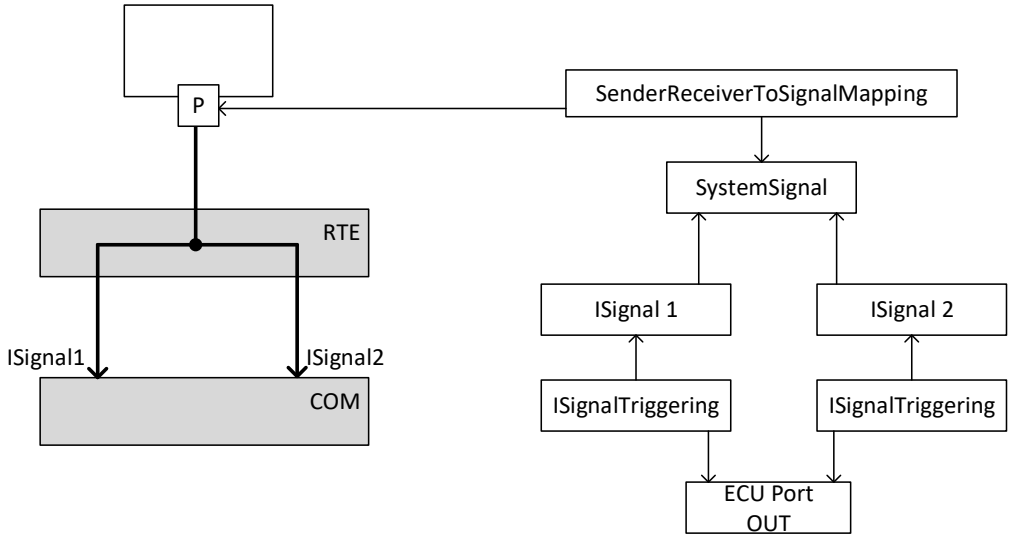

Figure 6.116: Scenario where RTE does perform a fan-out

Figure 6.117 shows a scenario where the RTE does not need to perform a fan-out since one of the `ISignal`s that is referencing the `SystemSignal` is transmitted by the `EcuInstance` and the second `ISignal` is received. In addition the `SenderReceiverToSignalMapping` that references the `SystemSignal` references a `VariableDataPrototype` in a `RPortPrototype` and therefore [TPS_SYST_01109] is not fulfilled.

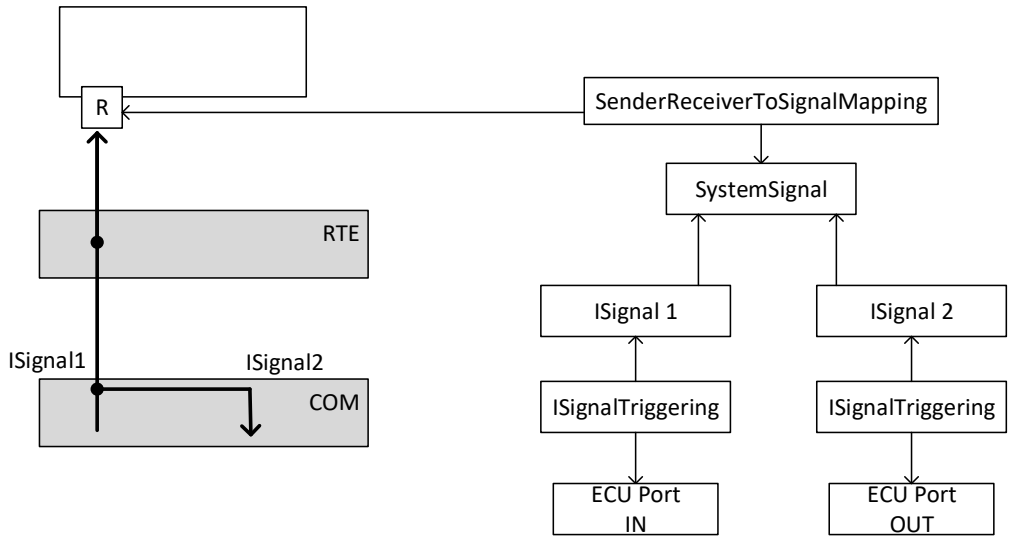

Figure 6.117: Scenario where RTE does not perform a fan-out since one ISignal is received and one is transmitted

[TPS_SYST_02309] RTE signal fan-out support for a `SystemSignalGroup` ⌈The RTE signal fan-out from a `SystemSignalGroup` to multiple `ISignalGroup`s is enabled in the System Description if the following conditions apply:

- several `ISignalGroup`s reference the `SystemSignalGroup` in the role `systemSignalGroup` and
- a `DataMapping` references the `SystemSignalGroup` and
- the `DataMapping` references an element in a `PPortPrototype` and
- each of the contained `ISignal`s of the `ISignalGroup` refers to its corresponding `SystemSignal` which in turn is part of the `SystemSignalGroup` and
- the `ISignalGroup`s are transmitted by the `EcuInstance` on which the RTE is running, i.e. an `ISignalTriggering` exists that references the `ISignalGroup` in the role `iSignalGroup` and references a `ISignalPort` of the `EcuInstance` with `communicationDirection` `out`.

⌋

In other words if two `ISignalGroup`s reference the same `SystemSignalGroup`, but one of the `ISignalGroup`s is received and one is transmitted no RTE signal fan-out is performed. Only if several `ISignalGroup`s that are transmitted reference the same `SystemSignalGroup` the RTE signal fan-out is performed.

Figure 6.118 shows a scenario where the RTE needs to perform a fan-out since the `SystemSignalGroup` is referenced by two `ISignalGroup`s, a `SenderReceiverToSignalGroupMapping` is defined that maps a `VariableDataPrototype` in a `SenderReceiverInterface` of a `PPortPrototype` to the `SystemSignalGroup` and both `ISignalGroup`s are transmitted by the `EcuInstance`.

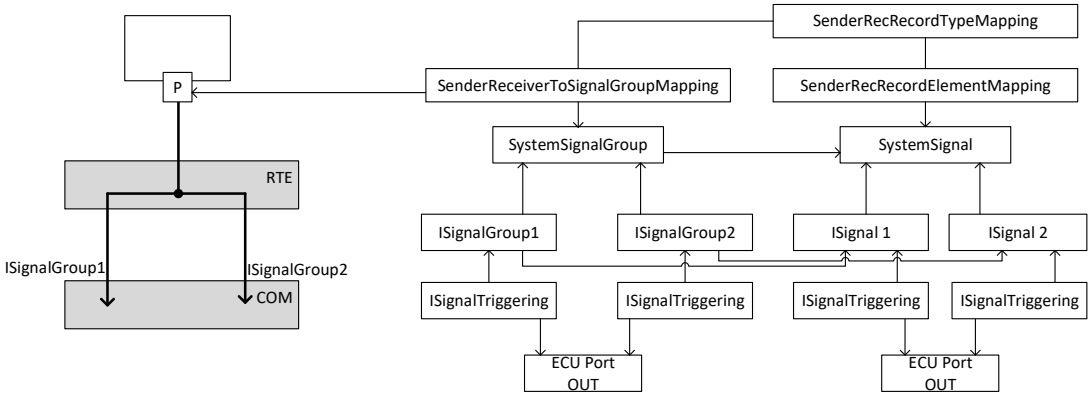

Figure 6.118: Scenario where RTE does perform a fan-out for a SystemSignalGroup

A combination of [TPS_SYST_01109] and [TPS_SYST_02309] is also supported, see also [TPS_SYST_01137] for the motivation of such a setup.

[TPS_SYST_03078] RTE signal fan-out support for a stand-alone SystemSignal out of a SystemSignalGroup ⌈The RTE signal fan-out of a SystemSignal inside a SystemSignalGroup to a stand-alone ISignal is enabled in the System Description if the following conditions apply:

- one or more ISignals reference the SystemSignal in the role systemSignal and
- the ISignals are not referenced by any ISignalGroup and
- a DataMapping references the SystemSignalGroup and
- the DataMapping references an element in a PPortPrototype and
- the ISignals are transmitted by the EcuInstance on which the RTE is running, i.e. an ISignalTriggering exists that references the ISignal in the role iSignal and references an ISignalPort of the EcuInstance with communicationDirection out.

⌋

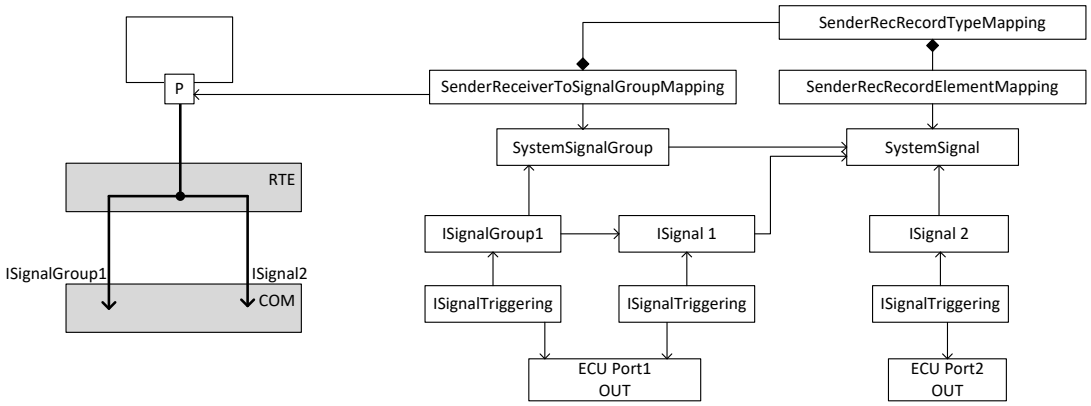

Figure 6.119: Scenario where RTE does perform a fan-out for a stand-alone SystemSignal out of a SystemSignalGroup

In the example in figure 6.119 the SystemSignal is part of a SystemSignalGroup and for that SystemSignalGroup a SenderReceiverToSignalGroupMapping is defined pointing to a PPortPrototype. Thus the application is able to send the payload of that SystemSignalGroup.

An ISignalGroup is defined referring to an ISignal (*ISignal1*) applicable for that SystemSignal. Thus [TPS_SYST_02309] applies and the SystemSignal will be sent as part of the SystemSignalGroup.

Also an ISignal applicable for the SystemSignal is defined (*ISignal2*), but *ISignal2* is not referenced by any ISignalGroup, thus it is intended to sent the SystemSignal as a standalone signal as well ([TPS_SYST_01109]).

A fan-out where different kind of transformers are applied is also supported. The straight forward example shown in Figure 6.120 illustrates a scenario where one SystemSignal is referenced by two ISignals. The SystemSignal is mapped to a single PPortPrototype with a SenderReceiverToSignalMapping.

One ISignal is referring a DataTransformation in the role dataTransformation, and that DataTransformation defines a DdsTransformationDescription as the serialization technology (see also Chapter 7.3.4). The second ISignal is referring a DataTransformation, and that DataTransformation defines a SOMEIPTransformationDescription as the serialization technology (see also Chapter 7.3.2). Similar to the example in Figure 6.116 the RTE needs to perform a fan-out utilizing two different transformation chains.

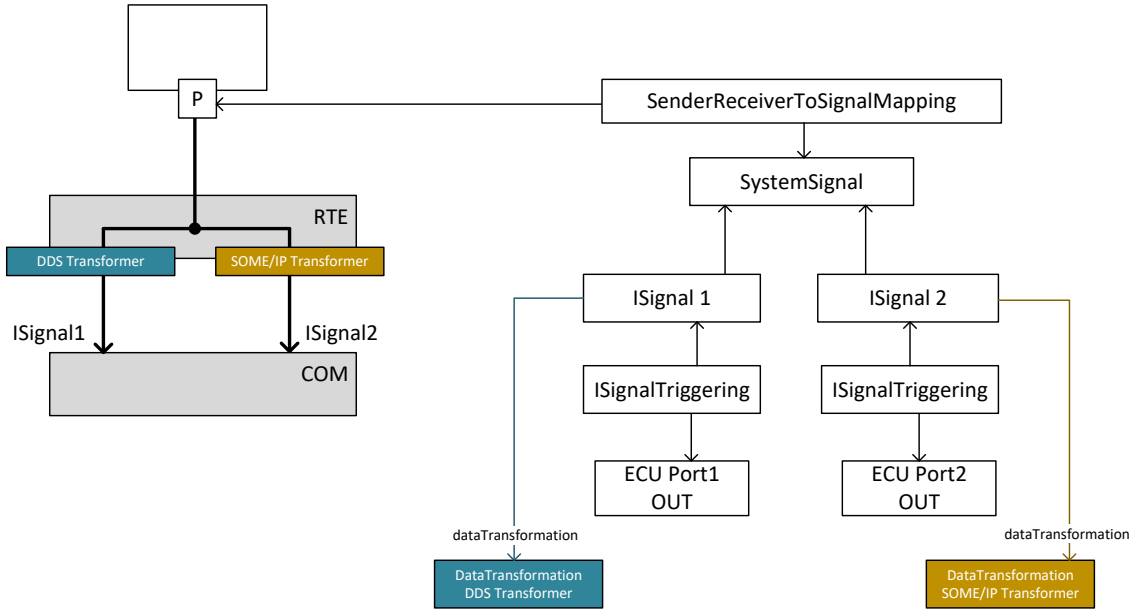

Figure 6.120: Scenario where RTE does perform a fan-out using different transformers

The COM Based Transformer on the other hand uses a different modeling approach, where the SenderReceiverToSignalGroupMapping is used to define the individual mapping of contained SystemSignals to the PortPrototype (see also Chapter 7.3.3). Thus it is not possible to directly define the fan-out on ISignalGroup level (the SOME/IP transformer does not use ISignalGroup for its definition), but the fan-out is defined on SystemSignalGroup level.

The example shown in Figure 6.121 illustrates a scenario where one application software produces data at it's PPortPrototype, and that PPortPrototype has two DataMappings defined to handle the different transformation chains: The SOME/IP transformer is defined using the SenderReceiverToSignalMapping and the COM Based Transformer is defined using the SenderReceiverToSignalGroupMapping. The RTE needs to perform a fan-out utilizing the two different transformation chains.

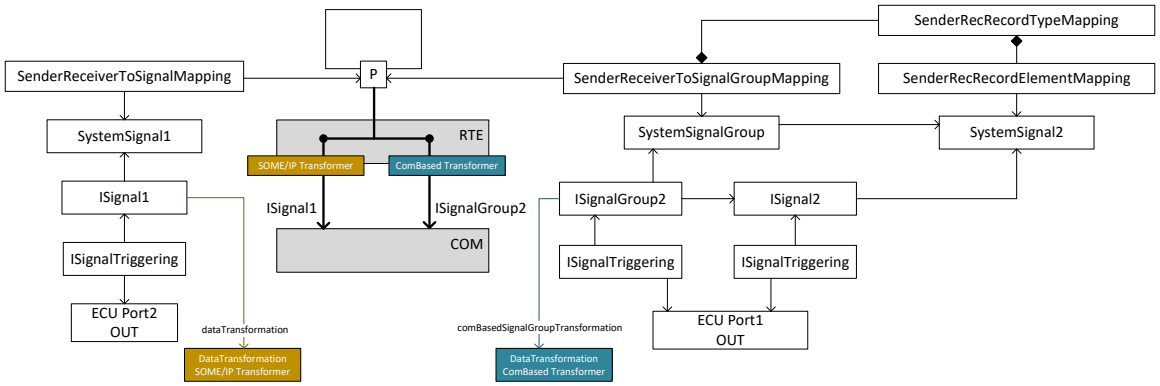

Figure 6.121: Scenario where RTE does perform a fan-out using different transformers, one being COM Based

A combined scenario is shown in Figure 6.122:

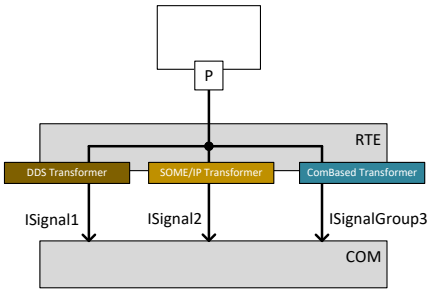

Figure 6.122: Scenario where RTE does perform a fan-out using 3 different transformers, one being COM Based

6.11.1.2 COM Signal Gateway fan-out

In Com [25] the Signal Gateway supports a fan-out where an incoming signal is routed to several destinations.

[TPS_SYST_01110] Com Signal Gateway fan-out support ⌈A `Signal Gateway` fan-out (1:n routing) is described with the definition of several `ISignalMapping`s in the `Gateway` description, which all refer to the same source `ISignalTriggering`.⌋

Note that [constr_3514] applies for the relation between `ISignalToIPduMapping` to `ISignal`.

Figure 6.123 shows a scenario where a Signal Gateway is defined in which the same source `ISignalTriggering` is mapped by two `ISignalMapping`s to two dedicated target `ISignalTriggering`s. Since a `DataMapping` is not defined for the `System-Signal` that is referenced by the received and transmitted `ISignal`s the RTE is not involved.

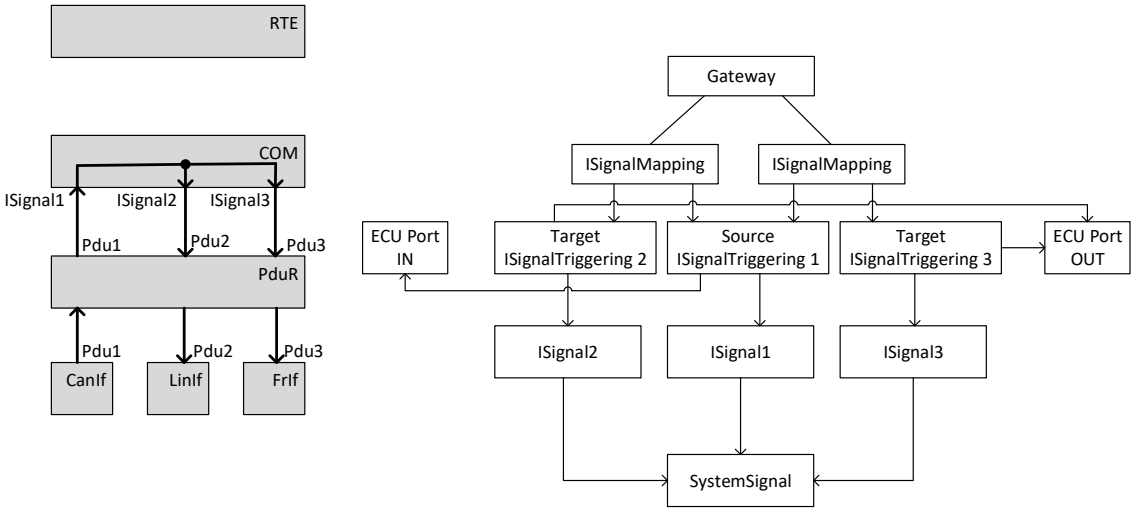

Figure 6.123: Scenario where the Com Gateway performs a fan-out and RTE is not involved

Figure 6.124 shows a scenario where a Signal Gateway is defined in which the same source `ISignalTriggering` is mapped by two `ISignalMapping`s to two dedicated target `ISignalTriggering`s. A `DataMapping` is defined for the `SystemSignal` that is referenced by the received and transmitted `ISignal`s. Therefore the RTE is involved. But the rule for the RTE signal fan-out (see [TPS_SYST_01109] does not apply since the `DataMapping` references an element in a `RPortPrototype`.

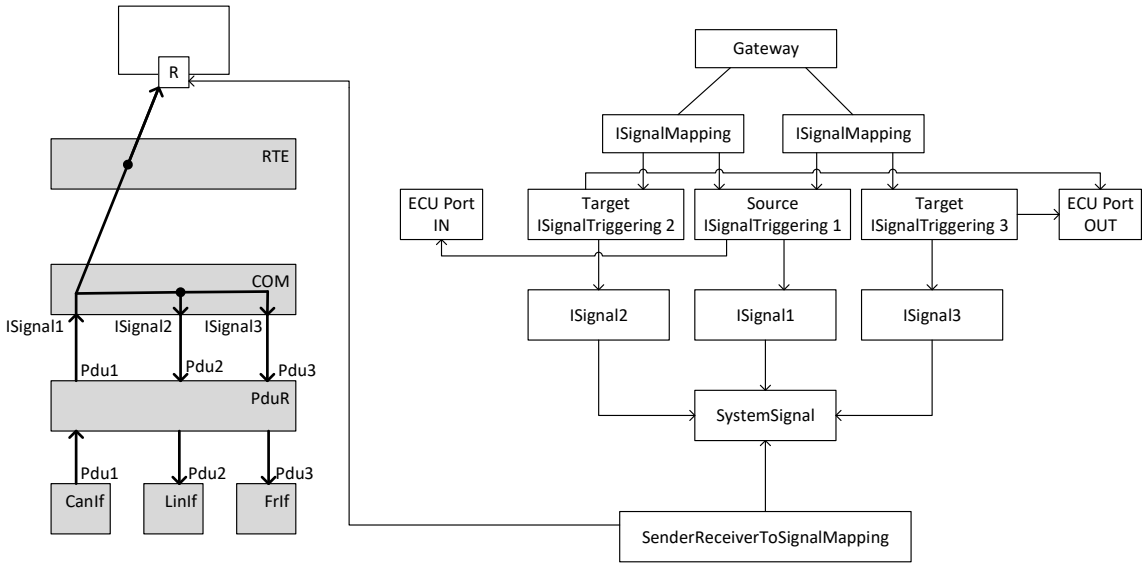

Figure 6.124: Scenario where the Com Gateway performs a fan-out and RTE is involved

6.11.2 Pdu fan-out

6.11.2.1 Pdu Router fan-out

The Pdu Router supports the "PDU fan-out" where one IPdu is sent to multiple destinations.

[TPS_SYST_01111] Pdu Router fan-out support ⌈The Pdu Router fan-out is described by several PduTriggering elements pointing to the same Pdu[3].

The sending ECU/PDU router has an output IPduPort that has the value of communicationDirection set to out and is referenced by the PduTriggering. According to the Cluster/Channel aggregation, the Pdu Router determines the clusters to use in its routing.⌋

[TPS_SYST_01112] FlexrayCluster Pdu Router interaction ⌈The following condition applies only in case of FlexRay on the same FlexrayCluster if two PduTriggerings refer to the same Pdu: this Pdu shall only be sent once to the FlexRay Interface. In other words the Pdu Router sends only one Pdu Transmission request to the FlexRay Interface.⌋

Several examples of various Pdu Router configurations are illustrated in the following section. In chapter 8.2.1 further examples of fan-out in combination with Pdu gateways are defined.

6.11.2.1.1 Pdu Router simple fan-out

The example in figure 6.125 shows a simple Pdu router fan-out where an ISignalIPdu is produced by the Com module and forwarded to the PduR. In the PduR a fan-out to two PhysicalChannels is performed.

For a simplified figure just one ISignal is shown to be part of the ISignalIPdu. Of course it is possible to define several ISignal being part of an ISignalIPdu.

[3] AUTOSAR Layered Architecture [39] defines which Pdu types are routed by the Pdu Router

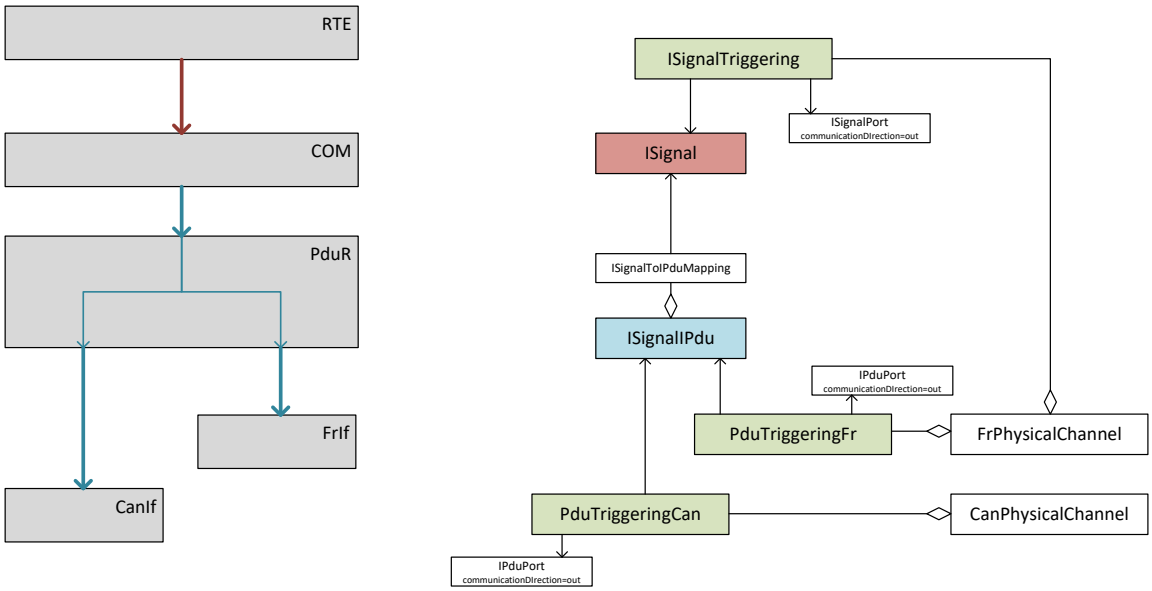

Figure 6.125: Pdu Router simple fan-out example

6.11.2.1.2 Pdu Router SecOC Pdu fan-out

In the example figure 6.126 the PduR takes an `ISignalIPdu` from the Com module and hands it over to the SecOC module. This is a plain forwarding operation (no fan-out involved here).

In the SecOC module a `SecuredIPdu` is created and handed back to the PduR. Then the PduR performs a fan-out to send the `SecuredIPdu` to two `PhysicalChannel`s.

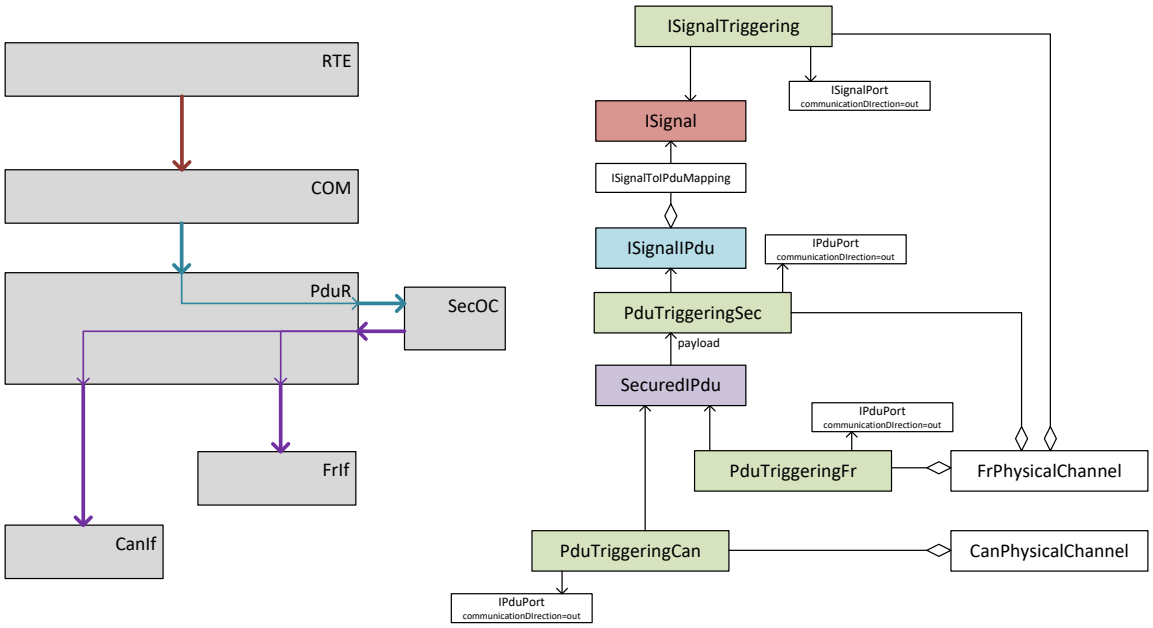

Figure 6.126: Pdu Router SecOC Pdu fan-out example

6.11.2.1.3 Pdu Router Container Pdu fan-out

In the example figure 6.127 the PduR takes one of several `ISignalIPdu`s from the Com module and hands them over to the IpduM module. This is a plain forwarding operation (no fan-out involved here).

In the IpduM module a `ContainerIPdu` is created and handed back to the PduR. Then the PduR performs a fan-out to send the `ContainerIPdu` to two `PhysicalChannel`s.

For a simplified figure just one `ISignal` is shown to be part of the *ISignalIPduB*. Of course also *ISignalIPduA* will have one or more `ISignal`s defined as well.

Also *PduTriggeringContA* being owned by the *FrPhysicalChannel* is not shown, but will be part of the example model. As well as the reference from *PduTriggeringContA* to an `IPduPort` is not shown.

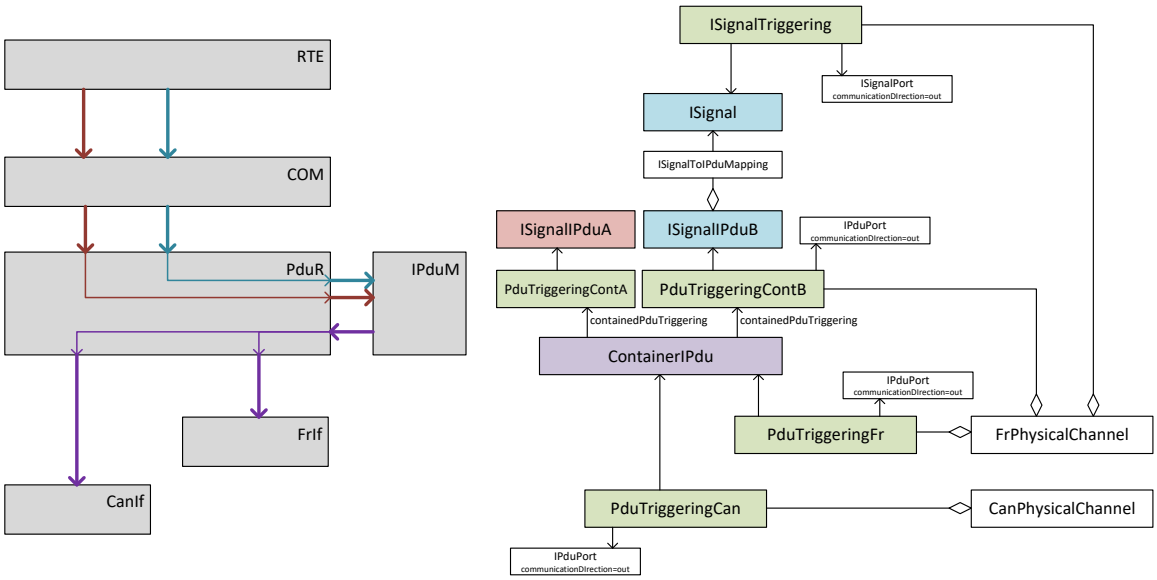

Figure 6.127: Pdu Router Container Pdu fan-out example

6.11.2.1.4 Pdu Router IPdu fan-out and SecOC / non SecOC

In the example figure 6.128 the PduR takes an `ISignalIPdu` from the Com module and performs a fan-out on that `ISignalIPdu`.

On one side the `ISignalIPdu` is handed over to the CanIf for direct transmission on the `CanPhysicalChannel`. The *PduTriggeringSec* is owned by the `CanPhysical-Channel` and has a corresponding *IPduPort* defined. Note that there is an alternative modeling approach discussed in section 6.1.1.

On the other side the `ISignalIPdu` is handed to the SecOC module. In the SecOC module a `SecuredIPdu` is created and handed back to the PduR. Then the PduR hands that `SecuredIPdu` to the FrIf for transmission on the `FlexrayPhysical-Channel`.

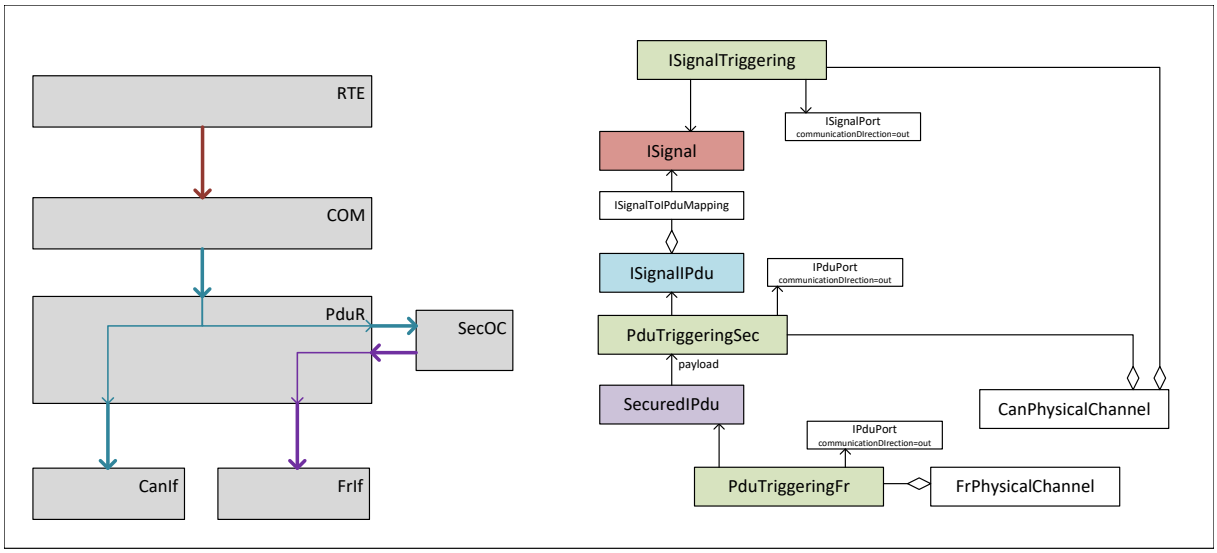

Figure 6.128: Pdu Router IPdu fan-out and SecOC / non SecOC example

6.11.2.2 Flexray Interface fan-out

The Flexray interface supports a fan-out where one `Pdu` is mapped into more than one frame on the same `CommunicationCluster`.

[TPS_SYST_01113] FlexRay Interface fan-out support ⌈The redundant transmission in the FlexRay Interface in the static segment is described by

- one `FlexrayFrameTriggering` on each `PhysicalChannel`
- both `FlexrayFrameTriggering`s refer to the same `FlexrayFrame` with the same `Pdu`
- each `FlexrayFrameTriggering` aggregates the same number of `FlexrayAbsolutelyScheduledTiming`s
- for every `FlexrayAbsolutelyScheduledTiming` on one `PhysicalChannel` a corresponding `FlexrayAbsolutelyScheduledTiming` with identical values shall be defined on the other `PhysicalChannel`

⌋

If the fan-out is specified between different FlexRay channels of the same cluster it shall be handled by the FlexRay Interface.

The Flexray Interface does NOT handle fan-out/in between different clusters.

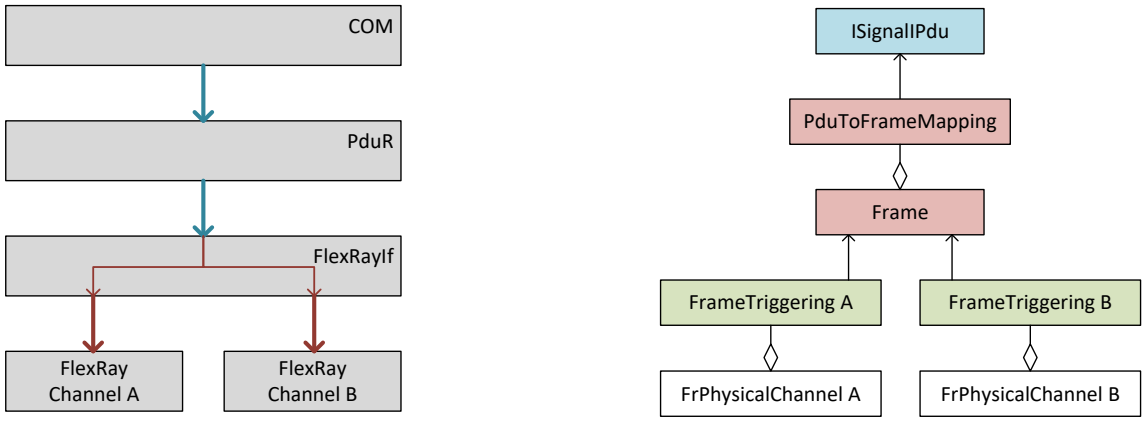

Figure 6.129: Bus Interface fan-out

6.11.2.3 Socket Adapter fan-out

The AUTOSAR Socket Adapter module [42] is capable to perform a PDU fan-out by itself.

In case of socket configuration for Service Discovery only the maximum number of socket connections is specified in the system description: `AbstractService-Instance` refers to an `ApplicationEndpoint` with `maxNumberOfConnections`.

The example in figure 6.130 illustrates a `ProvidedServiceInstance` with a reference to an `ApplicationEndpoint.maxNumberOfConnections` set to 3. The Socket Adapter will be configured with 3 socket connections to be used for this `ProvidedServiceInstance`.

The payload used in this example is an `ISignalIPdu` which gets assigned a SOME/IP `headerId` via the aggregated `SoConIPduIdentifier`.

Depending on the actually subscribed clients the number of used socket connections will depend on the runtime situation.

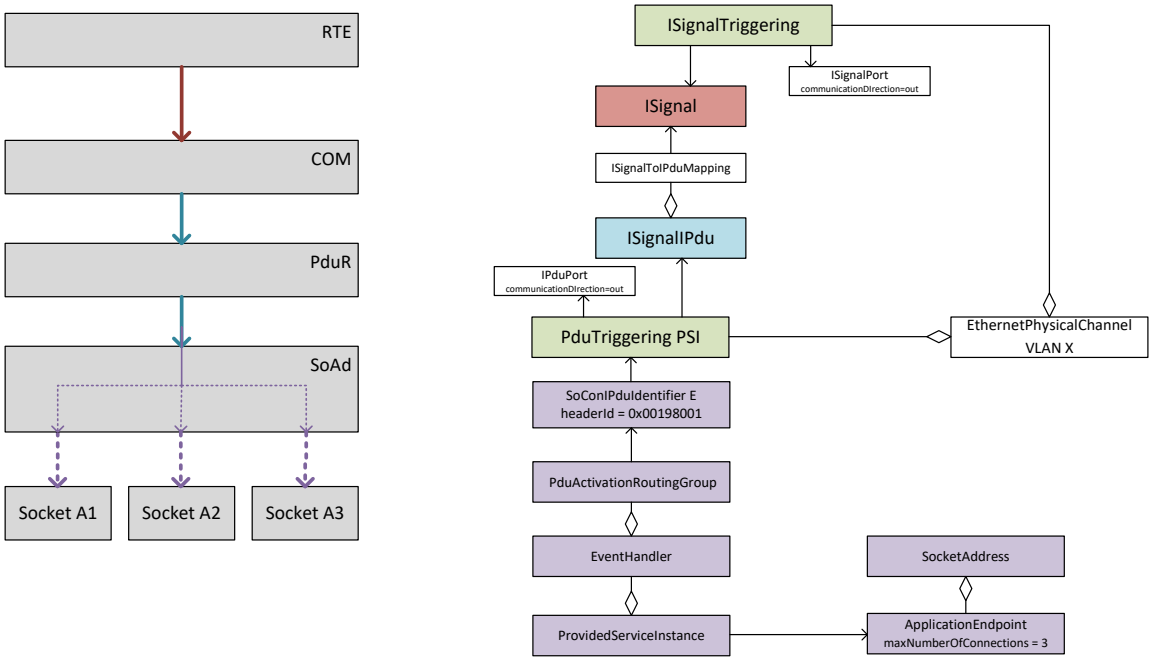

Figure 6.130: Socket Adapter fan-out with Service Discovery

In case of a static socket configuration the actual active number of possible Socket Adapter fan-out targets and their configuration is already known at configuration time. The example in figure 6.131 shows an `ISignalIPdu` which is fan-out to two statically configured VLANs.

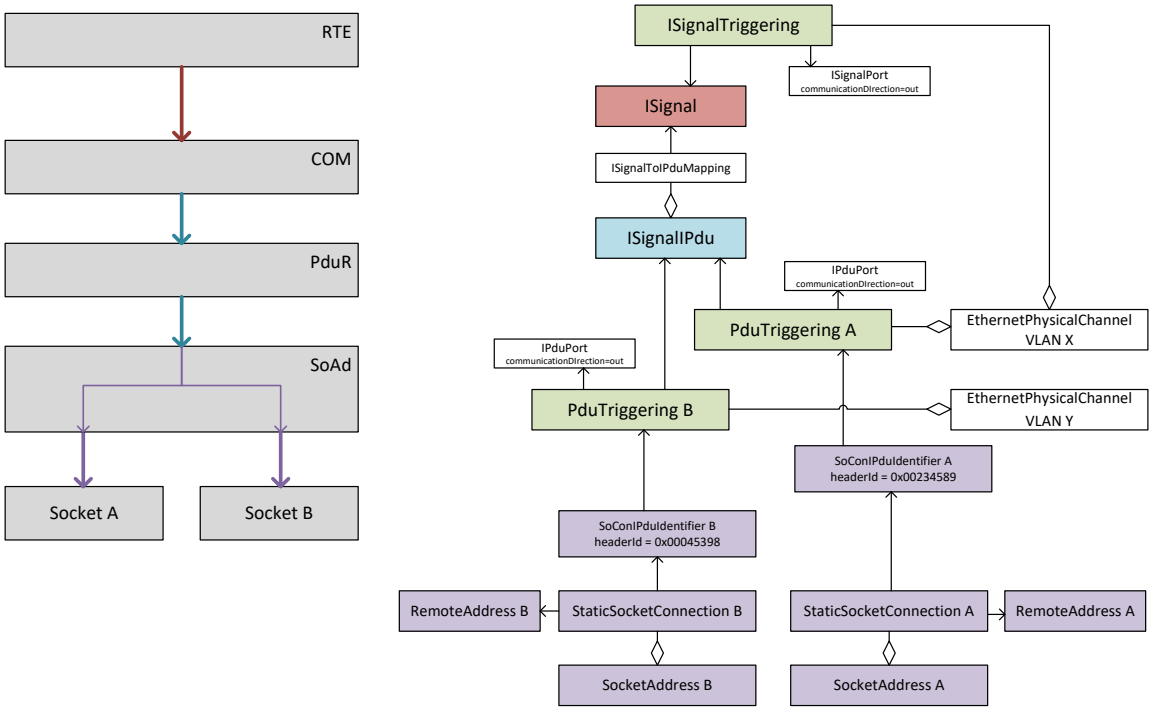

Figure 6.131: Static configured Socket Adapter fan-out to different VLANs

If the fan-out targets are located on the same VLAN several `StaticSocket-Connection`s can be defined referring to the same `SoConIPduIdentifier`. This is shown in figure 6.132.

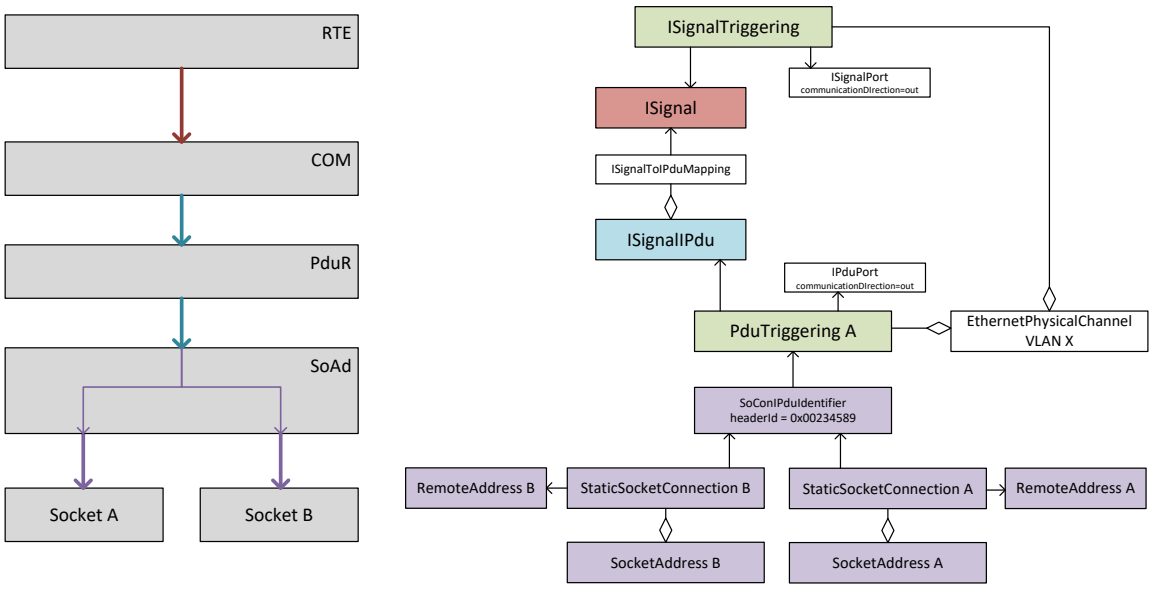

Figure 6.132: Static configured Socket Adapter fan-out to the same VLAN

6.11.3 Frame fan-out

[TPS_SYST_01114] Frame fan-out support ⌈AUTOSAR supports the Frame fan-out only on the FlexrayCluster (see [TPS_SYST_01113]).⌋

6.12 Fan-in

AUTOSAR supports the following fan-ins:

- RTE Signal fan-in
- Pdu fan-in
- IPdu Container fan-in

Note that the specification in this section does not apply for Client/Server communication. The respective details are described in section 5.2.1.3.

6.12.1 RTE fan-in

The RTE supports a "signal fan-in" where two or more ISignals of the same SystemSignal are received in multiple ISignalIPdus from potentially multiple senders, as described in [SWS_Rte_03760] and [SWS_Rte_03761].

[TPS_SYST_02357] RTE fan-in support for a SystemSignal ⌈RTE fan-in from multiple ISignals to a single SystemSignal is enabled in a System Description if the following conditions apply:

- several ISignals reference the SystemSignal in the role systemSignal and
- the SystemSignal is not referenced by a SystemSignalGroup
- a DataMapping references the SystemSignal and
- the DataMapping references an element in a RPortPrototype and
- the ISignals are received by the EcuInstance on which the RTE is running, i.e. an ISignalTriggering exists that references the ISignal in the role iSignal and references a ISignalPort of the EcuInstance with communicationDirection in.

⌋

Figure 6.115 shows a scenario where the RTE does not need to perform a fan-in since each SystemSignal is referenced by exactly one ISignal.

Please note that in all example scenarios that are shown in this chapter the ISignalToIPduMappings shall be modeled for each ISignal or ISignalGroup. For simplicity reasons, the following diagrams leave the ISignalToIPduMappings out and just assume that they exist.

In addition for simplicity reasons the examples scenarios are showing always one OUT EcuPort and/or one IN EcuPort. But the different ISignalTriggerings are also allowed to reference own ISignalPorts that are defined on the EcuInstance.

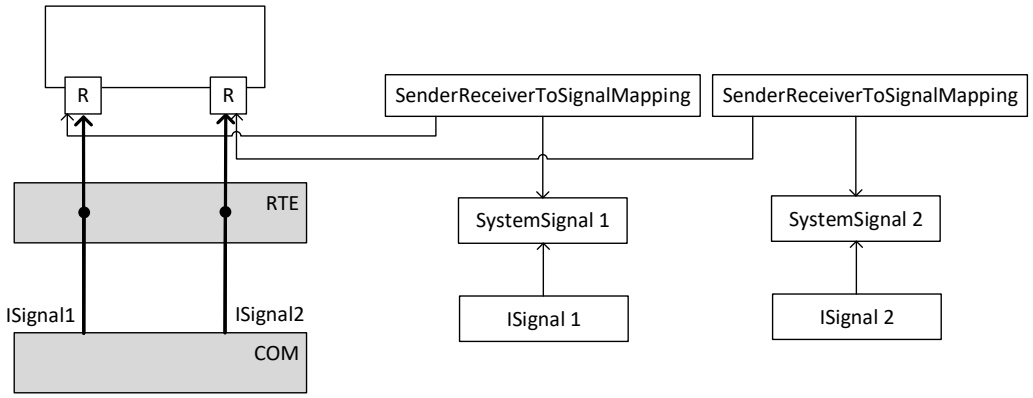

Figure 6.133: Scenario where RTE does not perform a fan-in

Figure 6.134 shows a scenario where the RTE needs to perform a fan-in since the SystemSignal is referenced by two ISignals, a SenderReceiverToSignalMapping is defined that maps a VariableDataPrototype in a SenderReceiverInterface of a RPortPrototype to the SystemSignal and the ISignals are both received by the EcuInstance.

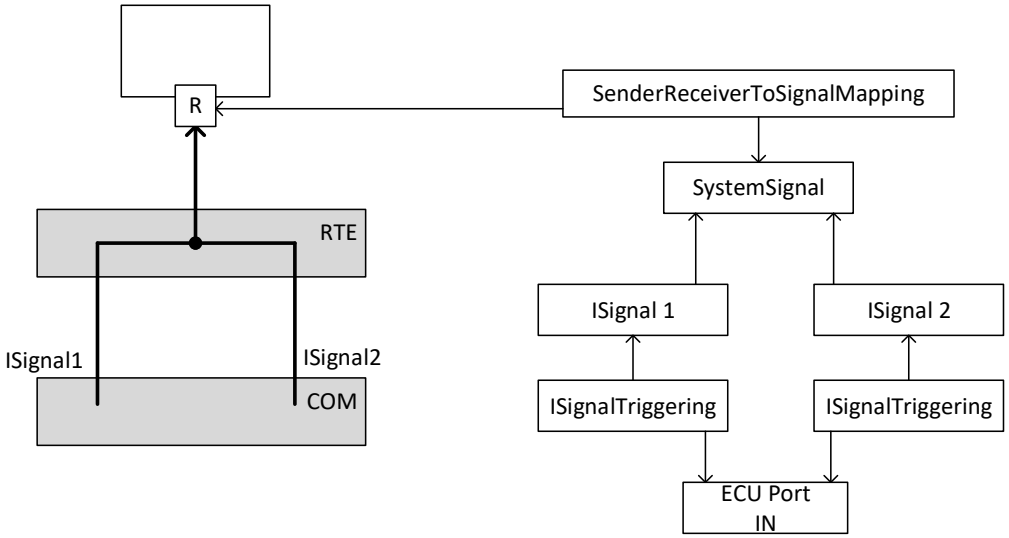

Figure 6.134: Scenario where RTE does perform a fan-in

A fan-in of two ISignalGroups can be modeled as well. In that case, two or more ISignalGroups refer to the same SystemSignalGroup, and each of the contained ISignals of the ISignalGroup needs to refer to its corresponding SystemSignal which in turn is part of the SystemSignalGroup.

[TPS_SYST_02358] RTE fan-in support for a SystemSignalGroup ⌈RTE fan-in from multiple ISignalGroups to a single SystemSignalGroup is enabled in a System Description if the following conditions apply:

- several ISignalGroups reference the SystemSignalGroup in the role systemSignalGroup and
- a DataMapping references the SystemSignalGroup and
- the DataMapping references an element in a RPortPrototype and
- each of the contained ISignals of the ISignalGroup refers to its corresponding SystemSignal which in turn is part of the SystemSignalGroup and
- the ISignalGroups are received by the EcuInstance on which the RTE is running, i.e. an ISignalTriggering exists that references the ISignalGroup in the role iSignalGroup and references a ISignalPort of the EcuInstance with communicationDirection in.

⌋

Figure 6.135 shows a scenario where the RTE needs to perform a fan-in since the SystemSignalGroup is referenced by two ISignalGroups, a SenderReceiverToSignalGroupMapping is defined that maps a VariableDataPrototype in a SenderReceiverInterface of a RPortPrototype to the SystemSignalGroup and both ISignalGroups are transmitted by the EcuInstance.

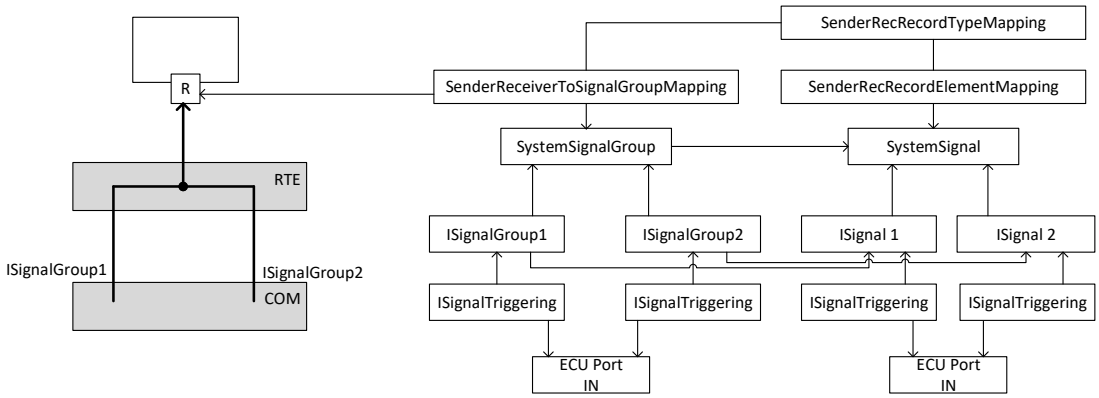

Figure 6.135: Scenario where RTE does perform a fan-in for a SystemSignalGroup

[TPS_SYST_03079] No RTE fan-in support for stand-alone SystemSignal and the same SystemSignal as part of a SystemSignalGroup ⌈There is no support by the RTE for a setup where a SystemSignal is received both, via a stand-alone

ISignal as well as part of an ISignalGroup, if the DataMapping references the SystemSignalGroup.⌋

[TPS_SYST_03079] implies that [TPS_SYST_03078] does not apply for SystemSignalGroup reception. The rational for this restriction is that the DataMapping references the SystemSignalGroup, and thus the application software component expects a composed data element. But in the case of the single SystemSignal reception only one member of that composed data element would be available for update. This setup is excluded.

This setup is excluded as illustrated in Figure 6.136, where the SenderReceiverInterface defines record composed of a and b. But the reception path of ISignal2 only is able to provide a, and thus the data consistency is violated.

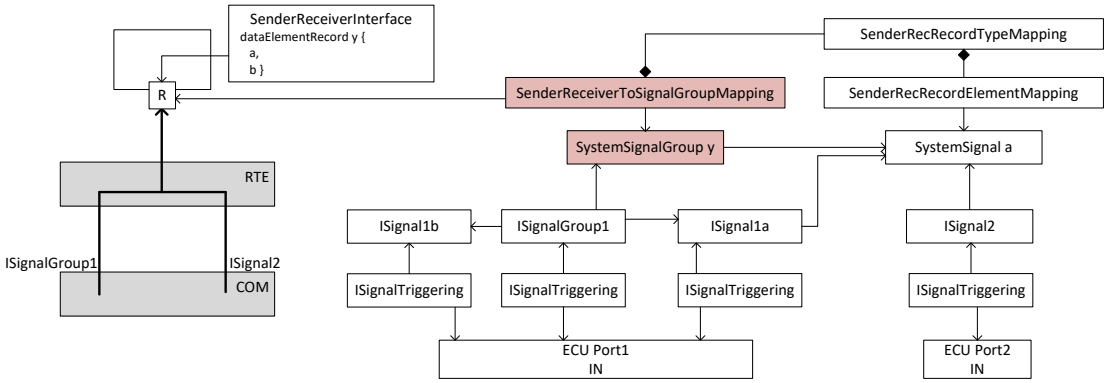

Figure 6.136: FanIn scenario which violates the data consistency

Figure 6.137 however defines a valid setup where the SenderReceiverInterface defines a primitive data element x to be received. And the data element x is received in both reception paths, via ISignal1x and also via ISignal2x (being part of the ISignal-Group2).

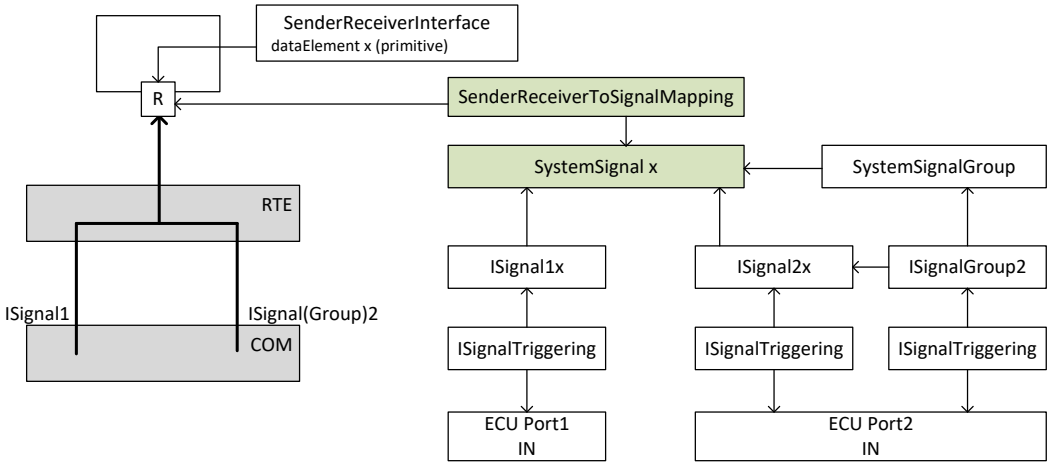

Figure 6.137: FanIn scenario in which SystemSignal x is received as part of a Signal Group and as a standalone Signal

6.12.2 Pdu fan-in

The Pdu Router supports the “PDU fan-in” where one IPdu is received from several sources.

[TPS_SYST_02376] Pdu Router fan-in support ⌈The Pdu Router fan-in is described by several PduTriggering elements that are referencing the same Pdu and each of these PduTriggerings is referencing an IPduPort with communicationDirection set to in of the same EcuInstance. According to the Cluster/Channel aggregation, the Pdu Router determines the clusters to use in its routing.⌋

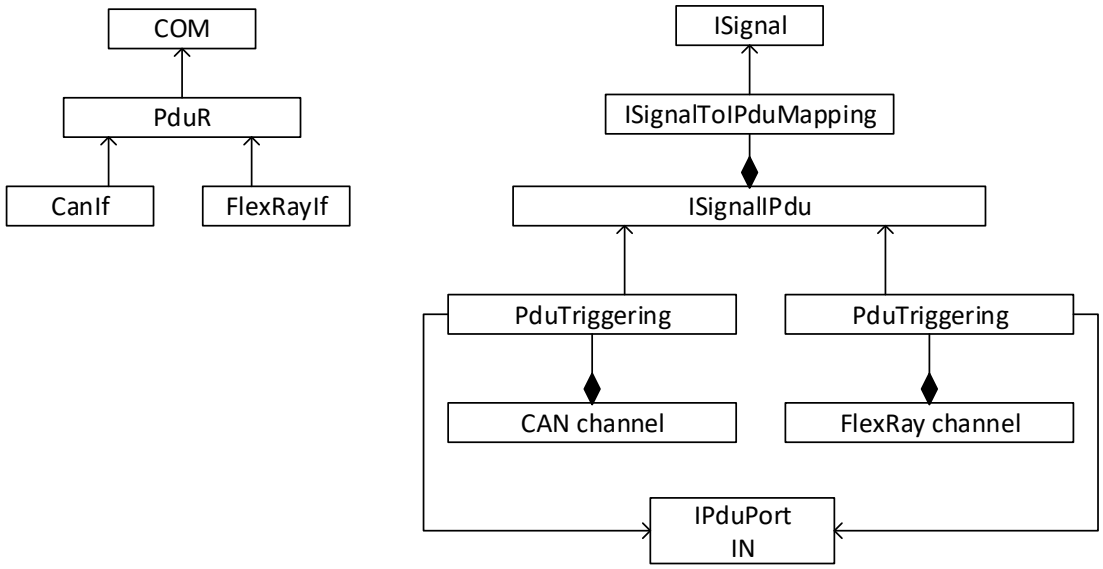

Figure 6.138: Scenario where PduR does perform a fan-in

6.12.3 IPdu Container fan-in

Using the ContainerIPdu (see section 6.3.1) it is possible to transport several Contained IPdus in one ContainerIPdu.

It is also possible to accept a Contained IPdu from any ContainerIPdu with ContainerIPdu.rxAcceptContainedIPdu set to RxAcceptContainedIPduEnum.acceptAll.

The example in figure 6.139 illustrates a scenario where in the configuration the ContainedIPdu is explicitly defined to be part of the two ContainerIPdus CIPdu1 and CIpdu3. Both ContainerIPdus have rxAcceptContainedIPdu set to acceptAll.

But also ContainerIPdu CIPdu2 has rxAcceptContainedIPdu set to acceptAll. And there is no explicit placement of ContainedIPdu in CIPdu2.

Due to the nature of acceptAll, if ContainedIPdu is received in CIPdu2 it will also be forwarded to the Pdu Router.

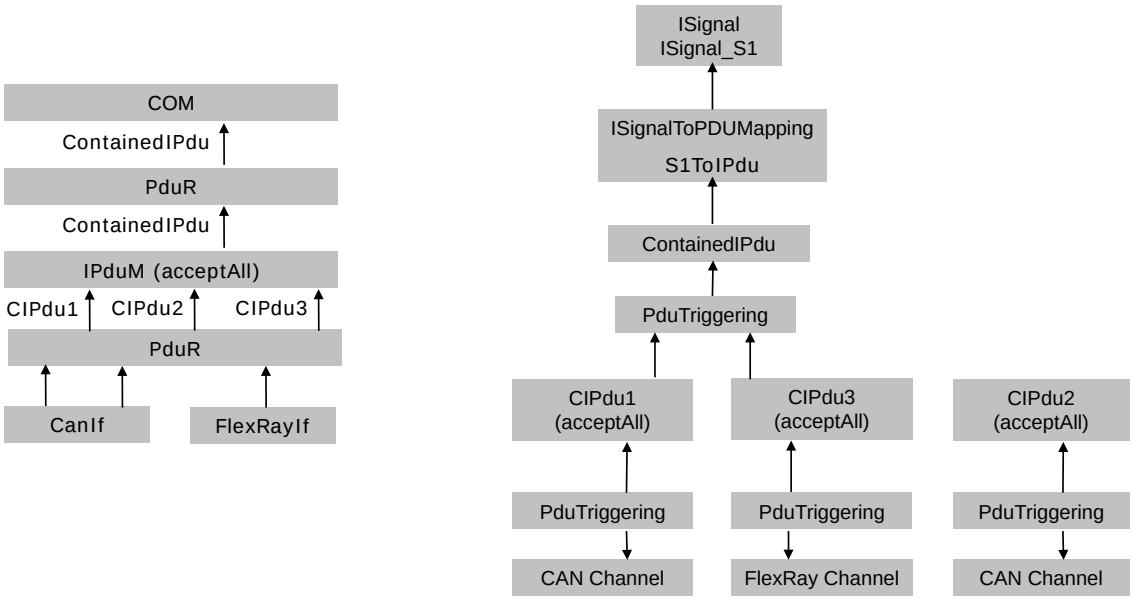

Figure 6.139: Container acceptAll fan-in

6.13 Log and Trace

The Dlt module collects debug information from applications or other software modules, adds metadata to the debug information, and sends the information to a Dlt sink.

The DltConfig element defines a Dlt module configuration for a specific EcuInstance and uses elements from the Log And Trace Template to describe the source of log and trace messages (application or module that produces the logging information) and the DltMessage that is sent out from the source to a sink.

[TPS_SYST_02373] Assignment of a Dlt Ecu Identifier to an EcuInstance ⌈The EcuInstance is represented in the Log And Trace Template by the DltEcu that is referenced from the EcuInstance via the aggregated DltConfig in the role dltEcu. The referenced DltEcu defines the ecuId that is transported in the standard header of the log and trace message.⌋

The DltApplication in the DltEcu is connected to a DltContext that is used to group DltMessages that are generated by the DltApplication. The PhysicalChannel on which the DltMessage is transported is represented by the DltLogChannel.

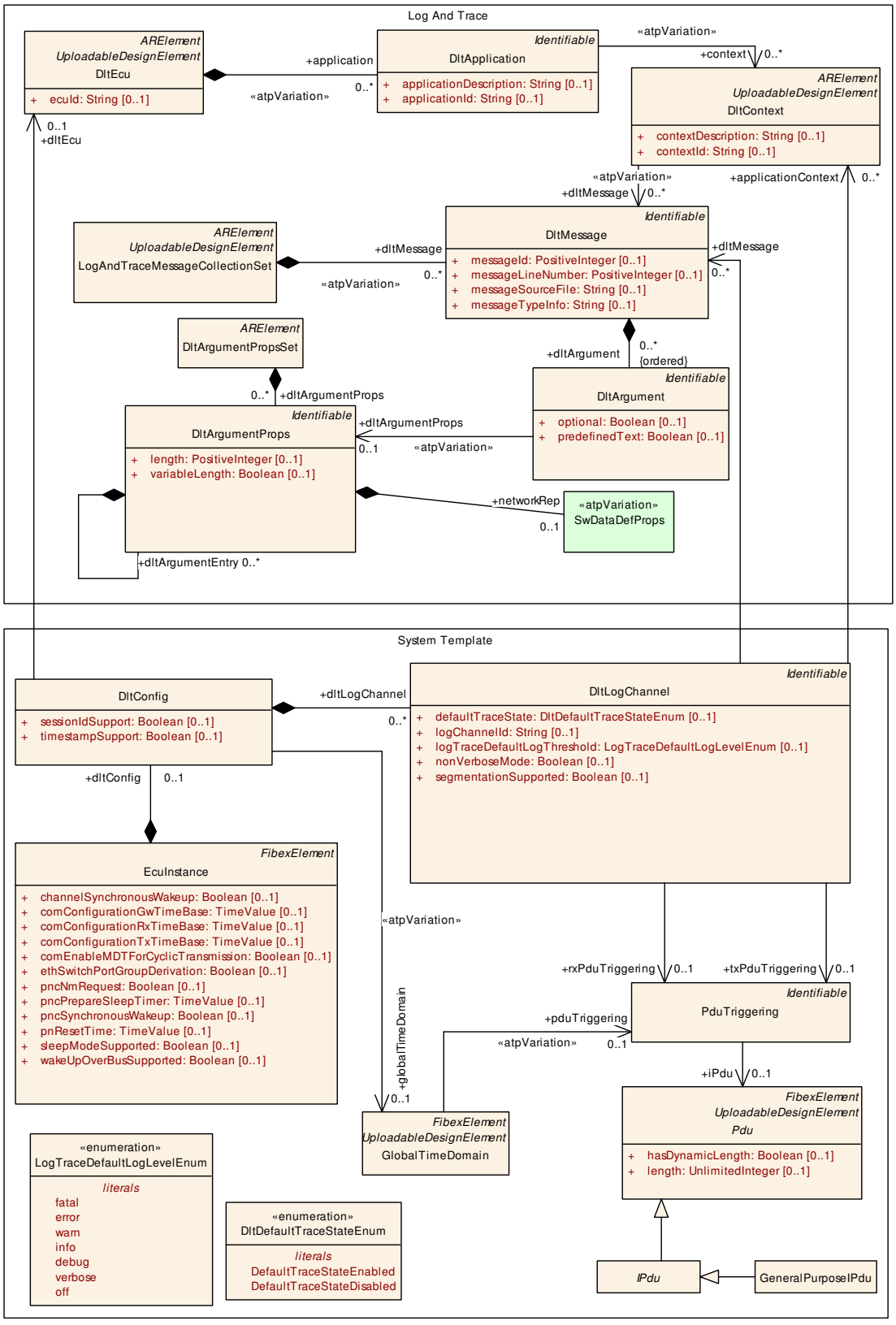

Figure 6.140: Dlt configuration of DltLogChannels and DltMessages

[TPS_SYST_02264] Usage of DltLogChannel ⌈In each DltConfig the PhysicalChannels that are used to transport the DltMessages are configured by the DltLogChannel elements.

Each DltLogChannel points to one PduTriggering in the role txPduTriggering to describe the Dlt Pdu that is transmitted by the DltLogChannel. The rxPduTriggering role is used to describe the Dlt Pdu that is received by the DltLogChannel.⌋

[TPS_SYST_02374] Assignment of DltMessage to DltLogChannels ⌈The assignment of DltMessages to a DltLogChannel for log/trace output is created with the dltMessage reference.⌋

[TPS_SYST_02375] Definition of DltLogChannels source ⌈The DltLogChannel references the DltContext in the role applicationContext to define the contextId and applicationId of the Software Component or Basic Software Module that produces the DltMessage for transmission to the sink.⌋

Class	**DltConfig**			
Note	This element defines a Dlt configuration for a specific Ecu. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	EcuInstance.dltConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dltEcu	DltEcu	0..1	ref	Reference to the Ecu representation in the Log And Trace Template. **Tags:** atp.Splitkey=dltEcu
dltLogChannel	DltLogChannel	*	aggr	Describes the DltLogChannels that are configured for the log/trace message output **Tags:** atp.Splitkey=dltLogChannel, dltLogChannel.shortName
globalTimeDomain	GlobalTimeDomain	0..1	ref	Reference to the GlobalTimeDomain this DltConfig shall be synchronized with **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomain, globalTimeDomain.globalTimeDomain, globalTimeDomain.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
sessionIdSupport	Boolean	0..1	attr	This attribute defines whether the sessionId is used or not.
timestampSupport	Boolean	0..1	attr	This attribute defines whether a timestamp shall be added to the Dlt messages or not.

Table 6.338: DltConfig

Class	**DltLogChannel**
Note	This element contains the settings for the log/trace message output for a tuple of ApplicationId and ContextId (verbose mode) or a SessionId (non-verbose mode). This Class is only used by the AUTOSAR Classic Platform.

▽

△

Class	**DltLogChannel**			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltConfig.dltLogChannel			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationContext	DltContext	*	ref	Reference to the Swc that produces the log or trace message. Please note that this reference shall not be set in case that the Bsw module produces the associated log or trace messages. **Tags:** atp.Splitkey=applicationContext
defaultTraceState	DltDefaultTraceStateEnum	0..1	attr	This attributes defines the default trace status.
dltMessage	DltMessage	*	ref	Reference to DltMessages that can be transported over the DltLogChannel in the DltPdu. **Tags:** atp.Splitkey=dltMessage
logChannelId	String	0..1	attr	This attribute identifies the Channel for usage within the Log And Trace protocol.
logTraceDefaultLogThreshold	LogTraceDefaultLogLevelEnum	0..1	attr	This attribute allows to set a log level Threshold for Log Level filtering.
nonVerboseMode	Boolean	0..1	attr	This attribute defines whether this channel supports non-Verbose Dlt messages. If disabled only verbose mode messages shall be used.
rxPduTriggering	PduTriggering	0..1	ref	Reference to DltPdu that is received by the DltLog Channel **Tags:** atp.Splitkey=rxPduTriggering
segmentationSupported	Boolean	0..1	attr	If enabled, segmentation will be used if a DLT message is larger than Pdu.length referenced via DltLogChannel.tx PduTriggering.
txPduTriggering	PduTriggering	0..1	ref	Reference to DltPdu that is transmitted by the DltLog Channel. **Tags:** atp.Splitkey=txPduTriggering

Table 6.339: DltLogChannel

Enumeration	**DltDefaultTraceStateEnum**
Note	This enumeration defines the supported values for the Dlt default trace state.
Aggregated by	DltLogChannel.defaultTraceState
Literal	***Description***
DefaultTraceStateDisabled	The default trace state is disabled **Tags:** atp.EnumerationLiteralIndex=1
DefaultTraceStateEnabled	The default trace state is enabled **Tags:** atp.EnumerationLiteralIndex=0

Table 6.340: DltDefaultTraceStateEnum

Enumeration	**LogTraceDefaultLogLevelEnum**
Note	This enum defines available log&trace log levels that may be used to define the severity level of a log message.
Aggregated by	DltLogChannel.logTraceDefaultLogThreshold, DltLogSink.defaultLogThreshold
Literal	***Description***

▽

△

Enumeration	LogTraceDefaultLogLevelEnum
debug	Detailed information for programmers **Tags:** atp.EnumerationLiteralIndex=4
error	Error with impact to correct functionality **Tags:** atp.EnumerationLiteralIndex=1
fatal	Fatal error **Tags:** atp.EnumerationLiteralIndex=0
info	High level information **Tags:** atp.EnumerationLiteralIndex=3
off	logging is turned off **Tags:** atp.EnumerationLiteralIndex=6
verbose	Verbose debug message **Tags:** atp.EnumerationLiteralIndex=5
warn	Warning if correct behavior cannot be ensured **Tags:** atp.EnumerationLiteralIndex=2

Table 6.341: LogTraceDefaultLogLevelEnum

[constr_5097] `DltLogChannel.txPduTriggering` and `DltLogChannel.rxPduTriggering` shall point to `GeneralPurposeIPdu`s of category DLT

Imposition time: CP: `IT_SysDesc`

⌈`DltLogChannel` shall only reference `PduTriggering`s that are pointing to `GeneralPurposeIPdu`s of category DLT in the roles `txPduTriggering` and `rxPduTriggering`.⌋

[constr_5306] Restriction of `DltLogChannel.logChannelId` attribute value

Imposition time: CP: `IT_SysDesc`

⌈The `DltLogChannel.logChannelId` attribute value shall be composed of maximum four ASCII characters.⌋

[constr_5307] Existence of `DltLogChannel.logChannelId`

Imposition time: CP: `IT_SysDesc`

⌈For each `DltLogChannel`, the attribute `logChannelId` shall be defined.⌋

[constr_5308] Existence of `DltLogChannel.nonVerboseMode`

Imposition time: CP: `IT_SysDesc`

⌈For each `DltLogChannel`, the attribute `nonVerboseMode` shall be defined.⌋

[constr_5309] Existence of `DltConfig.sessionIdSupport`

Imposition time: CP: `IT_SysDesc`

⌈For each `DltConfig`, the attribute `sessionIdSupport` shall be defined.⌋

[constr_5310] Existence of DltConfig.timestampSupport

Imposition time: CP: IT_SysDesc

⌈For each DltConfig, the attribute timestampSupport shall be defined.⌋

[constr_5311] Existence of DltLogChannel.logTraceDefaultLogThreshold

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute logTraceDefaultLogThreshold shall be defined.⌋

[constr_5312] Existence of DltLogChannel.defaultTraceState

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the attribute defaultTraceState shall be defined.⌋

[constr_5313] Existence of DltLogChannel.txPduTriggering

Imposition time: CP: IT_SysDesc

⌈For each DltLogChannel, the reference to PduTriggering in the role txPduTriggering shall be defined.⌋

[constr_5314] DltLogChannel txPduTriggering and rxPduTriggering shall be on the same network

Imposition time: CP: IT_SysDesc

⌈The PduTriggerings that are referenced by a DltLogChannel in the role txPduTriggering and rxPduTriggering shall be aggregated by the same PhysicalChannel.⌋

6.14 Support of Complex Drivers

The System Template allows the integration of custom communication means into AUTOSAR EcuInstances.

[TPS_SYST_01115] CDD communication support ⌈The elements UserDefinedPdu and UserDefinedIPdu shall be used to describe the Pdu-based communication via Complex Drivers.⌋

The UserDefinedPdu and UserDefinedIPdu elements are described in chapter 6.3 in more detail.

The UserDefinedIPdu can be used to describe the communication if a new BSW module was added above the PduR, e.g a Diagnostic Service.

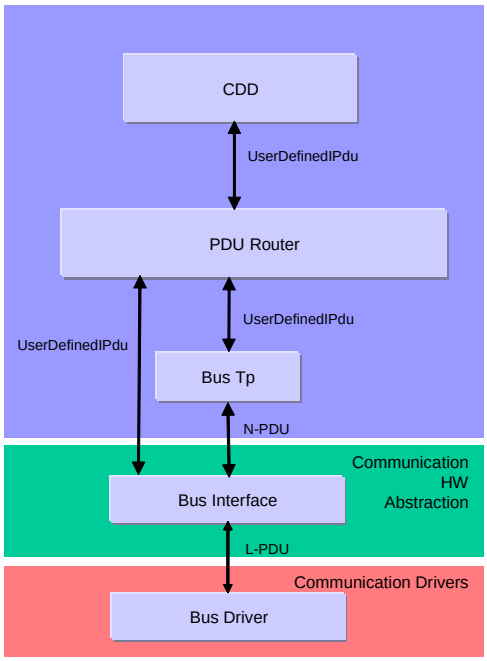

Figure 6.141: CDD over PduR

The `UserDefinedPdu` can be used to describe the communication if a new BSW module was added above an Interface, e.g. a new Nm module or XCP.

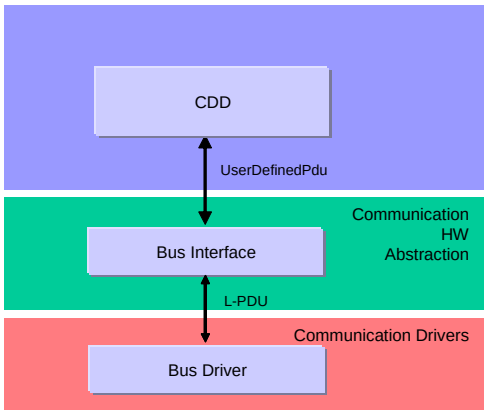

Figure 6.142: CDD over Bus Interface

6.15 MetaData in SenderReceiverInterface

As described in [4], there is the ability for the application software to unlock access to `Pdu` meta-data by using the aggregation `SenderReceiverInterface.metaDataItemSet`.

This modeling allows for the arbitrary definition of meta-data semantics that - of course - has to be supported by the `Pdu` where a specific `dataElement` is finally mapped to.

This consistency is ensured by deriving the Pdu meta-data from the modeling in SenderReceiverInterfaces. However, this approach can only work if all dataElements mapped to the given Pdu define the same meta-data structure or do not define the usage of meta-data at all.

[constr_5100] Compatibility of two MetaDataItemSets

Imposition time: CP: IT_SysDesc

⌈Under the condition that sender and receiver typed by a SenderReceiverInterface use meta-data and are mapped to the same EcuInstance the following condition applies: two MetaDataItemSets are compatible if all of the following conditions are fulfilled:

- They aggregate the same number of MetaDataItems.
- The value of MetaDataItem.length of corresponding MetaDataItems is identical.
- The value of MetaDataItem.metaDataItemType of corresponding MetaDataItems is identical.

⌋

[constr_5101] Consistent Definition of meta-data

Imposition time: CP: IT_SysDesc

⌈If the dataElement referenced by a SenderReceiverToSignalMapping is also referenced by a MetaDataItemSet in the role dataElement and the mapping via SystemSignal, ISignal, and ISignalToIPduMapping down to an ISignalIPdu exists then all other dataElements that are also mapped to the same ISignalIPdu shall either

- not be referenced by a MetaDataItemSet in the role dataElement (i.e. does not make use of meta-data) or
- the definition of meta-data in the context of the affected SenderReceiverInterfaces is compatible (according to the definition of compatible specification of meta-data described in [constr_5100]).

⌋

6.16 Signal Service Translation

AUTOSAR Adaptive Platform restricts communication paradigm to Service-oriented communication. A major part of the vehicle however still uses Signal-based communication means, therefore a translation of these two approaches has to be performed.

One major goal of AUTOSAR is that it covers both

- high-performance microprocessor Machines (connected via high payload and high bandwidth Ethernet networks)
- highly embedded microcontroller ECUs (connected via ethernet, but also via low payload and low bandwidth CAN and LIN networks).

A seamless development of a vehicle shall be supported with one AUTOSAR methodology. Therefore a translation is required which closes the gap between

- Signal-based communication on Classic platform
- Service-oriented communication on Adaptive platform.

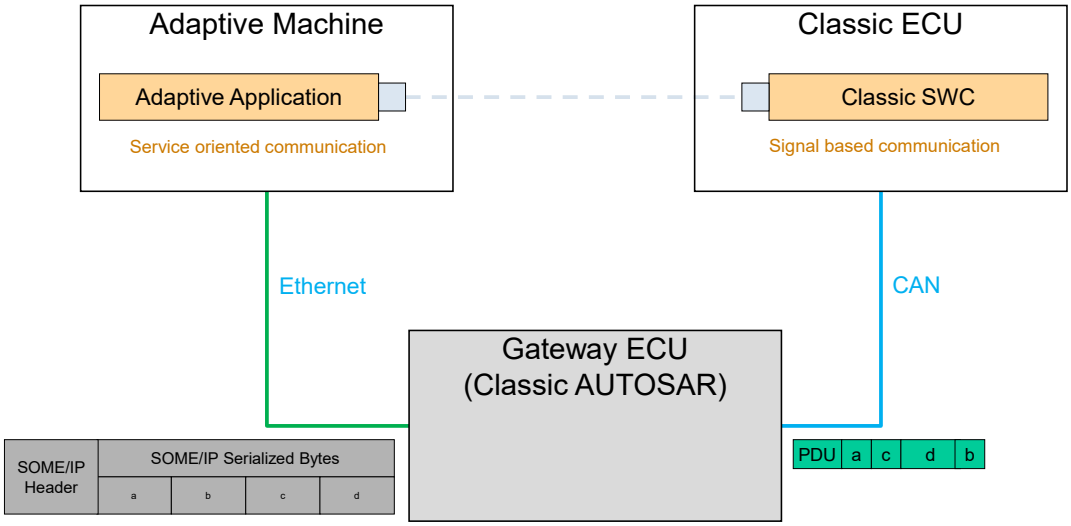

Figure 6.143: `Signal/Service Translation` in Classic platform gateway ECU

The goal of this chapter is to standardize a translation between the two communication transport configurations on an AUTOSAR Classic platform ECU. A similar approach is also provided in the Manifest specification for the Adaptive platform [38]. It is up to the vehicle architecture design to choose whether the `signal/service translation` shall be implemented on a Classic platform ECU or on an Adaptive platform Machine.

6.16.1 Architectural setup

The implementation of the `signal/service translation` on the Classic platform is done in an Application Software Component above the RTE. This applies for *events* and *field notifiers*. Methods are handled separately (see section 6.16.1.1).

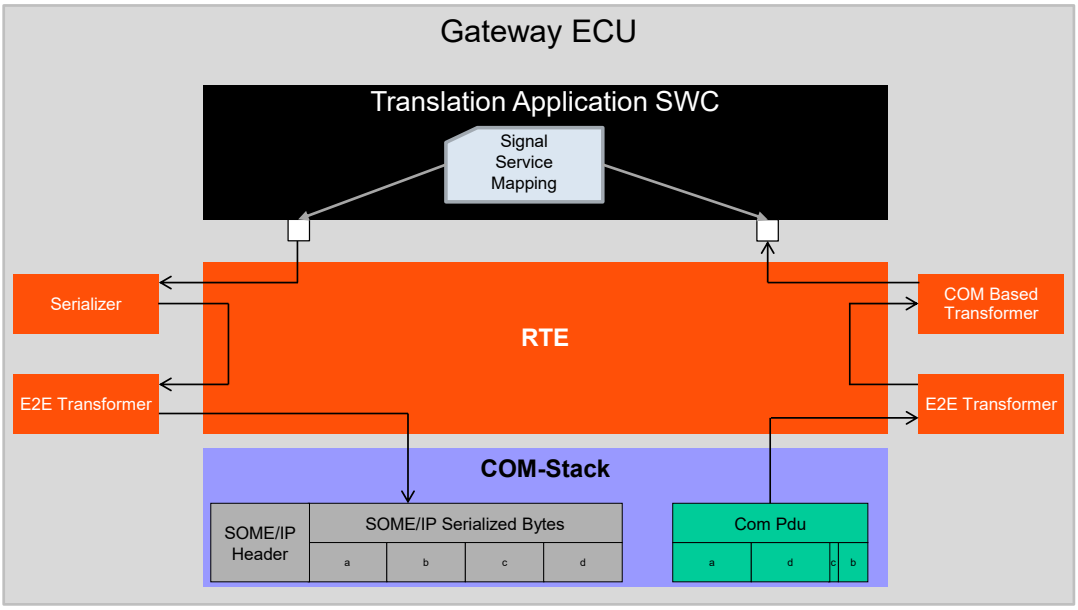

Figure 6.144: `Signal/Service Translation` Application Software Component

For the signal-based part the full functionality of the Classic platform COM-Stack is available and may be configured such that the signal-based `ISignalIPdu`s may originate from a variety of sources (Can, Lin, FlexRay) and the `ISignalIPdu`s may be safety and security protected.

For the service-oriented part it has to be guaranteed that the defined SOME/IP Service actually is compatible to the Adaptive platform. This applies for the payload part (e.g. the SOME/IP serializer has to be used) as well as for the control path using *BswM* and *ServiceDiscovery*.

The behavioral part of the Translation Software Component itself defines how the data from signal-based side is transported to the service-oriented side, and vice versa (see section 6.16.4).

The following terminology is used in the context of `signal/service translation`:

Signal/service translation defines the feature this chapter is concerned with. It does not prescribe a specific translation direction.

Signal-service-translation defines the translation direction from a signal-based to a service oriented representation.

Service-signal-translation defines the translation direction from a service oriented to a signal-based representation.

6.16.1.1 Method handling

The handling of methods (or getter/setter calls of fields) has to be serialized using the SOME/IP transformer. And this is only supported on Ethernet networks (see [constr_5117]).

Therefore there is no need to perform a translation because the methods are already usable on the adaptive platform.

6.16.2 Mapping description

`Signal/service translation` is used to alter the serialization representation of data to be compatible with the respective transport network. I.e. on an Ethernet network a SOME/IP serialized data representation is mostly suitable, while on a CAN network the packed signal-based data representation often required due to the low payload data size available.

As indicated in section 6.16.1 the implementation of the translation shall be done in an Application Software Component above the RTE. For the definition of the intended mapping and behavior however the `CompositionSwComponentType` is used. This allows to represent the requirements on the translation and still allow some freedom with respect to the actual implementation later on.

The element which defines the behavioral aspects of the `signal/service translation` is the meta-class `SignalServiceTranslationProps`. The references to the `VariableDataPrototype` in the role `translationTarget` define to which events the `SignalServiceTranslationProps` apply. For this reference a `VariableDataPrototypeInSystemInstanceRef` is used (see also chapter B.9).

- In case of `signal-service-translation` the `SignalServiceTranslationProps.translationTarget` collect all resulting events which belong to one provided service instance.
- In case of `service-signal-translation` the `SignalServiceTranslationProps.translationTarget` collect all resulting signals which are translated from one service instance.

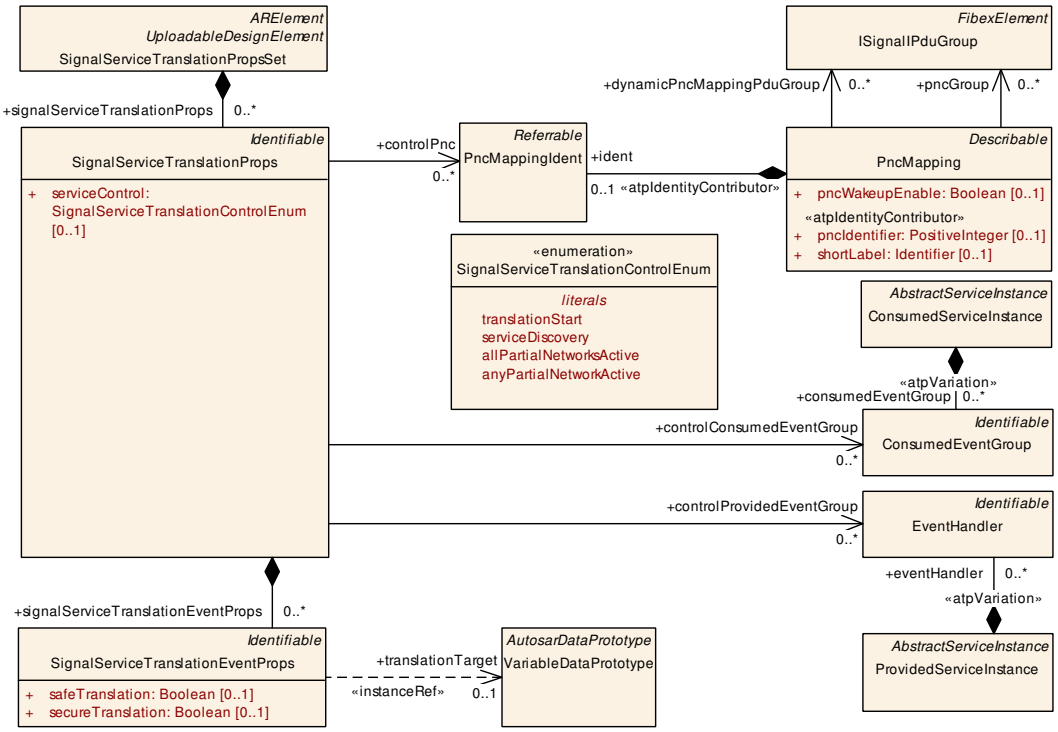

Figure 6.145: `Signal/Service Translation` properties

Class	**SignalServiceTranslationPropsSet**			
Note	Collection of SignalServiceTranslationProps. **Tags:** atp.recommendedPackage=SignalServiceTranslationProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
signalService Translation Props	SignalService TranslationProps	*	aggr	Collection of SignalServiceTranslationProps. **Tags:** atp.Splitkey=signalServiceTranslationProps, signal ServiceTranslationProps.shortName

Table 6.342: SignalServiceTranslationPropsSet

Class	**SignalServiceTranslationProps**			
Note	This element allows to define the properties which are applicable for the signal/service translation service.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationPropsSet.signalServiceTranslationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
control Consumed EventGroup	ConsumedEventGroup	*	ref	Reference to the EventGroup which encapsulates the signal-based payload. **Tags:** atp.Splitkey=controlConsumedEventGroup This Attribute is only used by the AUTOSAR Classic Platform.

△

Class	**SignalServiceTranslationProps**			
controlPnc	PncMappingIdent	*	ref	Reference to the PNCs which control the offer/subscribe behavior of the translated service instance. **Tags:** atp.Splitkey=controlPnc This Attribute is only used by the AUTOSAR Classic Platform.
controlProvided EventGroup	EventHandler	*	ref	Reference to the provided event group (aka Event Handler) which is automatically available when service Control equals translationStart. **Tags:** atp.Splitkey=controlProvidedEventGroup This Attribute is only used by the AUTOSAR Classic Platform.
serviceControl	SignalService TranslationControlEnum	0..1	attr	Defines how the service instance control shall behave.
signalService Translation EventProps	SignalService TranslationEventProps	*	aggr	Defines properties for a single translated event. **Tags:** atp.Splitkey=signalServiceTranslationEventProps, signalServiceTranslationEventProps.shortName

Table 6.343: SignalServiceTranslationProps

Class	**SignalServiceTranslationEventProps**			
Note	This element allows to define the properties which are applicable for the signal/service translation event.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationProps.signalServiceTranslationEventProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
elementProps	SignalService TranslationElement Props	*	aggr	Defines properties for a single translated element. **Tags:** atp.Splitkey=elementProps, elementProps.short Name This Attribute is only used by the AUTOSAR Classic Platform.
safeTranslation	Boolean	0..1	attr	Defined whether the translation shall happen in a safe way.
secure Translation	Boolean	0..1	attr	Defined whether the translation shall happen in a secure way.
translation Target	VariableDataPrototype	0..1	iref	Reference to a VariableDataPrototype representing the target of signal/service translation. **Tags:** atp.Splitkey=translationTarget, translation Target.contextComponent, translationTarget.context Composition, translationTarget.contextPort, translation Target.targetDataPrototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table 6.344: SignalServiceTranslationEventProps

A simple `signal/service translation` setup is to have a one-to-one correspondence between the signal-based data definition and the service-oriented data definition. This is illustrated in figure 6.146.

Here the setup allows for a simple `PassThroughSwConnector` because the involved `PortInterface`s are identical.

Please consult with [constr_1248] in the Software Component Template [4] for details on the compatibility of connected PortInterfaces.

Note that the translated *P1.dataX* and *P2.dataB* may be configured in the COM-Stack to be events that belong to one service instance or to belong to two different service instances. The setup in the example of figure 6.146 uses two SignalServiceTranslationProps elements and those use SignalService-TranslationEventProps, indicating that *P1.dataX* belongs to a different service instance than *P2.dataB*.

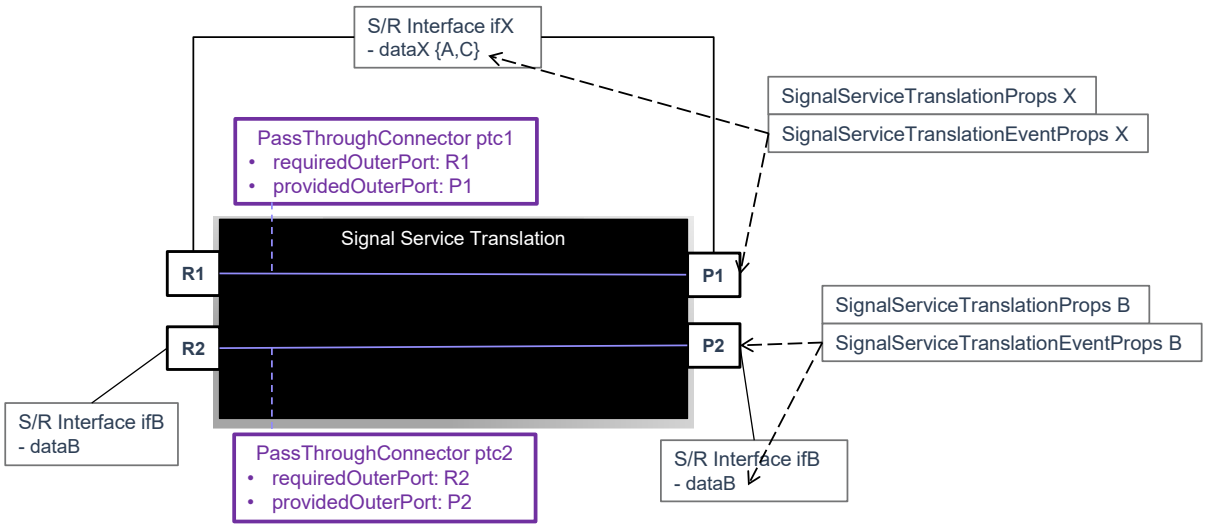

Figure 6.146: Mapping description using only PassThroughSwConnectors

The usage of the PassThroughSwConnector and (optionally) the accompanying PortInterfaceMapping already suffice to describe the intended mapping from a structural point of view:

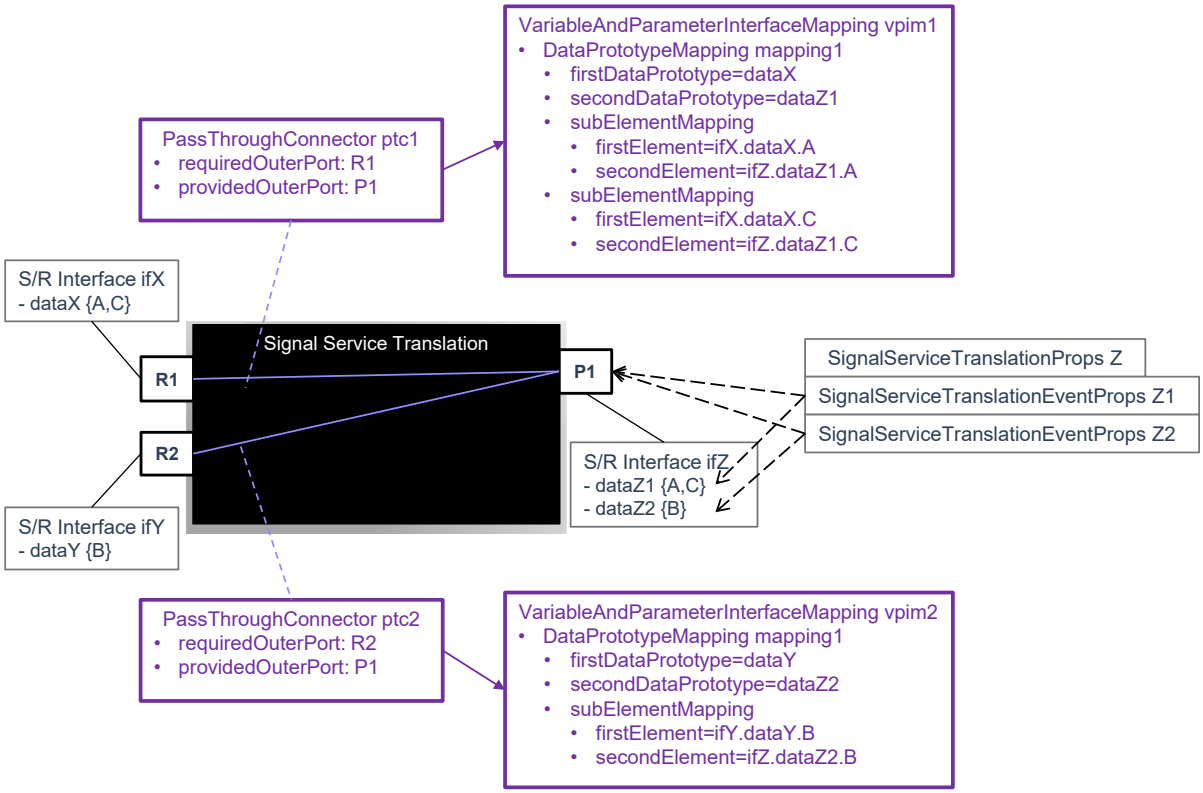

Figure 6.147: Mapping description using PassThroughSwConnectors and PortInterfaceMappings

In figure 6.147 the additional usage of `PortInterfaceMapping`s is required because the connected `PortInterface`s are not identical (in this example have different `shortName`s).

In this example there is only one `SignalServiceTranslationProps` element defined which uses two `SignalServiceTranslationEventProps` to refer to both the data elements *dataZ1* and *dataZ2*. Thus the translated events belong to the same service instance.

Although the figure 6.147 indicates that the output Port *P1* data *dataZ1* and *dataZ2* are composed of input from several sources, this is not true for the individual data elements. *P1.dataZ1* is solely composed out of *R1.dataX*, while *P1.dataZ2* is composed out of *R2.dataY*. Thus the example shown is still a mapping from *one source*.

The usage of `PortInterfaceMapping` (specifically `VariableAndParameterInterfaceMapping` with a `DataPrototypeMapping`) brings along a restriction on the applicability of the `PassThroughSwConnector` usage. Specifically the mapped elements have to be *both* either

- typed by an `ApplicationPrimitiveDataType` or
- typed by an `ApplicationCompositeDataType`.

The mixed case (where one AutosarDataPrototype is typed by an ApplicationPrimitiveDataType and the other is typed by an ApplicationCompositeDataType) is not supported by the DataPrototypeMapping. This is illustrated in figure 6.148: The interface *ifX* has two VariableDataPrototypes typed by ApplicationPrimitiveDataTypes, while the *ifZ* has a VariableDataPrototype typed by a ApplicationCompositeDataType (having the members *A, C*). The SubElementMapping requires to have two SubElementRefs, but *dataA* is primitive. This SubElementMapping is not supported, as indicated by the *???* in figure 6.148.

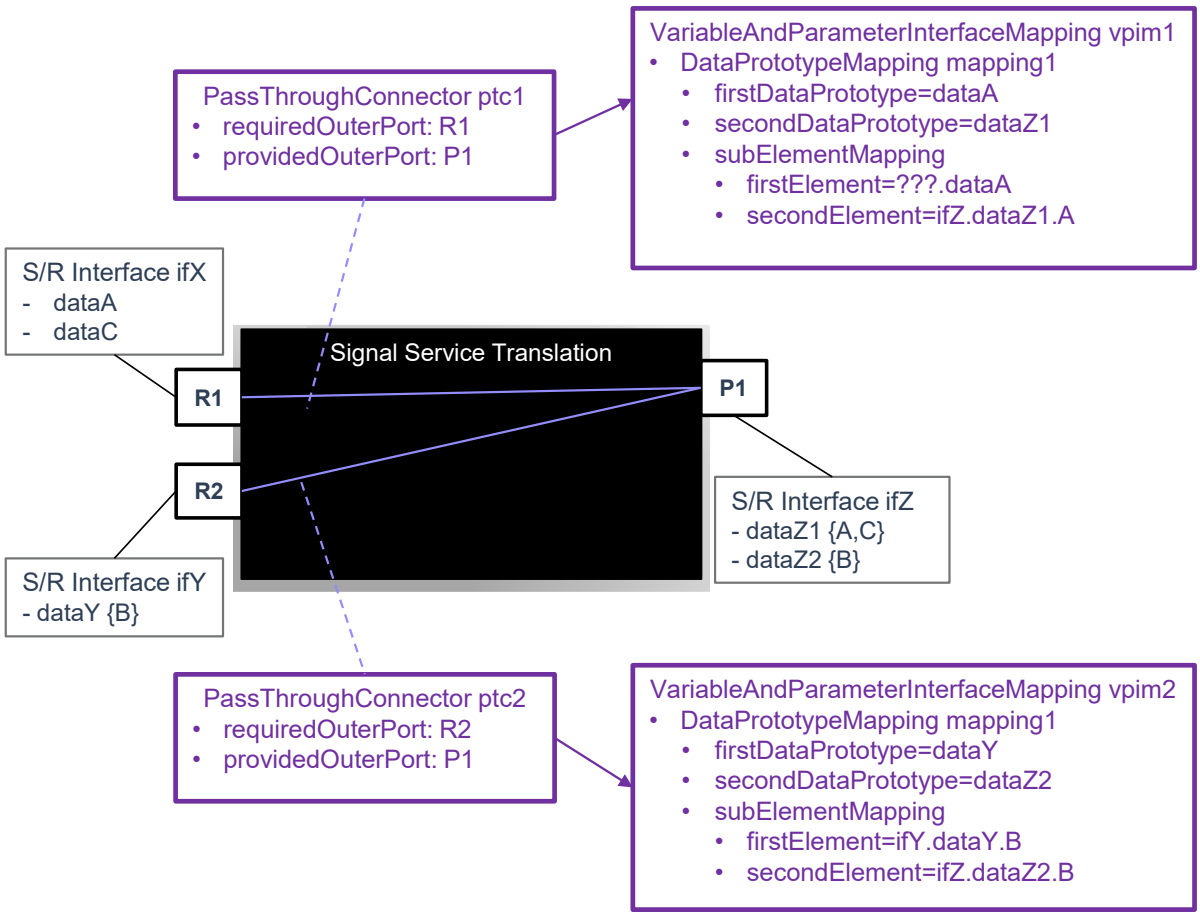

Figure 6.148: Not supported mapping description using PassThroughSwConnectors and PortInterfaceMappings

6.16.2.1 Filters and Transmission Triggers

The filters and transmissionTriggers can be used to define behavioral aspects of the signal/service translation. They are defined in the SignalServiceTranslationElementProps, which is aggregated by the SignalServiceTranslationEventProps.

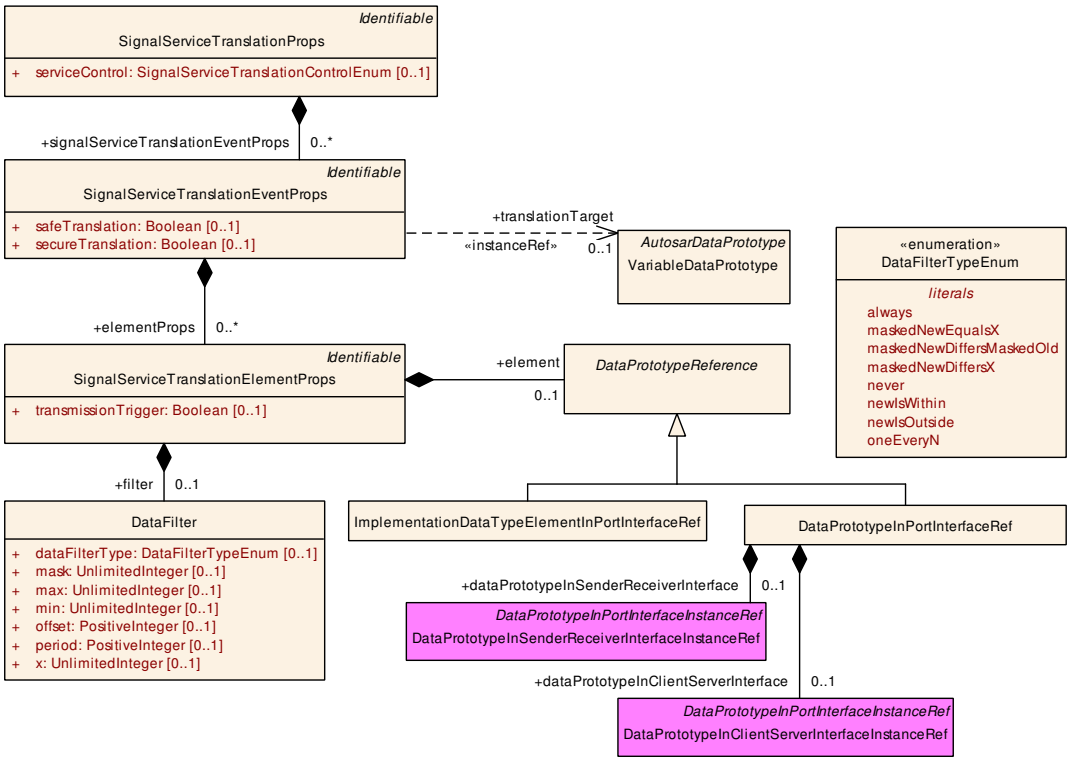

Figure 6.149: `Signal/Service Translation` element properties

Class	**SignalServiceTranslationElementProps**			
Note	Defined translation properties for individual mapped elements. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationEventProps.elementProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
element	DataPrototype Reference	0..1	aggr	Reference to the leaf element the SignalService TranslationElementProps apply to. **Tags:** atp.Splitkey=element
filter	DataFilter	0..1	aggr	Defines an optional filter to be applied during translation. **Tags:** atp.Splitkey=filter
transmission Trigger	Boolean	0..1	attr	Defines whether the source element (which is mapped to the referenced element) triggers the sending of the respective payload.

Table 6.345: SignalServiceTranslationElementProps

To which data the `filter` and/or `transmissionTrigger` applies is defined by the combination of `SignalServiceTranslationEventProps.translationTarget` and `SignalServiceTranslationElementProps.element` references.

In case the `filter` and/or `transmissionTrigger` shall apply to the `translationTarget` as a whole the reference `SignalServiceTranslationElementProps.element` shall not be defined. This is specifically true if `SignalServiceTranslationEventProps.translationTarget` refers to a `VariableDataPrototype` that is typed by a primitive `AutosarDataType`:

[constr_3651] No `element` in case `translationTarget` is primitive

Imposition time: CP: IT_SysDesc

⌈If SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a primitive AutosarDataType then the reference SignalServiceTranslationElementProps.element shall not be defined.⌋

Since the reference DataPrototypeReference is used in several scenarios (see [TPS_SYST_02195]) it needs to be constrained for the usage in scope of the signal/service translation (i.e., for SignalServiceTranslationElementProps.element): The DataPrototypeReference used to define the SignalServiceTranslationElementProps.element is restricted to either DataPrototypeInSenderReceiverInterfaceInstanceRef or ImplementationDataTypeElementInPortInterfaceRef.

[constr_3652] Allowed sub-classes of `DataPrototypeReference` in the context of `signal/service translation`

Imposition time: CP: IT_SysDesc

⌈If a DataPrototypeReference in the role SignalServiceTranslationElementProps.element is used then following sub-classes are supported:

- if the reference target is typed by an ApplicationDataType then the DataPrototypeInSenderReceiverInterfaceInstanceRef shall be used and shall target an ApplicationCompositeElementDataPrototype.
- if the reference target is typed by an ImplementationDataType then the ImplementationDataTypeElementInPortInterfaceRef shall be used.

⌋

It is important that the SignalServiceTranslationEventProps.translationTarget reference and the SignalServiceTranslationElementProps.element in the context of one SignalServiceTranslationEventProps are consistent.

Consistent usage of either ApplicationDataType or ImplementationDataType and consistent translationTarget target and element root:

[constr_3653] Consistent `translationTarget` and `element` in case `ApplicationDataType` is used

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by an ApplicationDataType (targetDataPrototype of the VariableDataPrototypeInSystemInstanceRef) then every SignalServiceTranslationElementProps.element reference that is defined in the context of the SignalServiceTranslationEventProps shall have that VariableDataPrototype as the rootDataPrototypeInSr of the DataPrototypeInSenderReceiverInterfaceInstanceRef.⌋

[constr_3654] Consistent translationTarget and element in case ImplementationDataType is used

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by an ImplementationDataType (targetDataPrototype of the VariableDataPrototypeInSystemInstanceRef) then every SignalServiceTranslationElementProps.element reference that is defined in the context of the SignalServiceTranslationEventProps shall have that VariableDataPrototype as the rootDataPrototype of the ImplementationDataTypeElementInPortInterfaceRef.⌋

[TPS_SYST_03062] Definition of a primitive target for SignalServiceTranslationElementProps ⌈The target of a SignalServiceTranslationElementProps is considered primitive if either:

- SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a primitive AutosarDataType or
- SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a composite AutosarDataType and SignalServiceTranslationElementProps.element refers to an AutosarDataPrototype that is typed by a primitive AutosarDataType.

⌋

[TPS_SYST_03063] Definition of a composite target for SignalServiceTranslationElementProps ⌈The target of a SignalServiceTranslationElementProps is considered composite if either:

- SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a composite AutosarDataType and no reference for SignalServiceTranslationElementProps.element is given or
- SignalServiceTranslationEventProps.translationTarget refers to a VariableDataPrototype that is typed by a composite AutosarDataType and SignalServiceTranslationElementProps.element refers to an AutosarDataPrototype that is typed by a composite AutosarDataType.

⌋

The filter can be used to define a guard on the translated data. If the input data does no pass the filter then the impacted data is not translated. It depends on the kind of the translated data and the filter intention which filter types are supported:

[constr_3655] Supported filter types for primitive SignalService-TranslationElementProps

Imposition time: CP: IT_SysDesc

⌈If the target for SignalServiceTranslationElementProps is defined as primitive according to [TPS_SYST_03062] then the following values for dataFilterType are supported:

- always
- maskedNewDiffersMaskedOld
- maskedNewDiffersX
- maskedNewEqualsX
- never
- newIsOutside
- newIsWithin
- oneEveryN.

⌋

[constr_3656] Supported filter types for composite SignalService-TranslationElementProps

Imposition time: CP: IT_SysDesc

⌈If the target for SignalServiceTranslationElementProps is defined as composite according to [TPS_SYST_03062] then the following values for dataFilter-Type are supported:

- always
- never
- oneEveryN.

⌋

The SignalServiceTranslationElementProps.transmissionTrigger defines whether translated data actually triggers the sending of the SignalService-TranslationEventProps.translationTarget.

One aspect is when the translationTarget is composed out of several sources (see [TPS_SYST_03040]). In this section the discussion is about whether transmissionTrigger is also applicable for the case where the translation-Target is created out of one source.

If the Translation Application Software Component is implemented to perform the signal/service translation in a periodic way (see [TPS_SYST_03042] and [TPS_SYST_03043]) the transmissionTrigger can be used to define whether the

signal/service translation of that specific data actually triggers the sending or whether the sending is deferred to the periodic invocation of the Translation Application Software Component.

If the signal/service translation is configured to be done in a periodic way, then

- if the transmissionTrigger is set to *true* at least once then the translationTarget will be produced based on the periodicity AND additionally every time the input data arrives where transmissionTrigger is set to *true*
- if the transmissionTrigger is set to *false* for all members of the translationTarget then the translationTarget will be produced based on the periodicity ONLY.

If the signal/service translation is configured to be done in a NON periodic way (event driven only), then

- if the transmissionTrigger is set to *true* at least once then the translationTarget will be produced every time the input data arrives where transmissionTrigger is set to *true*
- if the transmissionTrigger is set to *false* for all members of the translationTarget then the translationTarget will NOT be sent at all (as there is no triggering defined, neither periodic nor via transmissionTrigger).

6.16.2.2 Mapping description from several sources

The mapping setup in figure 6.150 shows an example of a true multi-source mapping definition: The content of *dataZ3* is composed out of several sources: *dataX* and *dataY*.

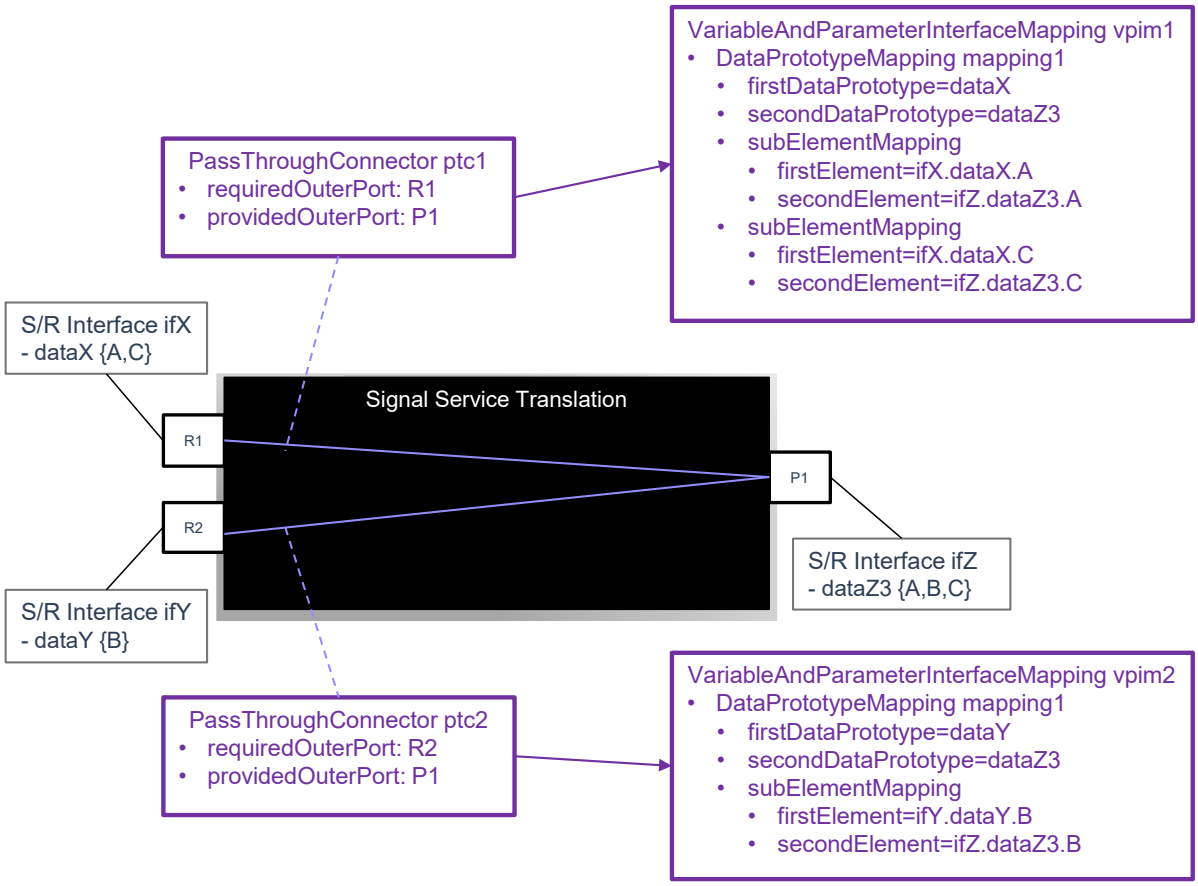

Figure 6.150: Mapping description using PassThroughSwConnectors and PortInterfaceMappings from several sources

The configuration of target data elements which are composed from several sources into the target data element is always defined from the perspective of the *target data element*.

The root of the *target data element* is defined by the reference `SignalServiceTranslationEventProps.translationTarget`. With this reference the `SwComponentPrototype` hosting the `signal/service translation` is defined, as well as the `PPortPrototype`, and the `translationTarget VariableDataPrototype`.

In case of `signal/service translation` from several sources the `translationTarget VariableDataPrototype` is composed out of several sources. But the definition of element translation attributes happens on the *target data elements*: `SignalServiceTranslationElementProps` is used to define `transmissionTrigger` and `filter` attributes for one leaf member of the `translationTarget` data element. In order to find the *source* data element to be translated the `PassThroughSwConnector`s and potential `VariableAndParameterInterfaceMapping`s have to be considered as well.

Since the `translationTarget` may be composed from several sources a restriction applies to the `swImplPolicy` of the involved source data.

A setup where several source data elements have the attribute swImplPolicy set to queued leads to runtime issues, as - at the point in time when the translationTarget shall be composed - all the source data elements need to be gathered. But in case of a source data element where swImplPolicy is set to queued a source data element queue might be empty. An empty queue might have a variety of reasons, e.g. the rate at which the queue is filled is slower than the rate the signal/service translation consumes the data.

Thus it would be very unlikely that all the source data element queues have at least one element to be read and consumed at the point in time when the signal/service translation tries to read them.

[TPS_SYST_03059] At most one queued source input in case of signal/service translation from several sources ⌈If the SignalServiceTranslationEventProps.translationTarget is composed out of several sources then at most one of these sources shall have the attribute swImplPolicy set to queued.⌋

[TPS_SYST_03060] Source input with queued semantics shall have transmissionTrigger set to true ⌈If a source data element has the attribute swImplPolicy set to queued then the corresponding target SignalServiceTranslationElementProps shall have the attribute transmissionTrigger set to *true*.

This shall be the only SignalServiceTranslationElementProps which has transmissionTrigger set to true in the scope of the SignalServiceTranslationEventProps.⌋

Note: [TPS_SYST_03060] implies that only one source can be queued and that source shall be the only source which has the transmissionTrigger set to *true*.

The queued reception of a source data element might cause an issue (e.g., unclear behavior in case of empty queues or queue overruns) with the periodic reception (see [TPS_SYST_03042]) or sending (see [TPS_SYST_03043]). Thus this combination is not supported by the signal/service translation.

[TPS_SYST_03061] No support for queued reception semantics in combination with periodic communication ⌈If in the scope of a SignalServiceTranslationEventProps.translationTarget the translationTarget has a periodic sending defined according to [TPS_SYST_03043] and/or at least one source data element has a periodic reception defined according to [TPS_SYST_03042] then none of the source data elements shall have a queued reception semantics.⌋

[TPS_SYST_03038] Definition of transmission triggers for translations with different sources ⌈The attribute SignalServiceTranslationElementProps.transmissionTrigger defines which translation elements (SignalServiceTranslationElementProps.element) contribute to the transmission triggering for the mapped payload.⌋

[TPS_SYST_03039] Full translation before transmission triggering ⌈In case there has been a transmission trigger caused by a source signal the signal/service translation shall process all other mapped source signals from the triggering source context (signal group or IPdu) before actually sending out the target.⌋

This basically means that for example if in the scenario depicted in figure 6.150 *dataZ3.B* were configured to trigger a transmission of *dataZ3*, then upon the reception event of *dataY* both *dataX* and *dataY* shall be received (by calling the corresponding RTE APIs), *dataZ3* shall be assembled out of the received *dataX* and *dataY* in order to fill *dataZ3.A*, *dataZ3.B*, and *dataZ3.C* with up-to-date information, and finally *dataZ3* shall be sent (by calling the corresponding RTE API).

[TPS_SYST_03040] Transmission trigger for translations with different sources ⌈If the attribute SignalServiceTranslationElementProps.transmissionTrigger equals *true* then
the reception of the respective source signal does cause the sending of the target (after all mapped sources from the same source context have been translated, see [TPS_SYST_03039]).⌋

[TPS_SYST_03041] No transmission trigger for translations with different sources ⌈If the attribute SignalServiceTranslationElementProps.transmissionTrigger is not defined or has the value *false* then
the reception of the respective source signal does not cause the sending of the target.⌋

6.16.2.3 Mapping description and data conversion

As the mapping for signal/service translation is implemented on Application level also the data conversion possibilities of the RTE may be used directly. [TPS_SWCT_01560] and [TPS_SWCT_01561] in the Software Component Template [4] define for which categorys of CompuMethods data conversion is supported.

6.16.2.4 Implementation of PassThroughSwConnectors

The definition of SignalServiceTranslationProps, PassThroughSwConnectors and potential VariableAndParameterInterfaceMappings specify the requirements what the signal/service translation shall do. The actual implementation needs to be done by code executed in the scope of a SwComponentPrototype typed by an AtomicSwComponentType.

Whether the SwComponentPrototype implementing the signal/service translation (typed by an AtomicSwComponentType) is put inside the CompositionSwComponentType which used to define the PassThroughSwConnectors or whether the SwComponentPrototype implementing the signal/service translation replaces the CompositionSwComponentType is left to the implementation.

The goal of this approach is that the model and the code of the signal/service translation AtomicSwComponentType can be automatically derived / generated from the requirements stated in the SignalServiceTranslationProps, PassThroughSwConnectors, and VariableAndParameterInterfaceMappings. This may include the definition of one or more

- ApplicationSwComponentType
- SwcInternalBehavior
- RunnableEntity
- VariableAccess

Figure 6.151 shows an example where the upper part illustrates the CompositionSwComponentType with PassThroughSwConnectors and VariableAndParameterInterfaceMappings. The lower part of the figure sketches an AtomicSwComponentType actually hosting the code which implements the signal/service translation.

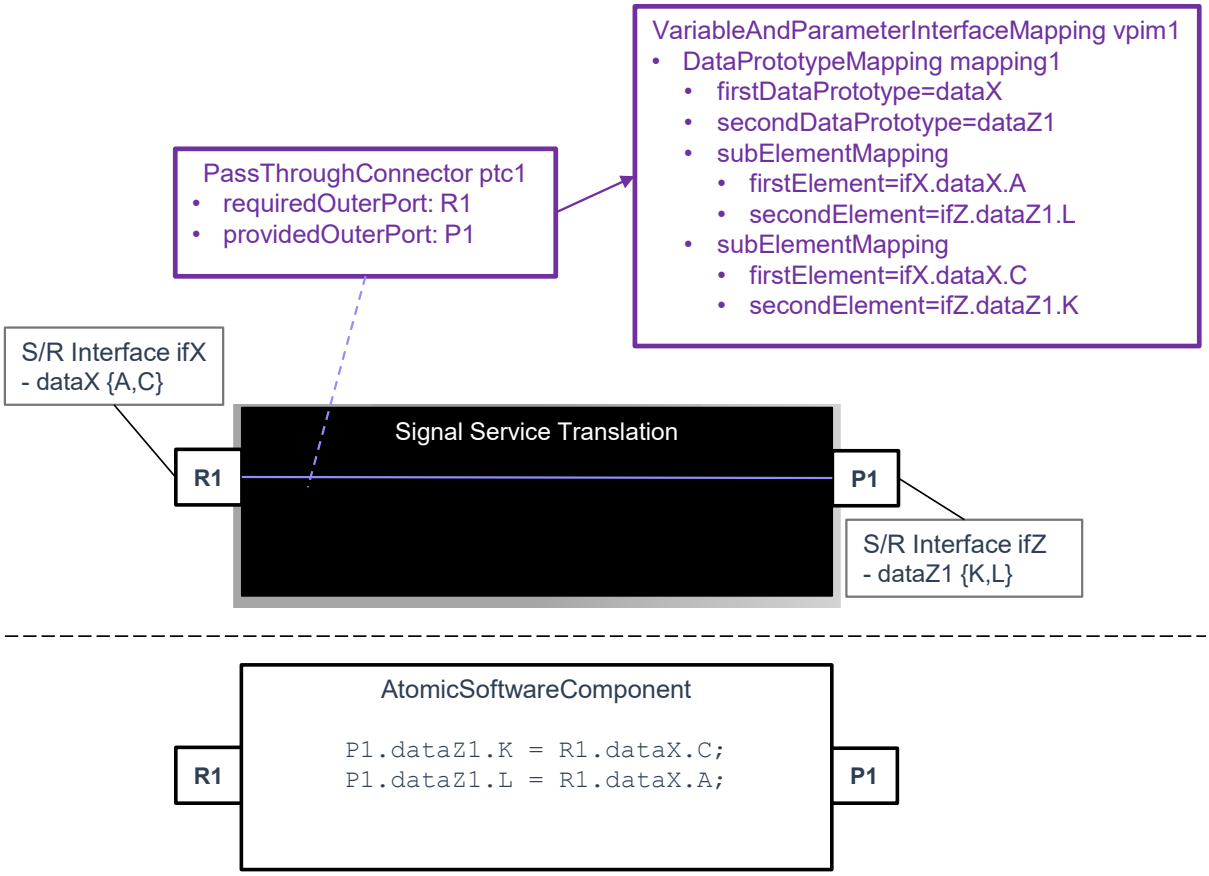

Figure 6.151: Implementation of PassThroughSwConnectors and PortInterfaceMappings

6.16.3 Service discovery control

The service discovery module [30] handles the offering/finding of service instances as well as the subscription to event groups. The behavior for each service instance may be controlled by Application software components via the BswM.

In scope of the `signal/service translation` the Translation Software Component needs to take control of the offering and subscribing to service instances. The general setup is illustrated in figure 6.152.

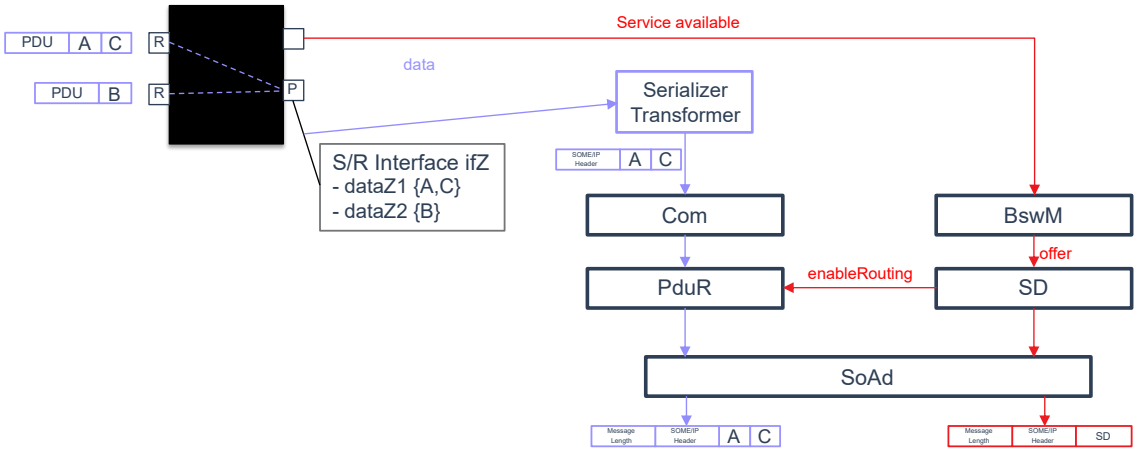

Figure 6.152: Interaction between translation SWC can Service Discovery

At which point in time a specific service instance (originating from `signal/service translation`) is actually offered / subscribed at the service discovery can be defined per service instance:

Possible approaches for service availability/subscription are:

- `translationStart` - start of ECU (see section 6.16.3.1)
- `allPartialNetworksActive` or `anyPartialNetworkActive` - availability of involved partial networks (see section 6.16.3.2)
- `serviceDiscovery` - availability of related service instance (see section 6.16.3.3)

The attribute `SignalServiceTranslationProps.serviceControl` defines the service instance control behavior.

Enumeration	**SignalServiceTranslationControlEnum**
Note	This enumeration allows to define how the service instance offer/subscribe control shall behave.
Aggregated by	SignalServiceTranslationProps.serviceControl
Literal	***Description***

▽

△

Enumeration	SignalServiceTranslationControlEnum
allPartialNetworks Active	Defines the start of service control when all specified partial networks are active. **Tags:** atp.EnumerationLiteralIndex=3 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
anyPartialNetwork Active	Defines the start of service control when any specified partial network is active. **Tags:** atp.EnumerationLiteralIndex=4 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
serviceDiscovery	Defines the start of service control when other service is available. **Tags:** atp.EnumerationLiteralIndex=2
translationStart	Defines the start of service control at translation start. **Tags:** atp.EnumerationLiteralIndex=0

Table 6.346: SignalServiceTranslationControlEnum

6.16.3.1 Service control at ECU start

The approach of service control at ECU start is to utilize the *AutoAvailable* / *AutoRequire* feature of the service discovery module where the service instance is automatically offered at startup of the ECU. In this case the translation software component does not need to interact with the BswM to control the service state of each individual service instance.

It is configurable per translated service event whether that event shall be automatically controlled at ECU start.

[TPS_SYST_03022] **`autoAvailable`** **setting for provided service instance with** **`translationStart`** ⌈For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart`, then the `ProvidedServiceInstance` owning the `EventHandler` referenced by `SignalServiceTranslationProps.controlProvidedEventGroup` shall have its `autoAvailable` attribute set to *true*.⌋

[TPS_SYST_03023] **`autoRequire`** **setting for required service instance with** **`translationStart`** ⌈For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart`, then the `ConsumedServiceInstance` owning the `ConsumedEventGroup` referenced by `SignalServiceTranslationProps.controlConsumedEventGroup` shall have its `autoRequire` attribute set to *true*.⌋

[TPS_SYST_03024] **`autoRequire`** **setting for required event groups of required service instance with** **`translationStart`** ⌈For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `translationStart`, then the `ConsumedEventGroup` referenced by `SignalServiceTranslationProps.controlConsumedEventGroup` shall have its `autoRequire` attribute set to *true*.⌋

6.16.3.2 Service control due to availability of partial networks

If the availability of the signal-based PDUs is controlled using Partial Networking then the respective translated services offers/subscriptions can only be activated if the specific partial networks are active. This relationship is used to control the availability of specific service instances.

In case there are several PNCs referenced in the role SignalServiceTranslationProps.controlPnc the activation of the translated service can be configured to either

- activate the service when ANY of the referenced PNCs is available (SignalServiceTranslationProps.serviceControl equals SignalServiceTranslationControlEnum.anyPartialNetworkActive)
- activate the service when ALL of the referenced PNCs are available (SignalServiceTranslationProps.serviceControl equals SignalServiceTranslationControlEnum.allPartialNetworksActive).

[constr_3545] Mandatory reference to a *Pnc* in case of anyPartialNetworkActive or allPartialNetworksActive

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationProps.serviceControl equals anyPartialNetworkActive or allPartialNetworksActive, then the reference SignalServiceTranslationProps.controlPnc shall point to at least one PncMappingIdent.⌋

[TPS_SYST_03025] Control of service instance in case of anyPartialNetworkActive or allPartialNetworksActive ⌈If the SignalServiceTranslationProps.serviceControl equals anyPartialNetworkActive or allPartialNetworksActive, then the respective service instance shall be configured to be controlled using the BswM/SD configuration.⌋

The ComM states relevant for the activation are defined in the SWS document [43].

Requested Partial Network: respective PNC is in ComM status COMM_PNC_REQUESTED or COMM_PNC_READY_SLEEP.

Released Partial Network: respective PNC is in ComM status COMM_PNC_PREPARE_SLEEP or COMM_PNC_NO_COMMUNICATION.

[TPS_SYST_03026] Monitoring of the requested *partial networks* status in case of anyPartialNetworkActive for provided service instance ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals anyPartialNetworkActive, then the translation software component shall monitor the status of the referenced *partial networks* (referenced via SignalServiceTranslationProps.controlPnc).

If at least one of the referenced *partial networks* is in the state `requested` then the translation software component shall *offer* the respective translated service instance.⌋

[TPS_SYST_03027] Monitoring of the `requested` *partial networks* status in case of `anyPartialNetworkActive` for required service instance ⌈For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `anyPartialNetworkActive`, then the translation software component shall monitor the status of the referenced *partial networks* (referenced via `SignalServiceTranslationProps.controlPnc`).

If at least one of the referenced *partial networks* is in the state `requested` then the translation software component shall *find* the respective translated service instance and *subscribe* to its event groups.⌋

[TPS_SYST_03056] Monitoring of the `released` *partial networks* status in case of `anyPartialNetworkActive` for required service instance ⌈For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `anyPartialNetworkActive`, then the translation software component shall monitor the status of the referenced *partial networks* (referenced via `SignalServiceTranslationProps.controlPnc`).

If all of the referenced *partial networks* are in the state `released` then the translation software component shall *unsubscribe* from its event groups.⌋

[TPS_SYST_03057] Monitoring of the `released` *partial networks* status in case of `anyPartialNetworkActive` for provided service instance ⌈For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `anyPartialNetworkActive` then the translation software component shall monitor the status of the referenced *partial networks* (referenced via `SignalServiceTranslationProps.controlPnc`).

If all of the referenced *partial networks* are in the state `released` then the translation software component shall stop *offer*ing the respective translated service instance.⌋

[TPS_SYST_02380] Monitoring of the requested partial networks status in case of `allPartialNetworksActive` for provided service instance ⌈For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `allPartialNetworksActive`, then the translation software component shall monitor the status of the referenced *partial networks* (referenced via `SignalServiceTranslationProps.controlPnc`). If all of the referenced *partial networks* are in the state `requested` then the translation software component shall *offer* the respective translated service instance.⌋

[TPS_SYST_02381] Monitoring of the requested partial networks status in case of `allPartialNetworksActive` for required service instance ⌈For a required translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `allPartialNetworksActive`, then the translation software component shall monitor the status of the referenced *partial networks* (referenced via

SignalServiceTranslationProps.controlPnc). If all of the referenced *partial networks* are in the state requested then the translation software component shall *find* the respective translated service instance and subscribe to its event groups.⌋

[TPS_SYST_02382] Monitoring of the released *partial networks* status in case of allPartialNetworksActive for required service instance ⌈For a required translated service instance, if the SignalServiceTranslationProps.serviceControl equals allPartialNetworksActive then the translation software component shall monitor the status of the referenced *partial networks* (referenced via SignalServiceTranslationProps.controlPnc).

If any of the referenced *partial networks* are in the state released then the translation software component shall unsubscribe from its event groups.⌋

[TPS_SYST_02383] Monitoring of the released *partial networks* status in case of allPartialNetworksActive for provided service instance ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals allPartialNetworksActive then the translation software component shall monitor the status of the referenced *partial networks* (referenced via SignalServiceTranslationProps.controlPnc).

If any of the referenced *partial networks* is in the state released then the translation software component shall stop *offer*ing the respective translated service instance.⌋

6.16.3.3 Service control due to availability of related service instance

There are scenarios where the transmission of signal-based PDUs is controlled by means of SOME/IP service discovery, i.e., the signal-based PDUs are effectively offered as events of a service and are only transmitted in case there are active subscriptions present. Since the payload of the signal-based PDUs is not serialized according to the SOME/IP transformer rules, even in such scenarios signal/service translation is required for the payload.

In this setup, however, the availability of the translated/target service instance depends on the availability of the corresponding source service instance(s). Note, that this dependency is irrespective of the actual direction of the translation:

- In case of *signal to service translation* the availability of the translated/target service instance (which exhibits a SOME/IP serialized payload) depends on the availability of the source service instance (which exhibits a signal-based payload).
- In case of *service to signal translation* the availability of the translated/target service instance (which exhibits a signal-based payload) depends on the availability of the source service instance (which exhibits a SOME/IP serialized payload).

Figure 6.153 illustrates an example setup showing exactly the above mentioned dependency in both directions as well as the approach to handle this dependency:

Service to signal translation (case a): A SOME/IP serialized source service instance shall be translated into a signal-based target service instance. Here the signal-based target service instance shall only be *offered* when the corresponding SOME/IP serialized source service instance is actually successfully subscribed to. Thus the approach is to initially issue a *find service* for the SOME/IP serialized source service instance. Upon the reception of an *offer service* a *subscribe event group* is sent and, finally, upon the reception of a *subscribe event group acknowledge* (i.e., upon the successful subscription) the signal-based translated target service instance is *offered*. For this setup the `ConsumedEventGroup` representing the SOME/IP serialized source service instance of *Ra* is used as `controlConsumedEventGroup`.

Signal to service translation (case b): A signal-based source service instance shall be translated into a SOME/IP serialized target service instance. Here the SOME/IP serialized target service instance shall only be *offered* when the corresponding signal-based source service instance is actually successfully subscribed to. The approach is analoguous to case *a* with source/target roles reversed. For this setup the `ConsumedEventGroup` representing the signal-based source service instance of *Rb* is used as `controlConsumedEventGroup`.

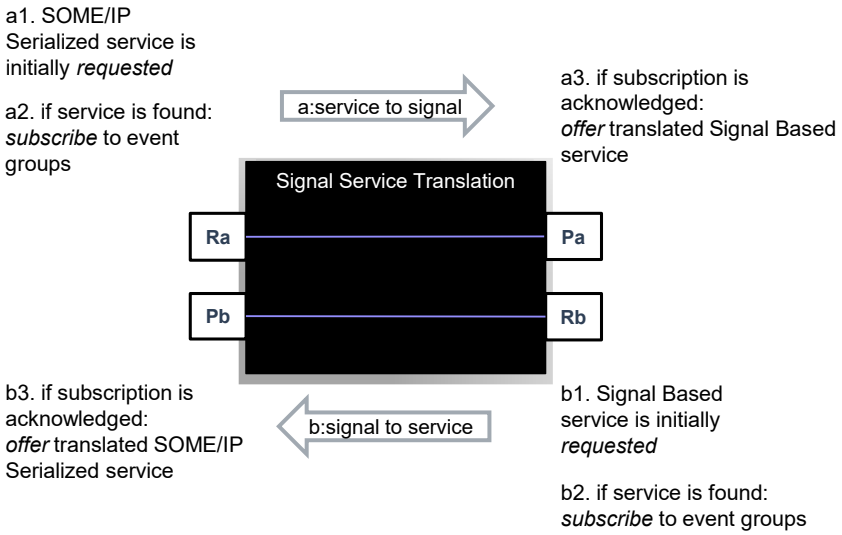

Figure 6.153: Translation based on service availability

[constr_3546] Mandatory reference to a `ConsumedEventGroup` in case of `serviceControl`

Imposition time: CP: `IT_SysDesc`

⌈For a provided translated service instance, if the `SignalServiceTranslationProps.serviceControl` equals `serviceDiscovery` then the reference `SignalServiceTranslationProps.controlConsumedEventGroup` shall point to at least one `ConsumedEventGroup`.⌋

In case SignalServiceTranslationProps.serviceControl is set to serviceDiscovery then the ConsumedEventGroups referenced in role controlConsumedEventGroup as well as the owning ConsumedServiceInstances are required for the availability of the provided translated service instance and thus have to be *found* and *subscribed to* by means of *find service* and *subscribe event group*. This *finding* and *subscription* is automatically (i.e., without the need of any SWC intervention) achieved by having the ConsumedServiceInstance and the ConsumedEventGroup autoRequire their instances.

[TPS_SYST_03028] *Auto require* for controlConsumedEventGroup in case of service instance with serviceControl ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals serviceDiscovery then every ConsumedEventGroup referenced in SignalServiceTranslationProps.controlConsumedEventGroup shall have the autoRequire attribute set to *true*.⌋

[TPS_SYST_03058] *Auto require* for ConsumedServiceInstance in case of service instance with serviceControl ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals serviceDiscovery then the every ConsumedServiceInstance owning at least one of the controlConsumedEventGroups shall have the autoRequire attribute set to *true*.⌋

[TPS_SYST_03029] *Offer* for a provided translated service instance with serviceControl ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals serviceDiscovery and the *subscribe* to **all** referenced controlConsumedEventGroups was successful, then the respective translated target service instance shall be *offer*ed. This applies for both, signal-service-translation as well as service-signal-translation.⌋

[TPS_SYST_03030] *Stop offer* for a provided service instance with serviceControl ⌈For a provided translated service instance, if the SignalServiceTranslationProps.serviceControl equals serviceDiscovery and the *subscription* of **at least one** controlConsumedEventGroup is not available, then a *stop offer* of the respective translated target service instance shall be issued. This applies for both, signal-service-translation as well as service-signal-translation.⌋

6.16.4 Translation behavior

There are two possible ways to define behavioral aspects for the signal/service translation use-case:

- COM-Stack
- Translation Application Software Component

6.16.4.1 COM-Stack translation behavior

The Classic platform COM-Stack can be configured to have an own periodic behavior for periodic sending and reception/time-out monitoring. In this case the existing COM-Stack definition of behavior is used.

Example features are:

- TransmissionModeTiming.cyclicTiming
- TransmissionModeTiming.eventControlledTiming
- IPduTiming.minimumDelay
- ISignalPort.timeout

Thus it is possible to register the Translation Application Software Component to be notified when data arrives and then perform the translation operation solely driven by the notifications from the COM-Stack.

There are use-cases where the COM-Stack can not be used to perform the specific behavioral aspects because the Transformers are required to have access to the raw received and sent data. In these cases the usage of the Translation Application Software Component behavior is required.

6.16.4.2 Translation Application Software Component translation behavior

If the Translation Application Software Component shall implement specific behavioral aspect these have to be defined at the definition of the CompositionSwComponentType which has the signal/service translation input and output PortPrototypes as well as the PassThroughSwConnectors.

For the periodic behavior the definition of SenderComSpec.transmissionProps.dataUpdatePeriod and ReceiverComSpec.receptionProps.dataUpdatePeriod define the expected periods for data update and check.

[TPS_SYST_03042] Periodic call in case of ReceiverComSpec.ReceiverComSpec.receptionProps.dataUpdatePeriod ⌈If the signal/service translation CompositionSwComponentType has a RPortPrototype defined with a ReceiverComSpec.receptionProps.dataUpdatePeriod defined then the signal/service translation software component implementation shall periodically call the respective Rte reception API with the defined period.⌋

[TPS_SYST_03043] Periodic call in case of SenderComSpec.transmissionProps.dataUpdatePeriod ⌈If the signal/service translation CompositionSwComponentType has a PPortPrototype defined with a SenderComSpec.transmissionProps.dataUpdatePeriod defined then the signal/service translation software component implementation shall periodically call the respective Rte sending API with the defined period.⌋

Data filtering:

If there is a filter defined at the NonqueuedReceiverComSpec then the evaluation of this DataFilter is performed in the COM-Stack. Thus the COM-Stack filtering usually can not be applied when there are transformers involved because the state machines of E2E transformers need to receive every message.

If a data filtering shall be applied *after* the data transformation inside the signal/service translation software component then there is the possibility to define a DataFilter at the SignalServiceTranslationElementProps in the role filter.

[TPS_SYST_03051] Data filter inside the signal/service translation software component ⌈If there is a SignalServiceTranslationElementProps.filter defined this filtering shall be implemented inside the signal/service translation software component.⌋

6.16.5 End-to-End considerations

In case there is an E2E header attached and/or a secure communication defined the translation needs to break the transport chain and re-calculate the E2E measures (CRC/MAC). The CRC/MAC is calculated over the uint8 array representation of the payload data. Since the signal/service translation changes the data layout, also the uint8 array representation of the payload data changes resulting in a different CRC/MAC value.

6.16.5.1 Safety

Due to the architectural approach of the signal/service translation the safety aspects can be configured using existing mechanisms. Thus it is possible to define dedicated safety end-to-end profiles for both the signal-based part, and the service-based part. The translation software component then links the two parts together and handles the behavioral aspects of the translation.

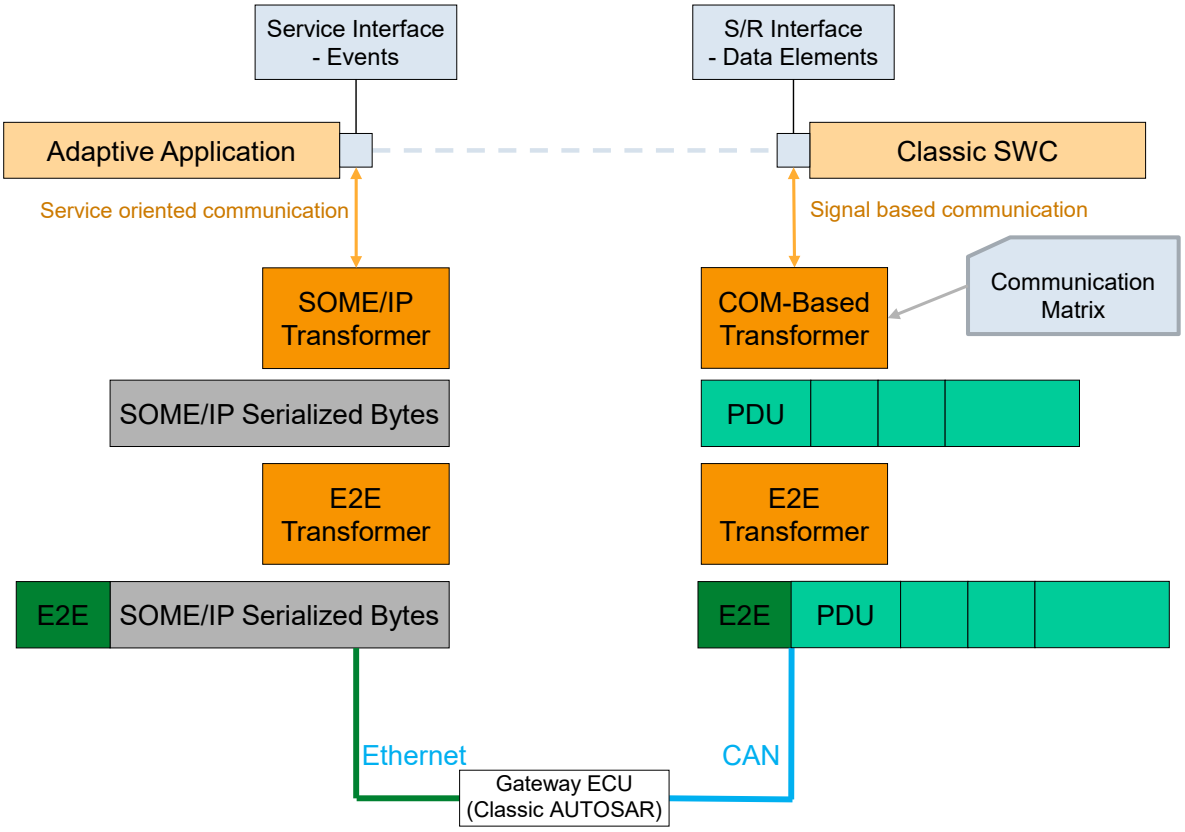

Figure 6.154: `Signal/Service Translation` and Safety

The attribute `SignalServiceTranslationEventProps.safeTranslation` is used to explicitly require that both ends of the translation shall be configured in a safe transport way and that the translation software component shall handle the translation activity in a end-to-end preserving way.

[TPS_SYST_03044] Handling of safe `signal/service translation` in one software component ⌈It is required that the `signal/service translation` (and vice versa) of one Service/SignalGroup pair which are mapped to each other, shall be handled in one software component to also cover a closed mapping from one E2E profile to another, if necessary. The `signal/service translation` of different (independent) Services/SignalGroups may be handled by different software component.⌋

[TPS_SYST_03036] `PortAPIOption` for `safeTranslation RPortPrototype` ⌈If `SignalServiceTranslationEventProps.safeTranslation` is set to *true* then a `PortAPIOption` referring to the `RPortPrototype` shall exist and the `PortAPIOption.errorHandling` attribute shall be set to `transformerError-Handling`.⌋

[TPS_SYST_03037] `PortAPIOption` for `safeTranslation PPortPrototype` ⌈If `SignalServiceTranslationEventProps.safeTranslation` is set to *true* then a `PortAPIOption` referring to the `PPortPrototype` shall exist and

the PortAPIOption.transformerStatusForwarding attribute shall be set to transformerStatusForwarding.⌋

[TPS_SYST_03045] Support for safe signal/service translation ⌈The translation of E2E protected data shall be supported in both directions, signal-service-translation and service-signal-translation.⌋

[TPS_SYST_03046] Support for safe signal/service translation with same or different E2E profiles ⌈The translation of E2E protected data shall support the occurrence of

- the same E2E profile on both sides of the communication and
- different E2E profiles on each side of the communication.

⌋

[TPS_SYST_03047] 1:n mapping for E2E protected data ⌈It shall be possible to map the same E2E protected source data to several E2E protected target data (1:n).⌋

[TPS_SYST_03048] E2E protected target out of E2E protected sources ⌈The content of one E2E protected target shall only be composed out of data from E2E protected sources.⌋

The rationale for [TPS_SYST_03048] is to support the use-case where target data shall be E2E protected and is composed from several sources.

[TPS_SYST_03049] No translation of not OK E2E protected composed data ⌈If a E2E protected source data is mapped into a composed E2E protected target data and if the E2E-Check for the source data returns any E2E error (not *E_OK*) then
this source data shall not be forwarded to the respective target data and (if applicable) shall not trigger the transmission of the target.⌋

If source data is not verified as *E_OK* it is not translated into a composed target. If the translated E2E protected data comes from several sources there may occur correlation and synchronicity issues during translation.

[TPS_SYST_03031] Sufficient ASIL level of translation software component ⌈If the SignalServiceTranslationEventProps.safeTranslation equals true then the implementation of the translation software component shall fulfill a sufficient ASIL.⌋

[constr_3548] EndToEnd profile for both ends of safeTranslation

Imposition time: CP: IT_SysDesc

⌈If the SignalServiceTranslationEventProps.safeTranslation equals true then both, the signal-based payload as well as the service-oriented payload shall have an EndToEnd profile defined.⌋

[TPS_SYST_03032] Data transmission in case of *E_OK* safe signal reception ⌈`Signal/service translation` shall check the end-to-end status of every received payload. If the safety transformer returns *E_OK* for the received payload then the data shall be forwarded to the respective sending of the translation software component.⌋

Error handling:

[TPS_SYST_03033] No data transmission in case of reception timeout ⌈If no message is received within the specified message cycle time (timeout is detected), then no data shall be translated to the respective sending of the translation software component.⌋

[TPS_SYST_03034] Handling safe signal reception ⌈

Source error	Forwarding status code	Comment
E_OK	STD_E_FORWARDCODE_OK	–
E_SAFETY_VALID_REP	STD_E_FORWARDCODE_STATUS_REP	–
E_SAFETY_VALID_SEQ	STD_E_FORWARDCODE_STATUS_SEQ	–
E_SAFETY_VALID_ERR	STD_E_FORWARDCODE_STATUS_ERR	–
E_SAFETY_VALID_NND	*NO TRANSLATION*	No data received, thus no translation
E_SAFETY_NODATA_OK	STD_E_FORWARDCODE_OK	Statemachine is in state *No Data*, but received data is ok
E_SAFETY_NODATA_REP	STD_E_FORWARDCODE_STATUS_REP	Statemachine is in state *No Data*, but received data has repeated counter
E_SAFETY_NODATA_SEQ	STD_E_FORWARDCODE_STATUS_SEQ	Statemachine is in state *No Data*, but received data has wrong sequence counter
E_SAFETY_NODATA_ERR	STD_E_FORWARDCODE_STATUS_ERR	Statemachine is in state *No Data*, but received data has CRC Error
E_SAFETY_NODATA_NND	*NO TRANSLATION*	–
E_SAFETY_INIT_OK	STD_E_FORWARDCODE_OK	–
E_SAFETY_INIT_REP	STD_E_FORWARDCODE_STATUS_REP	–
E_SAFETY_INIT_SEQ	STD_E_FORWARDCODE_STATUS_SEQ	–
E_SAFETY_INIT_ERR	STD_E_FORWARDCODE_STATUS_ERR	–
E_SAFETY_INIT_NND	*NO TRANSLATION*	No data received, thus no translation
E_SAFETY_INVALID_OK	STD_E_FORWARDCODE_OK	Statemachine is in status *invalid*, but data itself is valid, thus no error replication
E_SAFETY_INVALID_REP	STD_E_FORWARDCODE_STATUS_REP	–
E_SAFETY_INVALID_SEQ	STD_E_FORWARDCODE_STATUS_SEQ	–
E_SAFETY_INVALID_ERR	STD_E_FORWARDCODE_STATUS_ERR	–
E_SAFETY_INVALID_NND	*NO TRANSLATION*	No data received, thus no translation
E_SAFETY_SOFT_RUNTIMEERROR	*NO TRANSLATION*	–
E_SAFETY_HARD_RUNTIMEERROR	*NO TRANSLATION*	–

The behavior of transformer status code forwarding (Error to Forwarding status code mapping)

⌋

Please note that forwarding status represents the E2E status part of the transformer code provided by Xfrm (see [SWS_Xfrm_00032]) defined in [44]. The E2E status can be extracted from the source error by performing a bitwise AND operation with 0xFU on the received source error code. (e.g. E_SAFETY_VALID_REP & 0xFU -> STD_E_FORWARDCODE_STATUS_REP).

6.16.5.2 Security

The security aspects are handled in the communication stack of Classic platform. There are two technologies for secure communication available:

- Secure Onboard Communication (SecOC) [45]
- Transport Layer Security (TLS) [46]

Which security technology is used for a translated service instance is up to the communication design. The security settings have to match between the providers and the consumers on the network.

As on Classic platform the translation happens on the application software component level it is possible to have equal, similar, or different security technologies configured on signal and service level (which is part of the COM-Stack configuration).

It is for instance well possible to have signals coming from a CanFD network secured with SecOC and translated into a service which is secured using TLS on Ethernet. But also using SecOC on Ethernet is possible.

In any case, if secure communication is involved together with `signal/service translation` the translation needs to break the transport chain and re-secure the payload (e.g. re-calculate the MAC for the newly serialized payload).

The attribute `SignalServiceTranslationEventProps.secureTranslation` is used to explicitly require that both ends of the translation shall be configured in a secure transport way and that the translation software component shall handle the translation activity in a security-preserving way.

[constr_3549] Secure payload for both ends in case of `secureTranslation`

Imposition time: CP: `IT_SysDesc`

⌈If the `SignalServiceTranslationEventProps.secureTranslation` equals true then both, the signal-based payload as well as the service-oriented payload shall have a secure communication defined.⌋

7 Data Transformation

7.1 Outline

The transmission of data over a communication bus requires some effort to convey the information about the nature of the transmitted data from the sender to the receiver. Both sides need to agree on this part or else the communication will fail.

This aspect is complicated by the fact that in most cases it is uncommon to transmit information in an atomic manner piece by piece. For the sake of properly utilizing the available communication resources, pieces of data that may or may not have any semantic relationship with each other are packed into a single transmission unit.

In this case, the receiver does not only have to be informed about the nature of the individual pieces of information but also about the packing of these pieces into the transmission unit.

There are different approaches of how this goal can be achieved, these are described in the following sub-chapters.

7.1.1 Configuration of the Communication Layout

Use a configurable software package on both the sender and the receiver side that can adapt to virtually any possible packing of data. In this case the packing shall be described in machine-readable form on a very detailed level in order to allow for the communication software to adapt to it.

For the sake of this argument, it doesn't really matter whether the adaption to the configuration is done at run-time or whether the configuration ends up in dedicated source code. The point is that the very detailed machine-readable configuration description is required to exist.

This approach used to be one of the pillars of the AUTOSAR standard as it entitled the players in the business with a maximum amount of flexibility and especially the OEMs are able to develop specific patterns for the design of their communication matrices that can, despite the diversity, be expressed with this approach.

This approach also facilitates the monitoring of transmission during development and deployment of the automotive software because monitoring tools can use the same configuration information to set themselves up for the task. This aspect is very important for debugging and quality assurance.

The downside, however, is that the act of laying out pieces of information in a limited number of transmission units becomes cumbersome and time-consuming. This effect becomes even more prominent with the advent of more advanced communication technologies that allow for a much bigger payload in single transmission units.

7.1.2 Data Transformation by Software

Don't care about the individual layout of information on the bus and let a piece of software take care of marshaling data onto the communication bus on the sender side and the reverse process on the receiver side.

This approach gains attractiveness in an environment where large and complicated pieces of information need to be transmitted.

Of course, in order to make this approach work it is necessary to standardize the behavior of the marshaling software to the necessary extent such that sender and receiver agree on how data needs to be processed.

With this approach, the amount of configuration can be reduced dramatically at the potential expense of efficiency and code size.

But this is not the end of the story as the idea of letting software take care of data "manipulation" can **following pretty much the same pattern** be utilized for further use cases:

End-to-end Protection Data is wrapped into a harness of meta-data that allows for checking data integrity at the receiver side.

Data Security Data is cryptographically processed such that it shall become impossible for unauthorized parties to intercept the communication process.

In other words, the approach is not limited to marshaling of data but can in the same way also be used for an array of other useful data transformations. This is why the terminology in this regard is not limited to the marshaling but to data transformation in general, hence the term **`Data Transformer`** is coined.

`Data Transformer`s can be chained such that, on the sender side, one `Data Transformer` picks up the result of the transformation of another `Data Transformer` and applies a specific transformation to the already processed data.

The receiver then is required to apply the `Data Transformer`s in reverse order in order to finally yield the actual data and provide it to the consumer (e.g. an `ApplicationSwComponentType`).

A basic principle of the `Data Transformer` approach, however, is that the `Data Transformer` is only responsible for the actual data transformation but **not** concerned about the communication of data. This can be taken care of by other software modules.

In total, the second approach provides a sufficient level of utility that it becomes part of the AUTOSAR standard. This chapter lays out the details of how `Data Transformer`s can be used in the context of this document.

Further information can also be found in the SWS RTE [23].

7.2 Use Cases

This chapter describes Transformer use cases that are supported by AUTOSAR.

7.2.1 Transmission of large composite data types over networks with large PDUs (e.g Ethernet)

With a serializing transformer, it is not necessary any more to map the atomic sub-elements of composite data types to individual signals in the RTE. The sending application SWC sends the composite data element using Sender/Receiver communication and hands the data over to the RTE. Then the complex data get transformed to a linear byte array and handed over to Com which sends the data to the receiving ECU. There, the Com stack receives the serialized data and notifies the RTE. The Rte reads the data and calls the deserializing transformer. The deserializing transformer transforms them back into the composite data element and gives the result to the RTE. The receiving SWC can now read the data and access it in the same form the sending SWC has sent them.

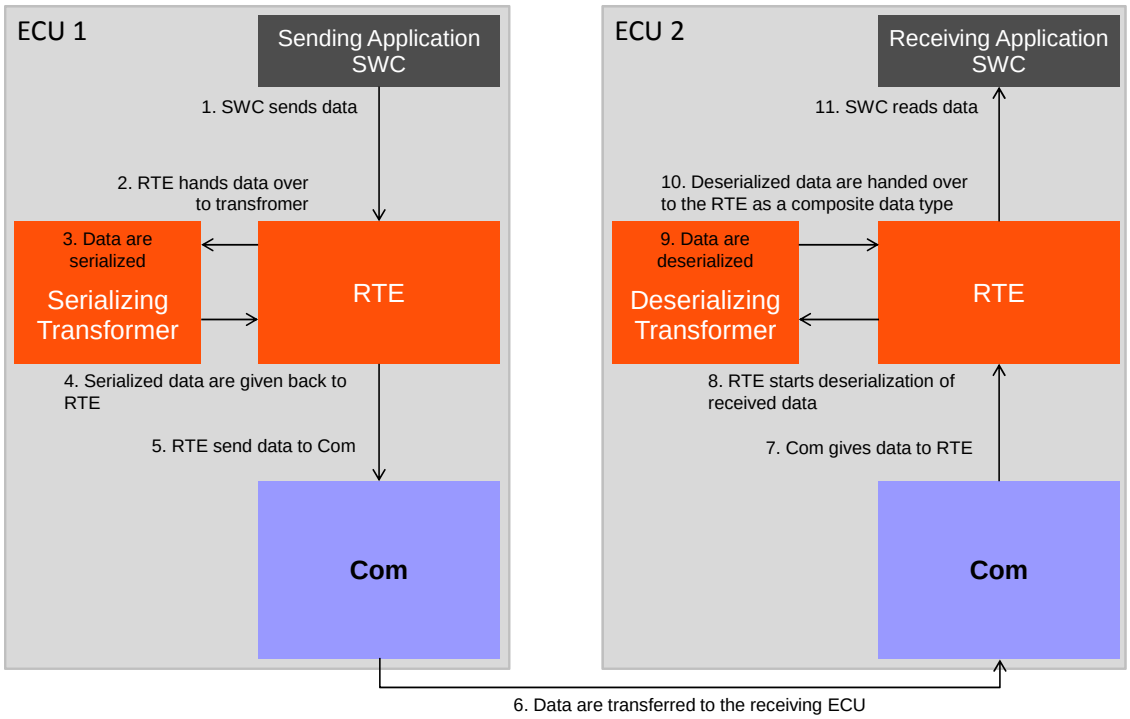

Figure 7.1: Transformer Use Case: Transmission of large composite data types over networks with large PDUs (e.g Ethernet)

7.2.2 Support of transmission from one sender to multiple receivers with Signal Fan-out

If a signal fan-out is configured in the System Description, the RTE has to hand over the data which should be transmitted multiple times to the Com stack. This is the case if multiple `ISignal`s reference the same `SystemSignal` in the System Description.

For each `ISignal` the following steps have to be performed individually:

- transform the data
- hand it over to COM

Every receiver has to deserialize the ISignal in its transformer independently.

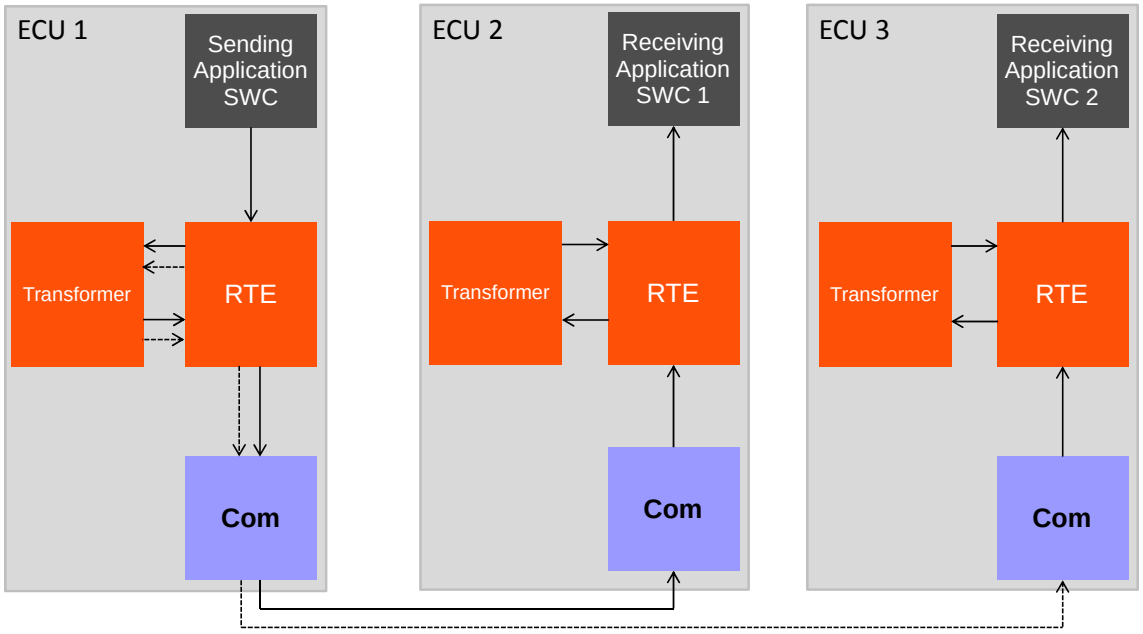

Figure 7.2: Transformer Use Case: RTE Fanout

7.2.3 Support of transmission from one sender to multiple receivers with PDU Fan-out

The transformation of inter-ECU Sender/Receiver communication should also work together with configurations that include `Pdu` fan-outs inside the COM stack (PduR fan-out). This is the case if multiple `PduTriggering`s reference the same `Pdu` in the System Description. In that scenario the data are sent by the sending application SWC to the RTE and transformed by the data transformer which is called by the RTE. Then the RTE hands the data over the Com. This happens only once. Due to the Pdu fan-out, the PduR sends the data multiple times to the Bus Interfaces using different Pdus.

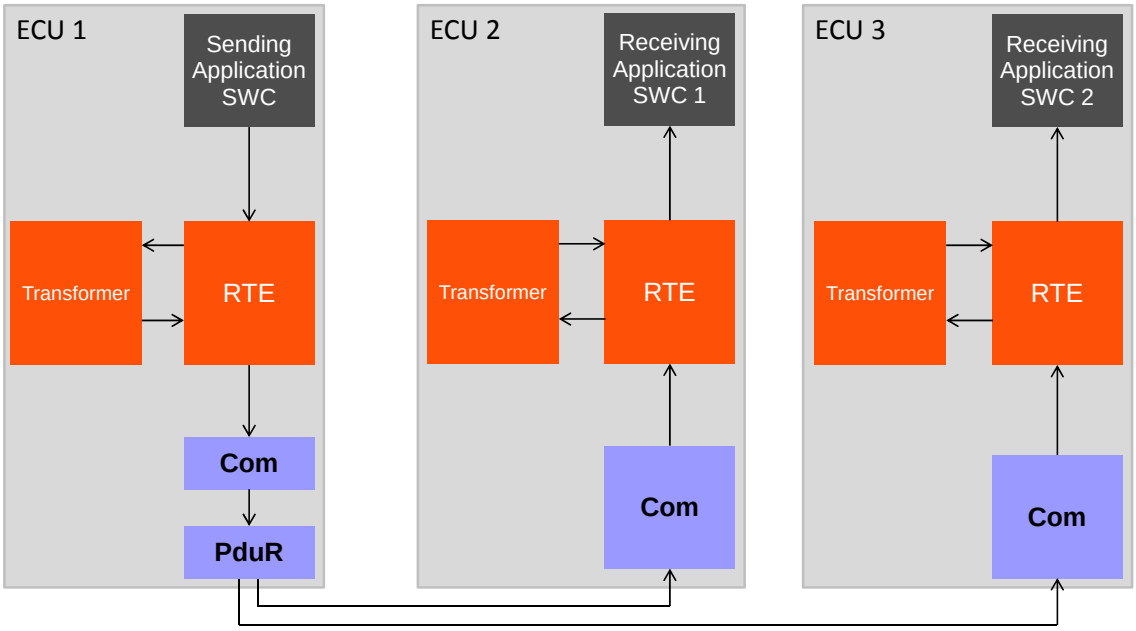

Figure 7.3: Transformer Use Case: PduR Fanout

7.2.4 Transformer Chaining

It is possible to chain multiple transformers. The output of one transformer then will be the input of the next transformer in the chain. Transformer for serialization data, for encrypting, digitally signing or compressing data can be implemented and used together. Such architecture could be used to assemble a system, where you can flexibly add functionality like compression or encryption to a serialized stream. In AUTOSAR the E2E-protection is implemented by an additional serializer which is appended to the chain.

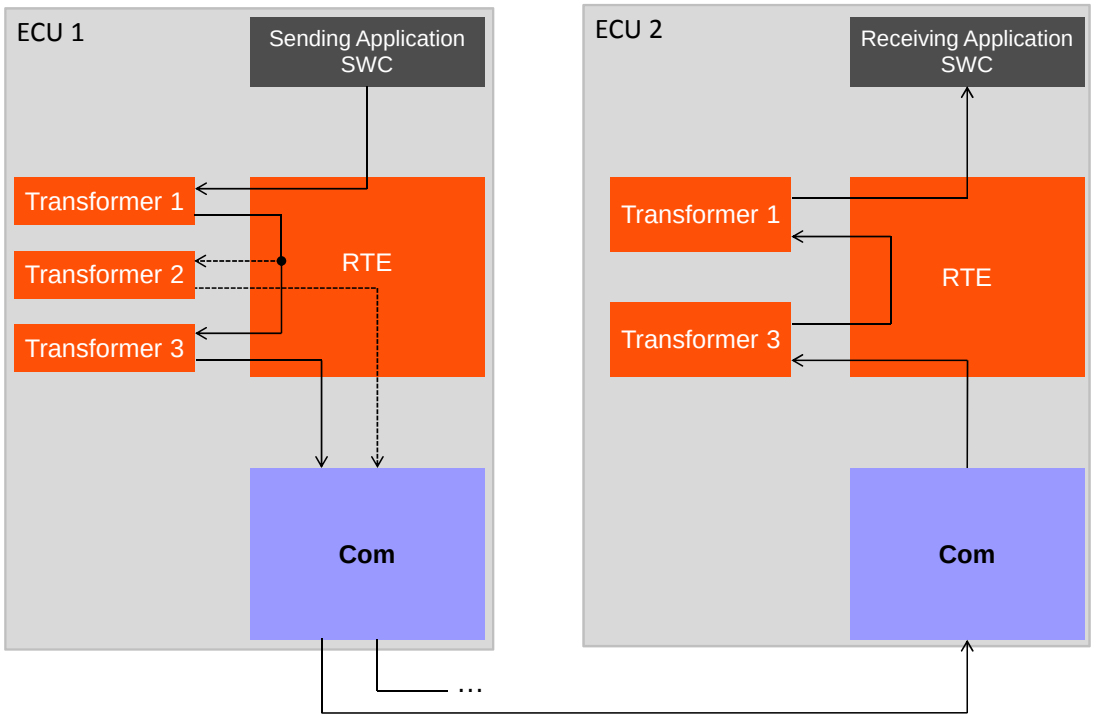

Figure 7.4: Transformer Use Case: Transformer Chain

7.2.5 Signal Group Based interaction of the transformer with the Com module

An initial transformer (serializer) performs the serialization according to the `ISignalToIPduMapping` from the system description. For each application data element the corresponding mapping to an `ISignalIPdu` position is respected. After the transformation chain is processed the serialized data is provided to the Com module. The Com module can have a signal based transmission mode selection defined and determines the respective transmission mode to be applied.

7.3 Transformer configuration

As a transformer provides well defined function signatures per each communication relation (`ISignal` based), which is marked for transformation, the function signature is NOT dependent from the transformation technology used, but only from the transmitted data elements (Client/Server operation signature or Sender/Receiver interface signature). The output of a transformer will be always a linear byte array.

Configuration of data transformation consists of three parts:

1. definition of the transformer chains with their transformers
2. configuration which communication is subject to transformation
3. configuration of the transformer properties for the transformed communication

The configuration of single transformers and whole transformer chains is shown in figure 7.5.

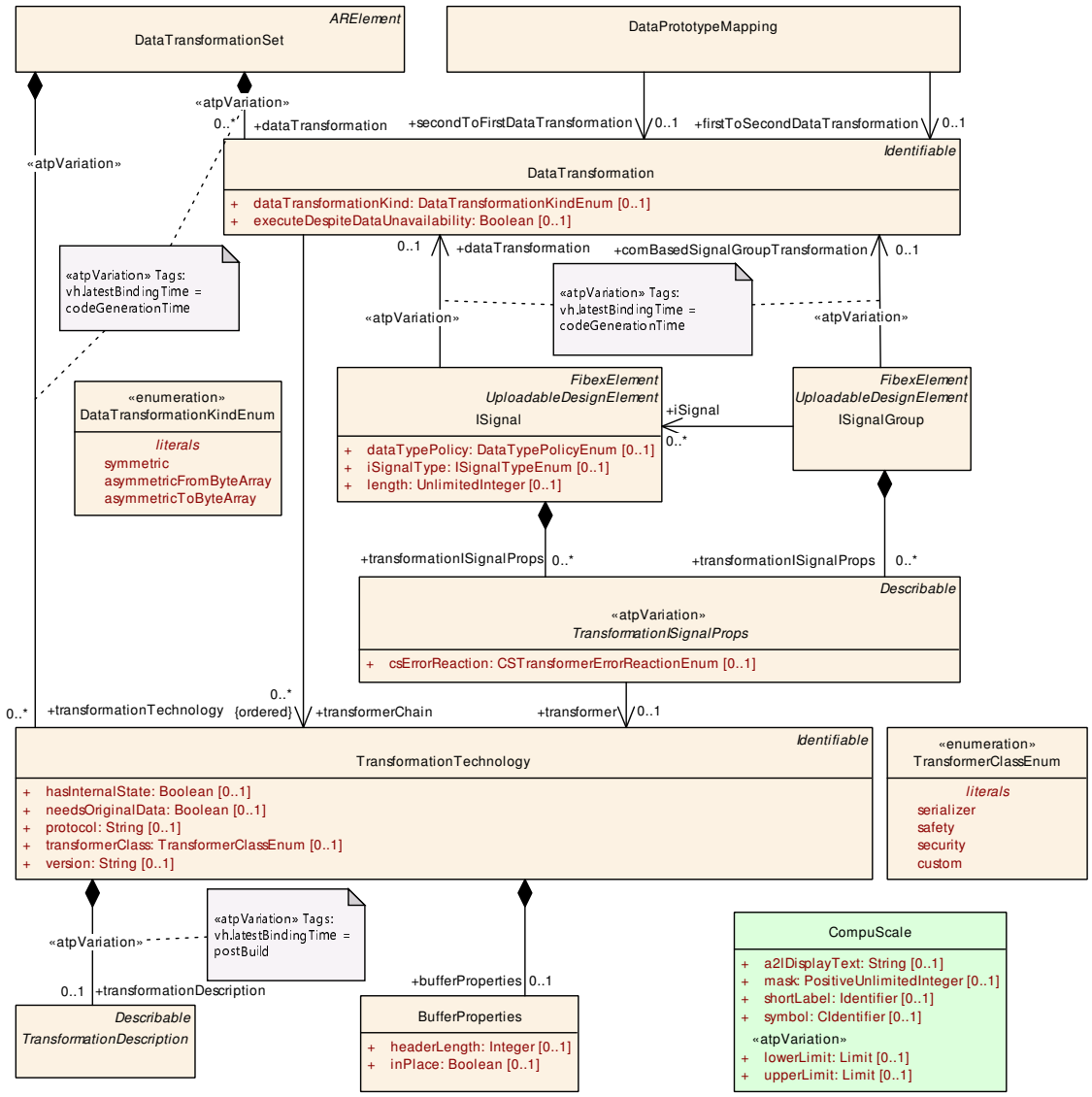

Figure 7.5: Configuration of transformers and transformer chains

The `DataTransformationSet` acts as a central container for the configuration of data transformation.

Class	**DataTransformationSet**			
Note	This element is the system wide container of DataTransformations which represent transformer chains. **Tags:** atp.recommendedPackage=DataTransformationSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DataTransformationSet**			
data Transformation	DataTransformation	*	aggr	This container consists of all transformer chains which can be used for transformation of data communication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataTransformation, data Transformation.shortName, dataTransformation.variation Point.shortLabel vh.latestBindingTime=codeGenerationTime
transformation Technology	Transformation Technology	*	aggr	Transformer that is used in a transformer chain for transformation of data communication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=transformationTechnology, transformation Technology.shortName, transformation Technology.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime

Table 7.1: DataTransformationSet

Class	**DataTransformation**			
Note	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.dataTransformation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data Transformation Kind	DataTransformationKind Enum	0..1	attr	This attribute controls the kind of DataTransformation to be applied.
executeDespite Data Unavailability	Boolean	0..1	attr	Specifies whether the transformer chain is executed even if no input data are available.
transformer Chain (ordered)	Transformation Technology	*	ref	This attribute represents the definition of a chain of transformers that are supposed to be executed according to the order of being referenced from DataTransformation.

Table 7.2: DataTransformation

[TPS_SYST_02030] The `DataTransformationSet` contains all transformer chains ⌈The `DataTransformationSet` contains transformer chains represented by `DataTransformation` elements.⌋

For each transformer chain it can be decided via the attribute `executeDespite-DataUnavailability` whether the RTE should try to execute the transformers of the transformer chain, even when no data are available as input. e.g. the queue is empty or there was an error in the COM stack. This is needed when no data are available but a transformer has to be executed anyway because it maintains an internal state which has to be updated to consider that data was expected but not available. This might be used in transformers which maintain an internal state. Of course the specifications and implementations of all transformers in the chain have to be able to cope with execution without valid input data.

[constr_3208] executeDespiteDataUnavailability usage restriction for received ISignals

Imposition time: CP: IT_SysDesc

⌈For each SystemSignal, the collection of ISignals that refer to the SystemSignal and also refer to an ISignalPort on the same EcuInstance with communicationDirection = in shall not include an ISignal that references a DataTransformation where executeDespiteDataUnavailability is set to true.⌋

In other words: There shall be no transformer chain which "belong" to the same SystemSignal due to signal fan-in where the attribute executeDespiteDataUnavailability is set to true.

[constr_9273] Existence of DataTransformation.executeDespiteDataUnavailability

Imposition time: CP: IT_SysDesc

⌈For each DataTransformation, the attribute executeDespiteDataUnavailability shall exist.⌋

[constr_9274] Existence of DataTransformation.transformerChain

Imposition time: CP: IT_SysDesc

⌈For each DataTransformation, at least one reference to TransformationTechnology in the role transformerChain shall exist.⌋

[TPS_SYST_02031] A transformer is represented by a TransformationTechnology ⌈A transformer is represented by a TransformationTechnology.⌋

Class	**TransformationTechnology**			
Note	A TransformationTechnology is a transformer inside a transformer chain. **Tags:** xml.namePlural=TRANSFORMATION-TECHNOLOGIES			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.transformationTechnology			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bufferProperties	BufferProperties	0..1	aggr	Aggregation of the mandatory BufferProperties. **Tags:** atp.Splitkey=bufferProperties
hasInternal State	Boolean	0..1	attr	This attribute defines whether the Transformer has an internal state or not.
needsOriginal Data	Boolean	0..1	attr	Specifies whether this transformer gets access to the SWC's original data.
protocol	String	0..1	attr	Specifies the protocol that is implemented by this transformer.

▽

△

Class	**TransformationTechnology**			
transformation Description	Transformation Description	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the Transformer Description. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=transformationDescription, transformation Description.variationPoint.shortLabel vh.latestBindingTime=postBuild
transformer Class	TransformerClassEnum	0..1	attr	Specifies to which transformer class this transformer belongs.
version	String	0..1	attr	Version of the implemented protocol.

Table 7.3: TransformationTechnology

[constr_9275] Existence of TransformationTechnology.bufferProperties

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, a BufferProperties element shall be aggregated by TransformationTechnology in the role bufferProperties.⌋

[constr_9276] Existence of TransformationTechnology.protocol

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute protocol shall exist.⌋

[constr_9277] Existence of TransformationTechnology.transformerClass

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute transformerClass shall exist.⌋

[constr_9278] Existence of TransformationTechnology.version

Imposition time: CP: IT_SysDesc

⌈For each TransformationTechnology, the attribute version shall exist.⌋

Enumeration	**TransformerClassEnum**
Note	Specifies the transformer class of a transformer.
Aggregated by	TransformationTechnology.transformerClass
Literal	***Description***
custom	The transformer is a custom transformer. **Tags:** atp.EnumerationLiteralIndex=0
safety	The transformer is a safety transformer. **Tags:** atp.EnumerationLiteralIndex=1
security	The transformer is a security transformer. **Tags:** atp.EnumerationLiteralIndex=2
serializer	The transformer is a serializing transformer. **Tags:** atp.EnumerationLiteralIndex=3

Table 7.4: TransformerClassEnum

[constr_3265] `TransformationTechnology.hasInternalState` setting for an E2E transformer

Imposition time: CP: `IT_SysDesc`

⌈The value of `hasInternalState` shall be set to true for a `TransformationTechnology` with `transformerClass` set to `safety`.⌋

[constr_3266] `TransformationTechnology.hasInternalState` setting for a SOME/IP Transformer

Imposition time: CP: `IT_SysDesc`

⌈The value of `hasInternalState` shall be set to true for a SOME/IP Transformer if the `ISignal` that is referencing this transformer is mapped into an `ISignalIPdu` and this `ISignalIPdu` is referenced by a `PduTriggering` that in turn is referenced by a `SomeipTpConnection` in the role `tpSdu` or `tpConcurrentProcessingSdu`.⌋

In other words the `TransformationTechnology.hasInternalState` has to be set to true in case that SOME/IP TP is used.

[TPS_SYST_02032] Transformer chains are ordered list of transformers ⌈A transformer chain consists of an ordered list of `TransformationTechnology`s (transformers).⌋

[constr_3121] The length of transformer chains is limited to 255 transformers

Imposition time: CP: `IT_SysDesc`

⌈The maximum number of `DataTransformation.transformerChain` references in the context of one `DataTransformation` shall be limited to 255.⌋

[constr_3122] At most one transformer of each transformer class inside a transformer chain

Imposition time: CP: `IT_SysDesc`

⌈If the value of a `transformerClass` of a `TransformationTechnology` referenced by a `DataTransformation` does not equal `custom`, it shall be different from all other `transformerClass` values of `TransformationTechnology`s referenced by the same `DataTransformation`.⌋

Only for `custom` transformers it is possible to specify more than one transformer of the same class in the same transformer chain. For all other transformer classes, at most one transformer of a transformer class is allowed to exist in the same transformer chain.

[constr_3123] Serializer transformer shall be the first in a chain

Imposition time: CP: `IT_SysDesc`

⌈A serializer transformer (`TransformationTechnology` with attribute `transformerClass` set to `serializer`) shall be the first transformer in a transformer chain.⌋

[TPS_SYST_02033] Order of the `transformerChain` references in the configuration represents the order on the sending side ⌈The order of `DataTransformation.transformerChain` references in the context of one `DataTransformation` represents the transformation order on the sending side.⌋

[TPS_SYST_02034] Order of the transformers on the receiving side is the reverse of the sending side ⌈The order of the transformers on the receiving side of the data shall be the inverse order of the order of the sending side.⌋

[TPS_SYST_02035] `protocol` contains the human readable protocol identifier ⌈The attribute `protocol` of a `TransformationTechnology` contains the protocol name as a String which this transformer implements.⌋

This attribute is used to distinguish transformers in a human readable way.

[TPS_SYST_02036] `version` contains the version of the `protocol` ⌈The attribute `version` of a `TransformationTechnology` contains the version of the protocol as a String implemented by this transformer.⌋

This attribute is used to distinguish transformers.

[TPS_SYST_02037] The attribute `needsOriginalData` configures a transformer's access to the original data ⌈The attribute `needsOriginalData` of a `TransformationTechnology` specifies whether transformer needs access to the original data.⌋

If it is set to true, the transformer will gain access to the original data. If it is set to false, the transformer will not gain access to the original data.

[constr_3124] Applicability of `needsOriginalData`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `needsOriginalData` of a `TransformationTechnology` shall only be used for the non-first transformers in the transformer chain.⌋

This will only influence the signatures of the transformer on the sender or client side, not on the receiver or server side of a communication.

[TPS_SYST_02038] Specification of transformer class ⌈The transformer class to which this transformer belongs to is specified in the attribute `transformerClass` of a `TransformationTechnology`⌋

[TPS_SYST_02039] Specification of transformer specific properties ⌈Further transformer specific properties can be stated inside the `TransformationDescription` in the role `transformationDescription` of a `TransformationTechnology`⌋

Note:
This is an abstract class without any specified content. If AUTOSAR specifies a transformer and this transformer need configuration possibilities, this class can be inherited to hold those as some kind of container.

[TPS_SYST_02040] Specification of transformer buffer handling ⌈The BufferProperties in the role bufferProperties of a TransformationTechnology specify the buffer handling which shall be executed by the RTE for this transformer.⌋

Class	**BufferProperties**			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	*ARObject*			
Aggregated by	TransformationTechnology.bufferProperties			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
headerLength	Integer	0..1	attr	Defines the length of the header (in bits) this transformer will add in front of the data.
inPlace	Boolean	0..1	attr	If set, the transformer uses the input buffer as output buffer.

Table 7.5: BufferProperties

[constr_9279] Existence of BufferProperties.headerLength

Imposition time: CP: IT_SysDesc

⌈For each BufferProperties, the attribute headerLength shall exist.⌋

[constr_9280] Existence of BufferProperties.inPlace

Imposition time: CP: IT_SysDesc

⌈For each BufferProperties, the attribute inPlace shall exist.⌋

[TPS_SYST_02041] In-place buffer handling of transformers ⌈The attribute inPlace of BufferProperties specifies whether the transformation happens in-place.⌋

[constr_3125] Value of attribute inPlace for the first transformer in a chain

Imposition time: CP: IT_SysDesc

⌈The attribute inPlace shall be set to false if the TransformationTechnology of the BufferProperties is referenced as first reference in the ordered list of references transformerChain from a DataTransformation.⌋

[TPS_SYST_02042] Header length to be considered by transformers ⌈The attribute headerLength of BufferProperties specifies the length of the header (in bits) which the transformer adds.⌋

[constr_3364] headerLength shall be a multiple of 8

Imposition time: CP: IT_SysDesc

⌈The header length in bits specified by headerLength shall be a multiple of 8.⌋

[TPS_SYST_02044] Buffer computation of transformer ⌈The buffer in the RTE that is needed:

- for the SOME/IP transformation will be calculated from the `length` of the `ISignal` that is referencing a transformer chain that includes the SOME/IP Transformer.
- for the ComBased transformation will be calculated from the `length` of the `Pdu` that contains the `ISignalGroup` that is referencing a transformer chain that includes the ComBased Transformer.

⌋

More details can be found in the RTE specification [23]. Please note that the buffer computation for custom transformers is not formalized.

The following examples are showing the calculation of the `length` of an `ISignal` that transports a `VariableDataPrototype` of `ImplementationDataType` of category STRUCTURE via a SOME/IP Transformer.

The example struct consists of five members:

Member1: UINT16

Member2: Struct with a UINT16 length field and a one-dimensional variableSize Array with UINT8 elements and arraySize = 8

Member3: UINT32

Member4: UINT64

Member5: Struct with a UINT16 length field and a one-dimensional variableSize Array with UINT8 elements and arraySize = 8

The SOME/IP Transformer takes the InputData and adds additional 8 bytes as header. In case of SOME/IP the signal based `SOMEIPTransformationISignalProps` and the `DataPrototype` based `SOMEIPTransformationProps` need to be considered as well for the calculation of the `ISignal.length` (see chapter 7.3.2.1 for more details). In our example the following `SOMEIPTransformationProps` settings are valid for the variableSize Array:

- `SOMEIPTransformationProps.alignment` = 64

All these settings lead to the `ISignal.length` of 368 bits as shown in figure Figure 7.6. A padding element is inserted after the first variable size array as described in [PRS_SOMEIP_00611]. The second variable size array is the last element in the serialized data stream and therefore no padding element is inserted afterwards. The automatic padding in SOME/IP after variable size data is described in more detail in [18].

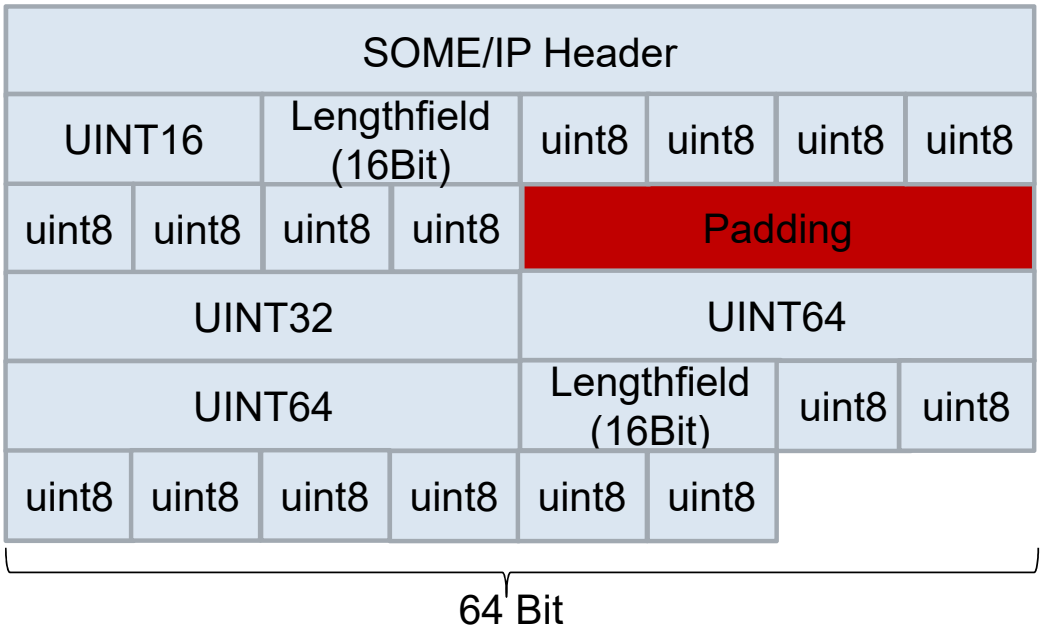

Figure 7.6: Example for calculation of the `ISignal.length`

Please note that the padding in the SOME/IP data stream depends on the actual number of elements that are transmitted in the variable data. Figure Figure 7.7 shows an example where only three elements are transmitted in the first variable size array and therefore the padding is restricted to 1 byte.

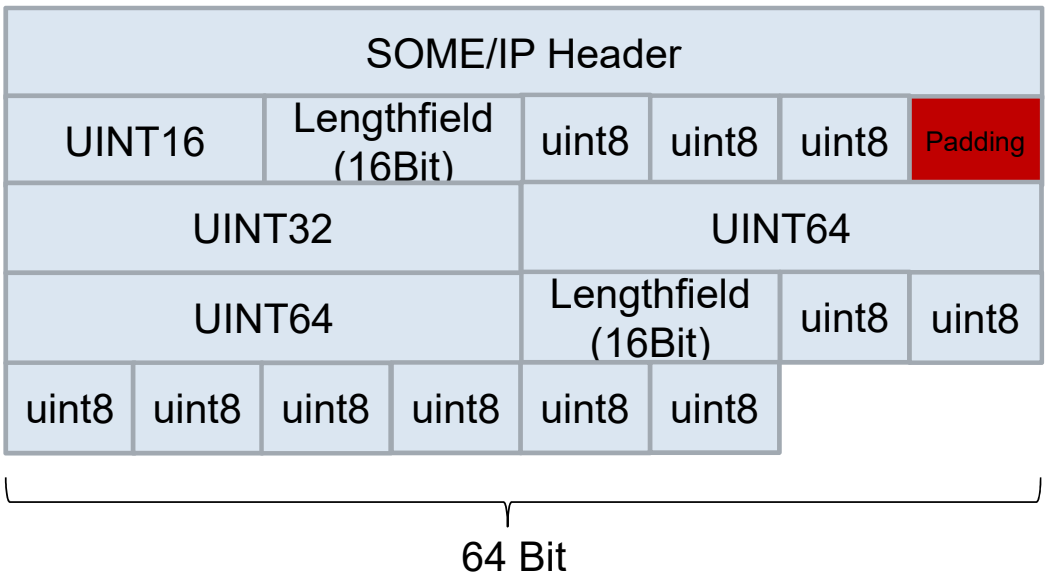

Figure 7.7: SOME/IP Padding Example

Transformer specific configuration can be done in the `TransformationDescription`.

[constr_5231] Allowed values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment

Imposition time: CP: IT_SysDesc

⌈The valid values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment shall be 8, 16, 32, 64, 128 or 256.⌋

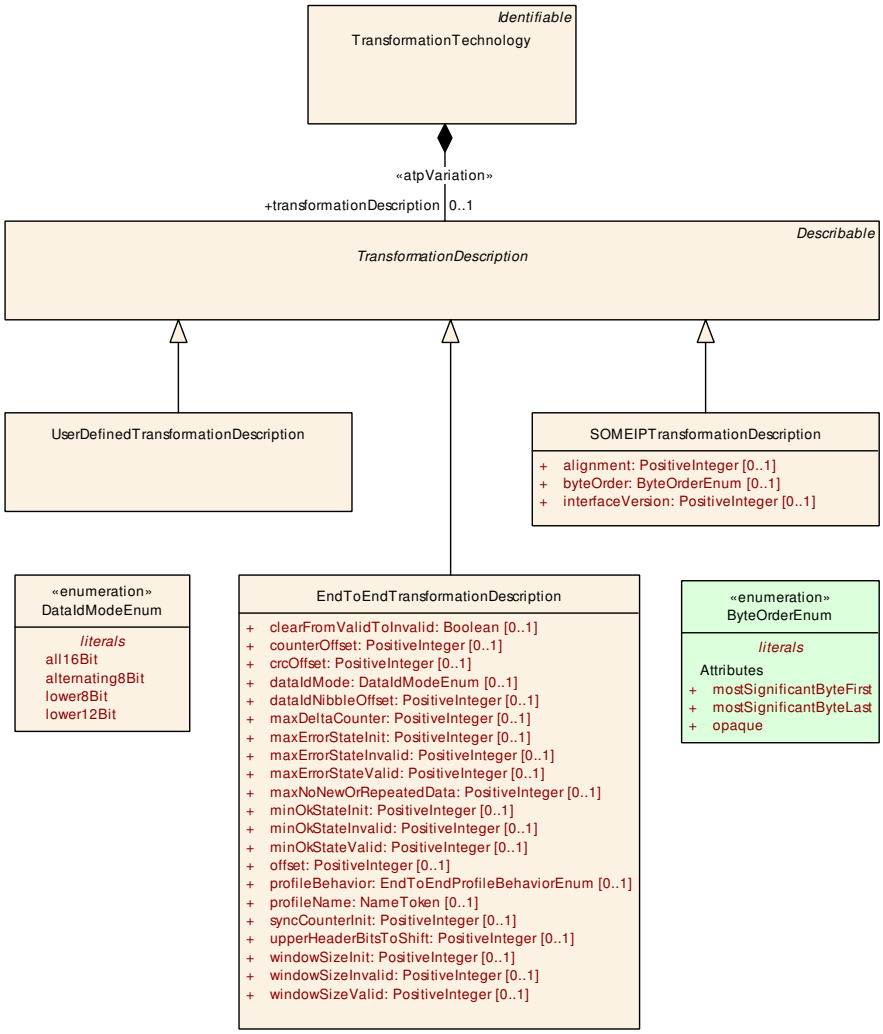

Figure 7.8: Configuration of transformers using TransformationDescription

Class	***TransformationDescription*** (abstract)			
Note	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.			
Base	*ARObject*, *Describable*			
Subclasses	DdsTransformationDescription, EndToEndTransformationDescription, SOMEIPTransformation Description, UserDefinedTransformationDescription			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.6: TransformationDescription

Class	**UserDefinedTransformationDescription**			
Note	The UserDefinedTransformationDescription is used to specify details and documentation for custom transformers.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 7.7: UserDefinedTransformationDescription

[TPS_SYST_02045] SOME/IP Transformer configuration ⌈SOME/IP Transformer shall be configured using `SOMEIPTransformationDescription`.⌋

[TPS_SYST_02046] E2E Transformer configuration ⌈E2E Transformer shall be configured using `EndToEndTransformationDescription`.⌋

For details how to configure those transformers please see chapter 7.3.2 and chapter 7.3.5.

[TPS_SYST_02047] Custom transformer configuration ⌈For custom transformers the specific configuration options shall be placed inside `UserDefinedTransformationDescription`.⌋

To place the custom data in `UserDefinedTransformationDescription` the `AdminData` could be used for example.

The configuration in `TransformationDescription` is valid for the transformer (`TransformationTechnology`) and all associated `ISignal`s. If `ISignal` specific configuration shall be realized which is only valid for the transformation of a specific `ISignal`, the `TransformationISignalProps` shall be used.

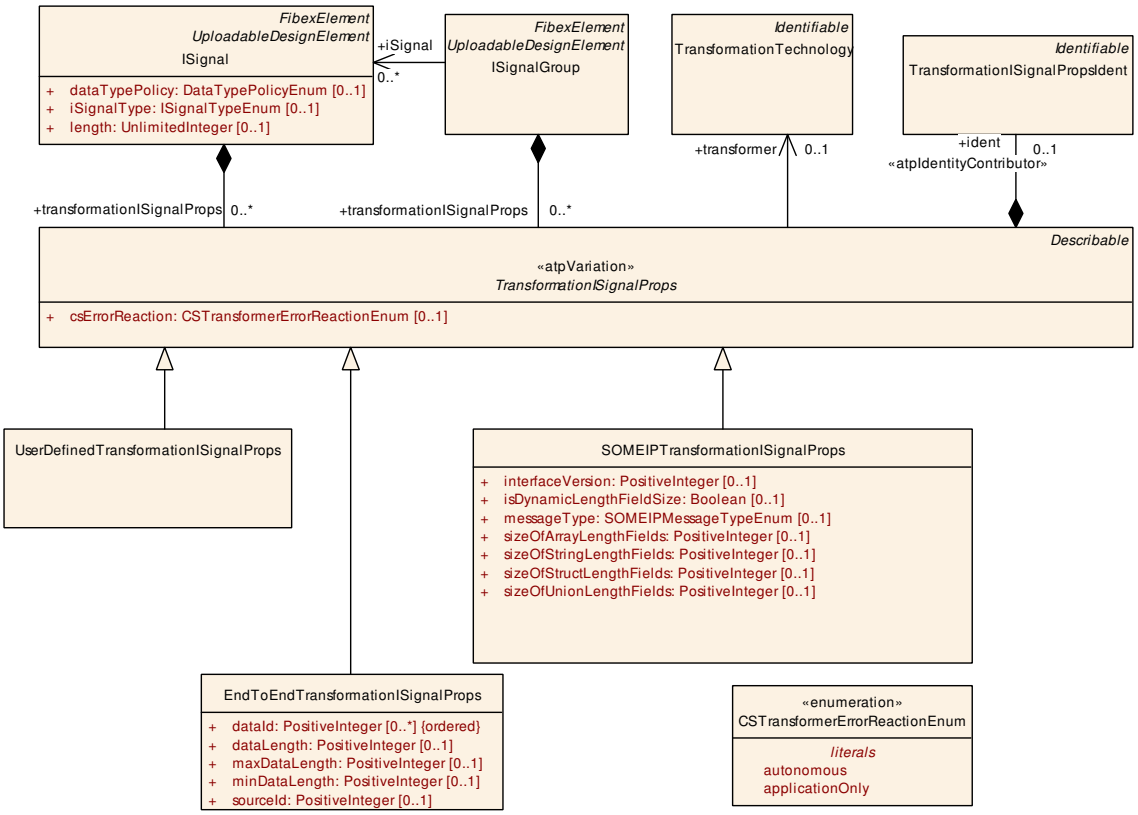

Figure 7.9: Configuration of transformers using `TransformationISignalProps`

Class	«atpVariation» ***TransformationISignalProps*** (abstract)			
Note	TransformationISignalProps holds all the attributes for the different TransformationTechnologies that are ISignal specific. **Tags:** vh.latestBindingTime=postBuild			
Base	*ARObject*, *Describable*			
Subclasses	DdsTransformationISignalProps, EndToEndTransformationISignalProps, SOMEIPTransformationISignalProps, UserDefinedTransformationISignalProps			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
csErrorReaction	CSTransformerError ReactionEnum	0..1	attr	Defines whether the transformer chain of client/server communication coordinates an autonomous error reaction together with the RTE or whether any error reaction is the responsibility of the application.
dataPrototype Transformation Props	DataPrototype TransformationProps	*	aggr	Fine granular modeling of TransfromationProps on the level of DataPrototypes. **Tags:** atp.Splitkey=dataPrototypeTransformationProps, dataPrototypeTransformationProps.ident.shortName
ident	TransformationISignal PropsIdent	0..1	aggr	This adds the ability to add a shortName to TransformationISignalProps. Please note that the short-name needs to be provided if the splitable mechanism is used. **Stereotypes:** atpIdentityContributor

△

Class	«atpVariation» ***TransformationISignalProps*** (abstract)			
transformer	Transformation Technology	0..1	ref	Reference to the TransformationTechnology description that contains transformer specific and ISignal independent configuration properties. **Tags:** atp.Splitkey=transformer

Table 7.8: TransformationISignalProps

Class	**TransformationISignalPropsIdent**			
Note	This meta-class is created to add a shortName to TransformationISignalProps. Please note that the short-name needs to be provided if the splitable mechanism is used and the TransformationISignalProps are distributed over different files.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*TransformationISignalProps*.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.9: TransformationISignalPropsIdent

Enumeration	**CSTransformerErrorReactionEnum**
Note	Possible kinds of error reaction in case of a hard transformer error.
Aggregated by	*TransformationISignalProps*.csErrorReaction
Literal	***Description***
applicationOnly	The application is responsible for any error reaction. No autonomous error reaction of RTE and transformer. **Tags:** atp.EnumerationLiteralIndex=0
autonomous	RTE and Transformer coordinate an autonomous error reaction on their own. **Tags:** atp.EnumerationLiteralIndex=1

Table 7.10: CSTransformerErrorReactionEnum

[constr_9281] Existence of `TransformationISignalProps.dataPdu`

Imposition time: CP: `IT_SysDesc`

⌈For each `TransformationISignalProps`, the reference to `TransformationTechnology` in the role `transformer` shall exist.⌋

[TPS_SYST_02048] `ISignal` specific transformation configuration ⌈If an `ISignal` references a `TransformationTechnology` in the role `dataTransformation` and this transformation shall be configured `ISignal` specific, the `ISignal` shall aggregate a `TransformationISignalProps` element.⌋

[TPS_SYST_02049] Transformer specific `TransformationISignalProps` ⌈The attribute `transformer` of `TransformationISignalProps` shall reference the `TransformationTechnology` in the transformer chain (`DataTransformation`) for which the `ISignal` specific configuration shall be given.⌋

[constr_3213] TransformationISignalProps.csErrorReaction setting in case that the serializer transformerClass and Client/Server communication is used

Imposition time: CP: IT_SysDesc

⌈In TransformationISignalProps the attribute csErrorReaction shall be set if the TransformationISignalProps specifies the details for a TransformationTechnology with transformerClass equal to serializer and the ISignal that aggregates the TransformationISignalProps transports a client/server communication.⌋

[constr_3214] TransformationISignalProps.csErrorReaction setting in case that a transformerClass different from serializer is used or the Client/Server communication is not used

Imposition time: CP: IT_SysDesc

⌈In TransformationISignalProps the attribute csErrorReaction shall not be used if the TransformationISignalProps specifies the details for a TransformationTechnology with transformerClass not equal to serializer or the ISignal that aggregates the TransformationISignalProps does not transport a client/server communication.⌋

[TPS_SYST_02074] Precedence of transformer configuration settings ⌈The same transformer configuration settings may exist in the TransformationDescription, TransformationISignalProps and TransformationComSpecProps elements. The following precedence is valid for such settings:

- TransformationDescription: configuration valid for several ISignals (in case the SOME/IP Transformer or Custom Transformer is used) or ISignalGroups (in case the ComBasedTransformer is used).
- TransformationISignalProps: defines the configuration options valid for a specific referenced ISignal or ISignalGroup. This settings override possible settings in the TransformationDescription.
- TransformationComSpecProps: defines the configuration settings valid for the port to which the ReceiverComSpec belongs (for more details see [4]). This settings override possible settings in the TransformationDescription and TransformationISignalProps.

⌋

[TPS_SYST_02405] TransformationISignalPropsIdent.shortName usage ⌈The TransformationISignalPropsIdent.shortName shall be provided if the splitable mechanism is used and the TransformationISignalProps are distributed over different files.⌋

[TPS_SYST_02075] Mandatory attributes in transformer configuration elements ⌈If a transformer configuration attribute is mandatory due to a particular constraint it means that it shall be defined in at least one of the three possible

locations: TransformationDescription, TransformationISignalProps or TransformationComSpecProps.⌋

Please note that it is not required to define the complete attribute set on each of those locations. It means that it is allowed to overwrite single attributes in elements according to the precedence defined in [TPS_SYST_02074].

[TPS_SYST_02050] ISignal specific configuration of the SOME/IP Transformer ⌈The ISignal specific configuration of the SOME/IP Transformer shall be configured using SOMEIPTransformationISignalProps.⌋

[TPS_SYST_02051] ISignal specific configuration of the E2E Transformer ⌈The ISignal specific configuration of the E2E Transformer shall be configured using End-ToEndTransformationISignalProps.⌋

For details how to configure those transformers ISignal specific please see chapter 7.3.2 and chapter 7.3.5.

[TPS_SYST_02052] ISignal specific configuration of custom transformers ⌈The ISignal specific configuration of custom transformers shall be configured using UserDefinedTransformationISignalProps.⌋

To place the custom data in UserDefinedTransformationDescription the AdminData could be used for example.

To configure which communication shall be subject to transformation is done via references from ISignals and ISignalGroups to DataTransformations. An overview is shown in figure 7.10.

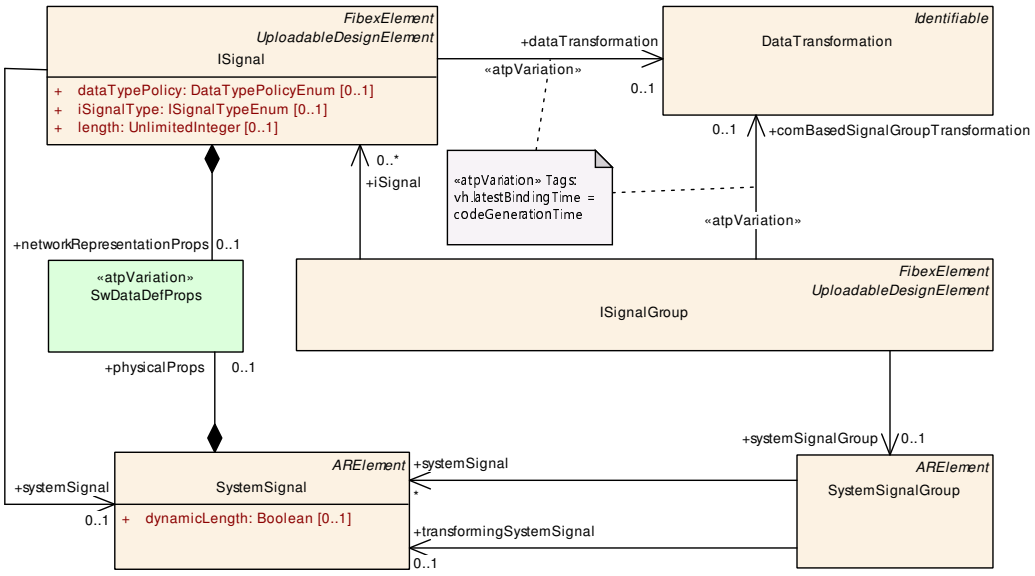

Figure 7.10: Configuration which communication shall be transformed

The DataTransformation element (which represents a transformer chain) is

- either referenced by the ISignal in the role dataTransformation which holds the transformed representation of the data
- referenced by the ISignalGroup in the role comBasedSignalGroupTransformation which holds the custom mapping of the data to the transformed representation or
- referenced by a DataPrototypeMapping in the role firstToSecondDataTransformation,

as defined in [constr_1400] in [4].

A VariableDataPrototype can either become a part of a DataPrototypeMapping based data transformation or of an ISignal-based data transformation as defined in [constr_1401] in [4].

[constr_1387] Transmission of Variable-Size Array Data Types by means of a Transformer

Imposition time: CP: IT_SysDesc

⌈If a Transformer is used for the transmission of a Variable-Size Array Data Types then the Variable-Size Array Data Type shall be a "new-world" variable-size array data type according to [TPS_SWCT_01644] and [TPS_SWCT_01645]. "Old-world" dynamic-size array data types according to [TPS_SWCT_01641] and [TPS_SWCT_01642] are not supported.⌋

7.3.1 Generic Transformer

[TPS_SYST_02053] A reference from ISignal to DataTransformation in the role dataTransformation enables data transformation ⌈To enable the transformation of data, the ISignal which shall hold the transformed data shall reference a DataTransformation in the role dataTransformation.⌋

[TPS_SYST_02054] Definition of data which shall be transformed ⌈If

1. an ISignal references a DataTransformation and
2. this ISignal references a SystemSignal and
3. the referenced SystemSignal is referenced by a SenderReceiverToSignalMapping in the role systemSignal or referenced by a ClientServerToSignalMapping in the role returnSignal and in the role callSignal

then the VariableDataPrototype referenced by the SenderReceiverToSignalMapping or the ClientServerOperation referenced by the ClientServerToSignalMapping shall be transformed.⌋

Using this configuration the result of the transformation will be put into the ISignal even if the data type is a composite type.

Furthermore, another SystemSignal can be added to a SystemSignalGroup in the role transformingSystemSignal to support the configuration where a complex data element is transferred via Sender/Receiver communication both using transformation and traditional mapping of RTE and COM.
The ISignal which references the SystemSignal which is referenced by a SystemSignalGroup in the role transformingSystemSignal shall reference a DataTransformation to transport the transformed data.
In parallel, the traditional mapping of RTE and COM maps all other SystemSignals of the SystemSignalGroup which are referenced in the role systemSignal.

[constr_3127] Certain ISignals always need a reference to DataTransformation

Imposition time: CP: IT_SysDesc

⌈An ISignal which references a SystemSignal which is referenced by a SystemSignalGroup in the role transformingSystemSignal shall always reference a DataTransformation.⌋

7.3.2 SOME/IP Transformer

The specific configuration for SOME/IP transformers takes place in SOMEIPTransformationDescription and SOMEIPTransformationISignalProps shown in Figure 7.11.

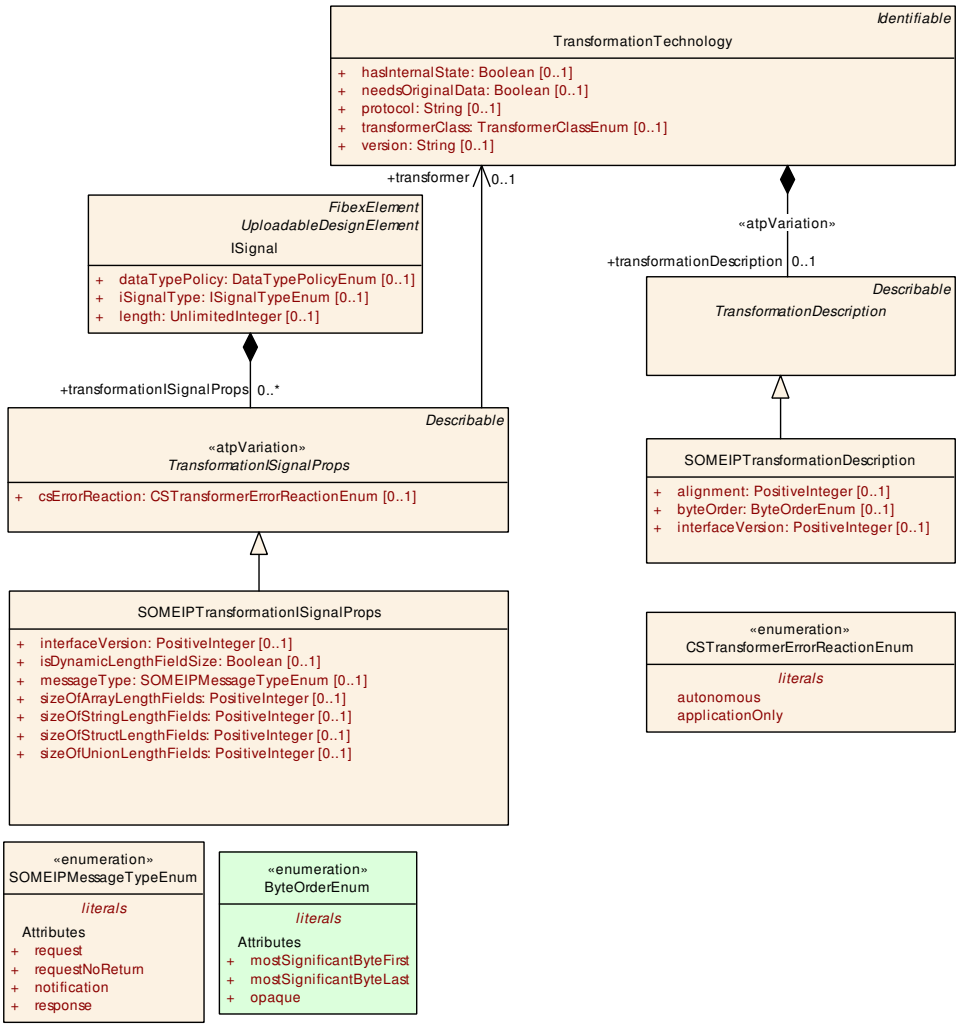

Figure 7.11: SOME/IP specific configuration

Class	**SOMEIPTransformationDescription**			
Note	The SOMEIPTransformationDescription is used to specify SOME/IP transformer specific attributes.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
alignment	PositiveInteger	0..1	attr	Defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element. The alignment shall be specified in Bits.
byteOrder	ByteOrderEnum	0..1	attr	Defines which byte order shall be serialized by the SOME/IP transformer
interfaceVersion	PositiveInteger	0..1	attr	The interface version the SOME/IP transformer shall use.

Table 7.11: SOMEIPTransformationDescription

[constr_9282] Existence of SOMEIPTransformationDescription.alignment

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute alignment shall exist.⌋

[constr_9283] Existence of SOMEIPTransformationDescription.byteOrder

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute byteOrder shall exist.⌋

[constr_9284] Existence of SOMEIPTransformationDescription.interfaceVersion

Imposition time: CP: IT_SysDesc

⌈For each SOMEIPTransformationDescription, the attribute interfaceVersion shall exist.⌋

Class	«atpVariation» **SOMEIPTransformationISignalProps**			
Note	The class SOMEIPTransformationISignalProps specifies ISignal specific configuration properties for SOME/IP transformer attributes.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
interfaceVersion	PositiveInteger	0..1	attr	The interface version the SOME/IP transformer shall use.
isDynamic LengthFieldSize	Boolean	0..1	attr	This attribute shall be used to determine the wire type in the context of using the TLV encoding.
messageType	SOMEIPMessageTypeEnum	0..1	attr	The Message Type which shall be placed into the SOME/IP header.
sizeOfArray LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of fixed-size arrays or dynamic size arrays in the SOME/IP message. This attribute is valid for all available occurrences of fixed-size arrays or dynamic size arrays in the SOME/IP message.
sizeOfString LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of dynamic or fixed length strings in the SOME/IP message. This attribute is valid for all available occurrences of strings in the SOME/IP message.
sizeOfStruct LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of structs in the SOME/IP message. This attribute is valid for all available occurrences of structures in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
sizeOfUnion LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of unions in the SOME/IP message. This attribute is valid for all available occurrences of Unions in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
tlvDataId Definition	TlvDataIdDefinitionSet	*	ref	This reference identifies the TlvDataIdDefinitions relevant for the enclosing SOMEIPTransformationISignalProps **Tags:** atp.Splitkey=tlvDataIdDefinition

Table 7.12: SOMEIPTransformationISignalProps

Enumeration	**ByteOrderEnum**
Note	When more than one byte is stored in the memory the order of those bytes may differ depending on the architecture of the processing unit. If the least significant byte is stored at the lowest address, this architecture is called little endian and otherwise it is called big endian. ByteOrder is very important in case of communication between different PUs or ECUs.
Aggregated by	ApSomeipTransformationProps.byteOrder, BaseTypeDirectDefinition.byteOrder, DiagnosticCommon Props.defaultEndianness, ISignalToIPduMapping.packingByteOrder, MultiplexedIPdu.selectorField ByteOrder, PduToFrameMapping.packingByteOrder, SegmentPosition.segmentByteOrder, SOMEIPTransformationDescription.byteOrder, System.containerIPduHeaderByteOrder
Literal	***Description***
mostSignificantByte First	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format) **Tags:** atp.EnumerationLiteralIndex=0
mostSignificantByte Last	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format) **Tags:** atp.EnumerationLiteralIndex=1
opaque	For opaque data endianness conversion has to be configured to Opaque. See AUTOSAR COM Specification for more details. **Tags:** atp.EnumerationLiteralIndex=2

Table 7.13: ByteOrderEnum

Enumeration	**SOMEIPMessageTypeEnum**
Note	Depending on the style of the communication different message types shall be set in the header of a SOME/IP message.
Aggregated by	SOMEIPTransformationISignalProps.messageType
Literal	***Description***
notification	A request of a notification expecting no response. **Tags:** atp.EnumerationLiteralIndex=1
request	A request expecting a response. **Tags:** atp.EnumerationLiteralIndex=2
requestNoReturn	A fire&forget request. **Tags:** atp.EnumerationLiteralIndex=3
response	The response message. **Tags:** atp.EnumerationLiteralIndex=4

Table 7.14: SOMEIPMessageTypeEnum

[constr_3128] SOME/IP transformer configuration

Imposition time: CP: `IT_SysDesc`

⌈For each `TransformationDescription` variant that is a `SOMEIPTransformationDescription`

- attribute `protocol` of `TransformationTechnology` shall be set to `SOMEIP`
- attribute `version` of `TransformationTechnology` shall be set to `1.0.0`
- attribute `transformerClass` of `TransformationTechnology` shall be set to `serializer`
- attribute `headerLength` of `BufferProperties` shall be set to `64` (bits).

⌋

The SOMEIPTransformationDescription contains the configuration for the transformer which shall be applied to all transformations. ISignal specific transformer configuration (which "override" the general ones) shall be done in SOMEIPTransformationISignalProps.

[TPS_SYST_02055] Alignment of SOME/IP ⌈The attribute alignment defines the alignment used in the SOME/IP transformer in Bits.⌋

[TPS_SYST_02056] Byte Order of SOME/IP ⌈The attribute byteOrder defines the byte order used in the SOME/IP transformer for creating the on wire format.⌋

[constr_3129] Byte Order of SOME/IP transformer

Imposition time: CP: IT_SysDesc

⌈The attribute byteOrder of SOMEIPTransformationDescription shall be different from opaque.⌋

[TPS_SYST_02057] Interface Version of SOME/IP ⌈The attribute interfaceVersion of SOMEIPTransformationDescription as well as interfaceVersion of SOMEIPTransformationISignalProps defines the interface version used by the SOME/IP transformer.⌋

[constr_3130] Range of Interface Version

Imposition time: CP: IT_SysDesc

⌈The value of the attribute interfaceVersion shall be in the range $[0; 255]$⌋

[TPS_SYST_02092] Size of Array Length Fields ⌈The attribute sizeOfArrayLengthFields of SOMEIPTransformationISignalProps defines the size of an length field generated by the SOME/IP transformer in front of all available fixed-size arrays or dynamic size arrays in the ISignal. See also [constr_3282].⌋

[constr_5244] Value of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields

Imposition time: CP: IT_SysDesc

⌈If attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields is configured, then the value of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all variable-size arrays that are transported in the SOME/IP message.

In other words, for each variable-size array contained in the SOME/IP message, the numerical value of *maximum number of elements * sizeof(data type of array element)* shall be computed which yields the maximum number of bytes required to store the individual variable-size array.

The size of the attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields shall be set such that the highest value (or bigger) obtained from the individual computations for the contained variable-size arrays can fit into the length

field. The unit of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields is bytes.⌋

[TPS_SYST_02093] Size of Structure Length Fields ⌈The attribute sizeOfStructLengthFields of SOMEIPTransformationISignalProps defines the size of an length field generated by the SOME/IP transformer in front of all available structures in the ISignal. See also [constr_3283].⌋

[TPS_SYST_02359] Size of String Length Fields ⌈The attribute sizeOfStringLengthFields of SOMEIPTransformationISignalProps defines the size of an length field generated by the SOME/IP transformer in front of all available strings in the ISignal. See also [constr_5246].⌋

[constr_5245] Value of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields

Imposition time: CP: IT_SysDesc

⌈If attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields is configured, then the value of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all strings that are transported in the SOME/IP message.

In other words, for each string contained in the SOME/IP message, the numerical value of *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding)* shall be computed which yields the maximum number of bytes required to store the individual string.

The size of the attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields shall be set such that the highest value (or bigger) obtained from the individual computations for the contained strings can fit into the length field. The unit of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields is bytes.⌋

[constr_1441] In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer

Imposition time: CP: IT_SysDesc

⌈If an ImplementationDataType according to [TPS_SWCT_01700], i.e. of category STRUCT that encloses an ImplementationDataTypeElement of category UNION, is used to directly or (via a DataTypeMap) indirectly type an AutosarDataPrototype and the latter is mapped to a SystemSignal then the ISignal that references that SystemSignal shall aggregate SOMEIPTransformationISignalProps in the role transformationISignalProps.⌋

[TPS_SYST_02094] Size of Union Length Fields ⌈The attribute sizeOfUnionLengthFields of SOMEIPTransformationISignalProps defines the size of an

length field generated by the SOME/IP transformer in front of all available unions in the ISignal. See also [constr_3284].⌋

In principle there is no need to define a size of the length indicator because the size can be computed from the data structure itself. However there is a use case to extend on the sender side while keeping the receiver side as it is. This means that there is the need to express the size of the length indicator because the extended data structure may reach a length that exceeds the capacity of the original computed size indicator.

[constr_3218] Range of Size of Array Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfArrayLengthFields of SOMEIPTransformationISignalProps shall be either 0, 1, 2 or 4.⌋

[constr_3220] Range of Size of Structure Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfStructLengthFields of SOMEIPTransformationISignalProps shall be either 0, 1, 2 or 4.⌋

[constr_3221] Range of Size of Union Length Fields

Imposition time: CP: IT_SysDesc

⌈The value of attribute sizeOfUnionLengthFields of SOMEIPTransformationISignalProps shall be either 0, 1, 2 or 4.⌋

[TPS_SYST_02080] Message type of SOME/IP ⌈The attribute messageType of SOMEIPTransformationISignalProps defines the message type used by the SOME/IP transformer for the serialized ISignal.⌋

7.3.2.1 SOME/IP Transformation Properties on the level of DataPrototypes

The serialization of SOME/IP is based on the interface specification. For certain datatypes like structures, unions and arrays SOME/IP supports the configuration of length fields that will be put in front of the serialized data. AUTOSAR supports the configuration of such SOME/IP settings on two different levels:

- modeling on ISignal level that is valid for all available occurrences of a datatype in the SOME/IP message (see [TPS_SYST_02092], [TPS_SYST_02093] and [TPS_SYST_02094])
- fine granular modeling on the level of DataPrototypes (see [TPS_SYST_02121])

To allow such a fine granular modeling SOMEIPTransformationProps are defined and collected in TransformationPropsSets.

Class	TransformationPropsSet			
Note	Collection of TransformationProps. **Tags:** atp.recommendedPackage=TransformationPropsSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
transformation Props	TransformationProps	*	aggr	Transformer specific configuration properties. **Tags:** atp.Splitkey=transformationProps, transformation Props.shortName

Table 7.15: TransformationPropsSet

Class	***TransformationProps*** (abstract)			
Note	This meta-class represents a abstract base class for transformation settings.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	SOMEIPTransformationProps, UserDefinedTransformationProps			
Aggregated by	TransformationPropsSet.transformationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table 7.16: TransformationProps

Class	SOMEIPTransformationProps			
Note	The class SOMEIPTransformationProps specifies SOME/IP specific configuration properties.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TransformationProps*			
Aggregated by	TransformationPropsSet.transformationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
alignment	PositiveInteger	0..1	attr	Defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element. The alignment shall be specified in Bits.
sizeOfArray LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced Array in the SOME/IP message.
sizeOfString LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced String in the SOME/IP message.
sizeOfStruct LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Structure in the SOME/IP message.
sizeOfUnion LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Union in the SOME/IP message.

Table 7.17: SOMEIPTransformationProps

The relation between `SOMEIPTransformationProps` and a `DataPrototype` is created with `DataPrototypeTransformationProps` in the context of an `ISignal`.

[TPS_SYST_02127] Usage of `DataPrototypeTransformationProps` in case of a `VariableDataPrototype` ⌈If a `VariableDataPrototype` is transported

in the ISignal the DataPrototypeTransformationProps can be used to assign SOMEIPTransformationProps to a DataPrototype that is or is part of the VariableDataPrototype.⌋

[TPS_SYST_02128] Usage of DataPrototypeTransformationProps in case of a ClientServerOperation ⌈If a ClientServerOperation is transported in the ISignal (callSignal or returnSignal) the DataPrototypeTransformationProps can be used to assign SOMEIPTransformationProps to a DataPrototype that is or is part of an ArgumentDataPrototype of the ClientServerOperation.⌋

[TPS_SYST_02129] Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType ⌈To assign the SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface that is typed by an ApplicationDataType the DataPrototypeInSenderReceiverInterfaceInstanceRef shall reference the DataPrototype with the targetDataPrototypeInSr reference. The rootDataPrototypeInSr and contextDataPrototypeInSr references shall not be used.⌋

[TPS_SYST_02212] Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a ClientServerInterface typed by an ApplicationDataType ⌈To assign the SOMEIPTransformationProps to a root AutosarDataPrototype in a ClientServerInterface that is typed by an ApplicationDataType the DataPrototypeInClientServerInterfaceInstanceRef shall reference the DataPrototype with the targetDataPrototypeInCs reference. The rootDataPrototypeInCs and contextDataPrototypeInCs references shall not be used.⌋

[TPS_SYST_02130] Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType ⌈To assign the SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a SenderReceiverInterface that is typed by an ApplicationDataType the DataPrototypeInSenderReceiverInterfaceInstanceRef shall reference the subElement with the targetDataPrototypeInSr reference. In addition the rootDataPrototypeInSr shall be set to define the context. Optionally it may be necessary to use contextDataPrototypeInSr references because the target subElement may be arbitrarily nested within the root AutosarDataPrototype.⌋

[TPS_SYST_02213] Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a ClientServerInterface typed by an ApplicationDataType ⌈To assign the SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a ClientServerInterface that is typed by an ApplicationDataType the DataPrototypeInClientServerInterfaceInstanceRef shall reference the subElement with the targetDataPrototypeInCs reference. In addition the rootDataPrototypeInCs shall be

set to define the context. Optionally it may be necessary to use contextDataPrototypeInCs references because the target subElement may be arbitrarily nested within the root AutosarDataPrototype.⌋

[TPS_SYST_02131] Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ImplementationDataType ⌈To assign the SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface that is typed by an ImplementationDataType the DataPrototypeInSenderReceiverInterfaceInstanceRef shall reference the AutosarDataPrototype with the targetDataPrototypeInSr reference. The rootDataPrototypeInSr and contextDataPrototypeInSr references in the DataPrototypeInSenderReceiverInterfaceInstanceRef shall not be used.⌋

[TPS_SYST_02214] Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a ClientServerInterface typed by an ImplementationDataType ⌈To assign the SOMEIPTransformationProps to a root AutosarDataPrototype in a ClientServerInterface that is typed by an ImplementationDataType the DataPrototypeInClientServerInterfaceInstanceRef shall reference the AutosarDataPrototype with the targetDataPrototypeInCs reference. The rootDataPrototypeInCs and contextDataPrototypeInCs references in the DataPrototypeInClientServerInterfaceInstanceRef shall not be used.⌋

[TPS_SYST_02132] Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ImplementationDataType ⌈To assign the SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a ClientServerInterface or SenderReceiverInterface that is typed by an ImplementationDataType the ImplementationDataTypeElementInPortInterfaceRef shall reference the targetImplementationDataTypeElement. In addition the rootDataPrototype shall be set to define the context. Optionally it may be necessary to use contextImplementationDataElement references because the target subElement may be arbitrarily nested within the root AutosarDataPrototype.⌋

[TPS_SYST_02195] Applicable use cases for DataPrototypeReference ⌈

Use case	Role
AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType	DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr
AutosarDataPrototype in a ClientServerInterface typed by an ApplicationDataType	DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs

▽

△

Use case	Role
DataPrototype in AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationCompositeDataType	DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr
DataPrototype in AutosarDataPrototype in a ClientServerInterface typed by an ApplicationCompositeDataType	DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs
AutosarDataPrototype in a SenderReceiverInterface typed by an ImplementationDataType	DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr
AutosarDataPrototype in a ClientServerInterface typed by an ImplementationDataType	DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs
DataPrototype in AutosarDataPrototype typed by an ImplementationDataType	ImplementationDataTypeElementInPortInterfaceRef.targetImplementationDataTypeElement

Possible use cases for the usage of DataPrototypeReference

⌋

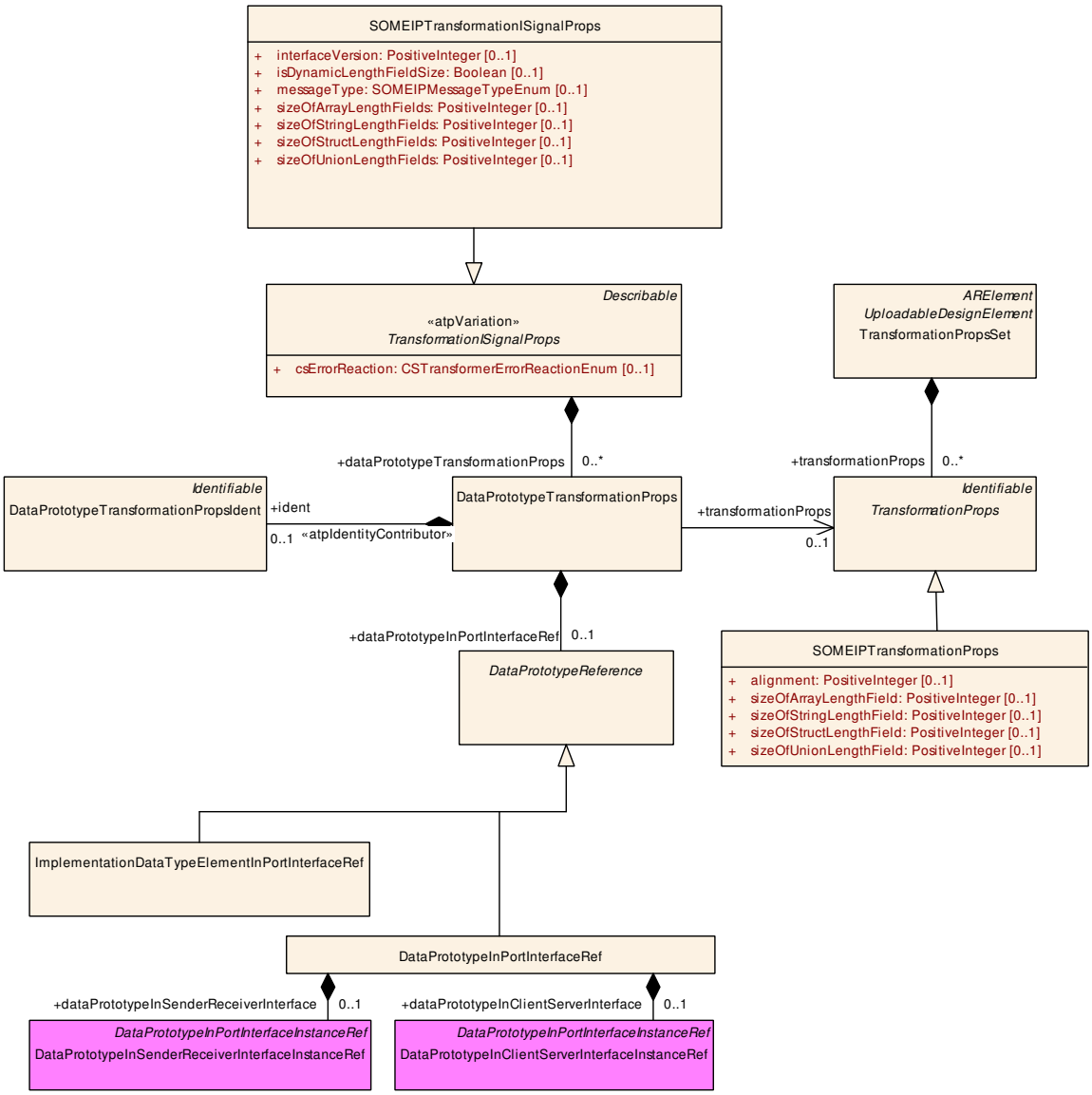

Figure 7.12: Transformation Properties on the level of DataPrototypes

Class	**DataPrototypeTransformationProps**			
Note	DataPrototypeTransformationProps allows to set the attributes for the different Transformation Technologies that are DataPrototype specific.			
Base	*ARObject*			
Aggregated by	*TransformationISignalProps*.dataPrototypeTransformationProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataPrototypeInPortInterfaceRef	DataPrototypeReference	0..1	aggr	Reference to a DataPrototype that is transported in the serialized ISignal. **Tags:** atp.Splitkey=dataPrototypeInPortInterfaceRef
ident	DataPrototypeTransformationPropsIdent	0..1	aggr	This adds the ability to add a shortName to DataPrototype TransformationProps. Please note that the short-name needs to be provided if the splitable mechanism is used. **Stereotypes:** atpIdentityContributor

△

Class	**DataPrototypeTransformationProps**			
network Representation Props	SwDataDefProps	0..1	aggr	Specification of the actual network representation for the referenced primitive DataPrototype. If a network representation is provided then the baseType shall be used by the Transformer as input for the serialization/ deserilaization. **Tags:** atp.Splitkey=networkRepresentationProps
transformation Props	TransformationProps	0..1	ref	Collection of AutosarDataPrototype related configuration settings for a transformer. **Tags:** atp.Splitkey=transformationProps

Table 7.18: DataPrototypeTransformationProps

Class	**DataPrototypeTransformationPropsIdent**			
Note	This meta-class is created to add a shortName to DataPrototypeTransformationProps. Please note that the short-name needs to be provided if the splitable mechanism is used and the DataPrototype TransformationProps are distributed over different files.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataPrototypeTransformationProps.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.19: DataPrototypeTransformationPropsIdent

Class	***DataPrototypeReference*** (abstract)			
Note	This meta-class provides the ability to reference a DataPrototype.			
Base	*ARObject*			
Subclasses	DataPrototypeInPortInterfaceRef, ImplementationDataTypeElementInPortInterfaceRef			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.20: DataPrototypeReference

Class	**DataPrototypeInPortInterfaceRef**			
Note	This class represents a RootDataPrototype that is typed by an ApplicationDataType or Implementation DataType or a DataTypeElement that is aggregated within a composite application data type (record or array).			
Base	*ARObject*, *DataPrototypeReference*			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**DataPrototypeInPortInterfaceRef**			
dataPrototypeIn ClientServer Interface	DataPrototype	0..1	iref	This element defines a reference to a DataPrototype in the context of a ClientServerInterface. **Tags:** atp.Splitkey=dataPrototypeInClientServerInterface, dataPrototypeInClientServerInterface.rootDataPrototype InCs, dataPrototypeInClientServerInterface.contextData PrototypeInCs, dataPrototypeInClientServer Interface.targetDataPrototypeInCs **InstanceRef implemented by:** DataPrototypeInClient ServerInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.
dataPrototypeIn SenderReceiver Interface	DataPrototype	0..1	iref	This element defines a reference to a DataPrototype in the context of a SenderReceiverInterface. **Tags:** atp.Splitkey=dataPrototypeInSenderReceiver Interface, dataPrototypeInSenderReceiverInterface.root DataPrototypeInSr, dataPrototypeInSenderReceiver Interface.contextDataPrototypeInSr, dataPrototypeIn SenderReceiverInterface.targetDataPrototypeInSr **InstanceRef implemented by:** DataPrototypeInSender ReceiverInterfaceInstanceRef This Attribute is only used by the AUTOSAR Classic Platform.

Table 7.21: DataPrototypeInPortInterfaceRef

Class	**DataPrototypeInSenderReceiverInterfaceInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*, *DataPrototypeInPortInterfaceInstanceRef*			
Aggregated by	DataPrototypeInPortInterfaceRef.dataPrototypeInSenderReceiverInterface			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	SenderReceiver Interface	0..1	ref	**Stereotypes:** atpDerived
contextData PrototypeInSr (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=20
rootData PrototypeInSr	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=10
targetData PrototypeInSr	DataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=30

Table 7.22: DataPrototypeInSenderReceiverInterfaceInstanceRef

[constr_9285] Existence of DataPrototypeInSenderReceiverInterface-InstanceRef.targetDataPrototypeInSr

Imposition time: CP: IT_SysDesc

⌈For each DataPrototypeInSenderReceiverInterfaceInstanceRef, the reference to DataPrototype in the role targetDataPrototypeInSr shall exist.⌋

Class	**DataPrototypeInClientServerInterfaceInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*, *DataPrototypeInPortInterfaceInstanceRef*			
Aggregated by	DataPrototypeInPortInterfaceRef.dataPrototypeInClientServerInterface, DiagnosticServiceSw Mapping.accessedDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	ClientServerInterface	0..1	ref	**Stereotypes:** atpDerived
contextData PrototypeInCs (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=20
rootData PrototypeInCs	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=10
targetData PrototypeInCs	DataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=30

Table 7.23: DataPrototypeInClientServerInterfaceInstanceRef

[constr_9286] Existence of DataPrototypeInClientServerInterface-InstanceRef.targetDataPrototypeInCs

Imposition time: CP: IT_SysDesc

⌈For each DataPrototypeInClientServerInterfaceInstanceRef, the reference to DataPrototype in the role targetDataPrototypeInCs shall exist.⌋

Class	**ImplementationDataTypeElementInPortInterfaceRef**			
Note	This meta-class represents the ability to refer to the internal structure of an AutosarDataPrototype which is typed by an ImplementationDatatype in the context of a PortInterface. In other words, this meta-class shall not be used to model a reference to the AutosarDataPrototype as a target itself, even if the AutosarDataPrototype is typed by an ImplementationDataType and even if that ImplementationDataType represents a composite data type.			
Base	*ARObject*, *DataPrototypeReference*			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element, TransmissionComSpecProps.onChangeDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context Implementation DataElement (ordered)	AbstractImplementation DataTypeElement	*	ref	This is a context in case there are subelements with explicit types. The reference has to be ordered to properly reflect the nested structure. **Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	This refers to the AutosarDataPrototype which is typed by the ImplementationDatatype. The targetDataPrototype and all defined contextDataPrototypes can be found within this rootDataPrototype. **Tags:** atp.Splitkey=rootDataPrototype xml.sequenceOffset=10
target Implementation DataType Element	AbstractImplementation DataTypeElement	0..1	ref	This is a target ImplementationDataTypeElement in case that the rootDataPrototype is composite and the target is a subElement of the rootDataPrototype. **Tags:** atp.Splitkey=targetImplementationDataTypeElement xml.sequenceOffset=30

Table 7.24: ImplementationDataTypeElementInPortInterfaceRef

[TPS_SYST_02121] Scope of DataPrototypeTransformationProps ⌈DataPrototypeTransformationProps is defined either

- for the root DataPrototype that is transmitted in the serialized ISignal
- for each of the composite subElements of the composite root DataPrototype

⌋

[TPS_SYST_02406] DataPrototypeTransformationPropsIdent.shortName usage ⌈The DataPrototypeTransformationPropsIdent.shortName shall be provided if the splitable mechanism is used and the DataPrototypeTransformationProps are distributed over different files.⌋

[TPS_SYST_02123] Size of a length field for a chosen array ⌈The attribute sizeOfArrayLengthField of SOMEIPTransformationProps defines the size of a length field generated by the SOME/IP transformer in front of the fixed-size or dynamic size array for which the DataPrototypeTransformationProps is defined according to [TPS_SYST_02121].⌋

[constr_5247] Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField

Imposition time: CP: IT_SysDesc

⌈If the configuration of length field is done using DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField then the value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of elements * sizeof(data type of array element)*.⌋

[TPS_SYST_02124] Size of a length field for a chosen structure ⌈The attribute sizeOfStructLengthField of SOMEIPTransformationProps defines the size of a length field generated by the SOME/IP transformer in front of the structure for which the DataPrototypeTransformationProps is defined according to [TPS_SYST_02121].⌋

[TPS_SYST_02125] Size of a length field for a chosen union ⌈The attribute sizeOfUnionLengthField of SOMEIPTransformationProps defines the size of a length field generated by the SOME/IP transformer in front of the union for which the DataPrototypeTransformationProps is defined according to [TPS_SYST_02121].⌋

[TPS_SYST_02360] Size of a length field for a chosen string ⌈The attribute sizeOfStringLengthField of SOMEIPTransformationProps defines the size of a length field generated by the SOME/IP transformer in front of the string for which the DataPrototypeTransformationProps is defined according to [TPS_SYST_02121].⌋

[constr_5248] Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField

Imposition time: CP: IT_SysDesc

⌈If the configuration of length field is done using DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField then the value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding)*.⌋

[TPS_SYST_02126] Alignment of a dynamic DataPrototype ⌈The attribute alignment of SOMEIPTransformationProps defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element for which the DataPrototypeTransformationProps is defined according to [TPS_SYST_02121].⌋

[constr_3278] Usage of SOMEIPTransformationProps.sizeOfArrayLengthField

Imposition time: CP: IT_SysDesc

⌈The attribute sizeOfArrayLengthField of SOMEIPTransformationProps shall only be defined if the DataPrototypeTransformationProps is defined for a static size array according to [TPS_SYST_02121].⌋

[constr_3279] Usage of SOMEIPTransformationProps.sizeOfStructLengthField

Imposition time: CP: IT_SysDesc

⌈The attribute sizeOfStructLengthField of SOMEIPTransformationProps shall only be defined if the DataPrototypeTransformationProps is defined for a structure according to [TPS_SYST_02121].⌋

[constr_3280] Usage of SOMEIPTransformationProps.sizeOfUnionLengthField

Imposition time: CP: IT_SysDesc

⌈The attribute sizeOfUnionLengthField of SOMEIPTransformationProps shall only be defined if the DataPrototypeTransformationProps is defined for a union according to [TPS_SYST_02121].⌋

[constr_3281] Usage of SOMEIPTransformationProps.alignment

Imposition time: CP: IT_SysDesc

⌈The attribute alignment of SOMEIPTransformationProps shall only be defined if the DataPrototypeTransformationProps is defined for a variable data length data element according to [TPS_SYST_02121].⌋

[constr_3282] SOME/IP Transformation settings for arrays in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈In the context of an ISignal the usage of DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField is only allowed if the SOMEIPTransformationISignalProps.sizeOfArrayLengthFields is not defined.⌋

[constr_3283] SOME/IP Transformation settings for structures in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈In the context of an ISignal the usage of DataPrototypeTransformationProps.transformationProps.sizeOfStructLengthField is only allowed if the SOMEIPTransformationISignalProps.sizeOfStructLengthFields is not defined.⌋

[constr_3284] SOME/IP Transformation settings for unions in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈In the context of an ISignal the usage of DataPrototypeTransformationProps.transformationProps.sizeOfUnionLengthField is only allowed if the SOMEIPTransformationISignalProps.sizeOfUnionLengthFields is not defined.⌋

[constr_5246] SOME/IP Transformation settings for strings in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈In the context of an ISignal the usage of DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField is only allowed if the SOMEIPTransformationISignalProps.sizeOfStringLengthFields is not defined.⌋

[constr_3285] Alignment of variable data length data elements in the context of an ISignal

Imposition time: CP: IT_SysDesc

⌈The definition of DataPrototypeTransformationProps.transformationProps.alignment is only allowed if the SOMEIPTransformationDescription.alignment is not defined.⌋

7.3.2.2 Network Representation

In order to assure that the serialization of the transported data on the sender side and its deserialization on the receiver side(s) is done correctly, system designers need to

assure that the same datatypes (i.e., SwBaseTypes) are used for the serialization/deserialization on both sides. However, this agreement does not imply the use or equality of the SwBaseTypes defined by the ImplementationDataType used by the application software on the sender and (possibly multiple) receiver sides. This means that each EcuInstance, regardless if it belongs to a sender or receiver, can use one datatype for the serialization/deserialization (e.g., UInt16 in the actual SOME/IP transformer code) and another datatype in the application software (e.g., Float32 in the actual application software component code).

In order to define the commonly agreed datatypes for the serialization/deserialization of the transported data by the sender and possibly multiple receivers, AUTOSAR defines the following two approaches:

- serialization based on the network representation ([TPS_SYST_02136])
- serialization based on the ImplementationDataTypes ([TPS_SYST_02137])

[TPS_SYST_02136] Serialization based on the network representation ⌈If a network representation that defines a SwBaseType is provided for each DataPrototype typed by a primitive data type that is part of the serialized ISignal (ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps), these SwBaseTypes shall be used for the serialization/deserialization.⌋

[TPS_SYST_02137] Serialization based on the ImplementationDataTypes ⌈For primitive DataPrototypes that are part of the serialized ISignal where no network representation is provided (ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps), SwBaseType shall be provided by the ImplementationDataTypes that either types the corresponding PortPrototypes on the top level Software Composition that represents the communicating EcuInstances, or it is mapped to the ApplicationDataType that types it.⌋

[constr_3317] Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the ImplementationDataTypes

Imposition time: CP: IT_SysDesc

⌈In order to assure the same interpretation of the serialized data by the SOME/IP transformers on the sender and receiver sides in case of serialization based on either a primitive or a composite ImplementationDataType, the same SwBaseType shall be defined

- for this primitive DataPrototype or
- for each primitive DataPrototype of the leaf elements of the composite DataPrototype starting from the first element until and including the last element that is requested by the receiver,

by the ImplementationDataTypes that either types the corresponding PortPrototypes on the top level Software Composition of the communicating EcuInstances, or it is mapped to the ApplicationDataType that types it.⌋

If the serialization is based on the ImplementationDataTypes, the same data has to be transmitted on all buses, i.e., it is not possible to transmit different precision (i.e., number of bits) on different buses, as with the serialization based on the network representation on the ISignal level.

ImplementationDataTypes used by the actual application for the transported data shall be defined by the corresponding PortPrototypes on the AtomicSwComponentTypes of the communicating EcuInstances. The RTE is responsible for the possible type conversion and scaling in case of different ImplementationDataTypes used for the serialization/deserialization and in the application.

[TPS_SYST_02138] Definition of the network representation ⌈The network representation for each DataPrototype typed by a primitive data type in the serialized data shall be defined by the SwDataDefProps that is aggregated by the DataPrototypeTransformationProps in the role networkRepresentationProps.⌋

In other words: If a DataPrototype is transported in the ISignal the DataPrototypeTransformationProps can be used to assign a network representation to each primitive DataPrototype that is part of the enclosing DataPrototype.

[TPS_SYST_02139] Applicability of the SwDataDefProps attributes for the network representation of the serialized data ⌈

Attributes of SwDataDefProps	networkRepresentationProps
additionalNativeTypeQualifier	N/A
annotation	N/A
baseType	D
compuMethod	D
dataConstr	D
displayFormat	D
displayPresentation	N/A
implementationDataType	N/A
invalidValue	N/A
swAddrMethod	N/A
swAlignment	N/A
swBitRepresentation	N/A
swCalibrationAccess	N/A
swCalprmAxisSet	N/A
swComparisonVariable	N/A
swDataDependency	N/A
swHostVariable	N/A

▽

△

swImplPolicy	N/A
swIntendedResolution	N/A
swInterpolationMethod	N/A
swIsVirtual	N/A
swPointerTargetProps	N/A
swRecordLayout	N/A
swRefreshTiming	N/A
swTextProps	N/A
swValueBlockSize	N/A
unit	D
valueAxisDataType	N/A

Usage of DataPrototypeTransformationProps.networkRepresentationProps shall follow the restrictions that are documented in this table.

⌋

The following settings apply in [TPS_SYST_02139]:

D Attribute can be **defined** in the scope of this element.

N/A Attribute is **not applicable** for usage in the scope of this element.

[constr_3318] Allowed use of ISignal.networkRepresentationProps

Imposition time: CP: IT_SysDesc

⌈If a reference from ISignal to DataTransformation in the role dataTransformation exists, this ISignal SHALL NOT aggregate SwDataDefProps in the role networkRepresentationProps.⌋

This means that aggregating SwDataDefProps by an ISignal is applicable only if this ISignal is not transformed by an algorithmic transformer like SOME/IP.

[constr_3319] Existence of DataPrototypeTransformationProps.networkRepresentationProps

Imposition time: CP: IT_SysDesc

⌈ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps shall either

- not exist at all or
- shall be defined for all leaf elements of the root DataPrototype transmitted in the ISignal

⌋

This means that either all leaf elements of the transformed ISignal shall have a network representation, or none.

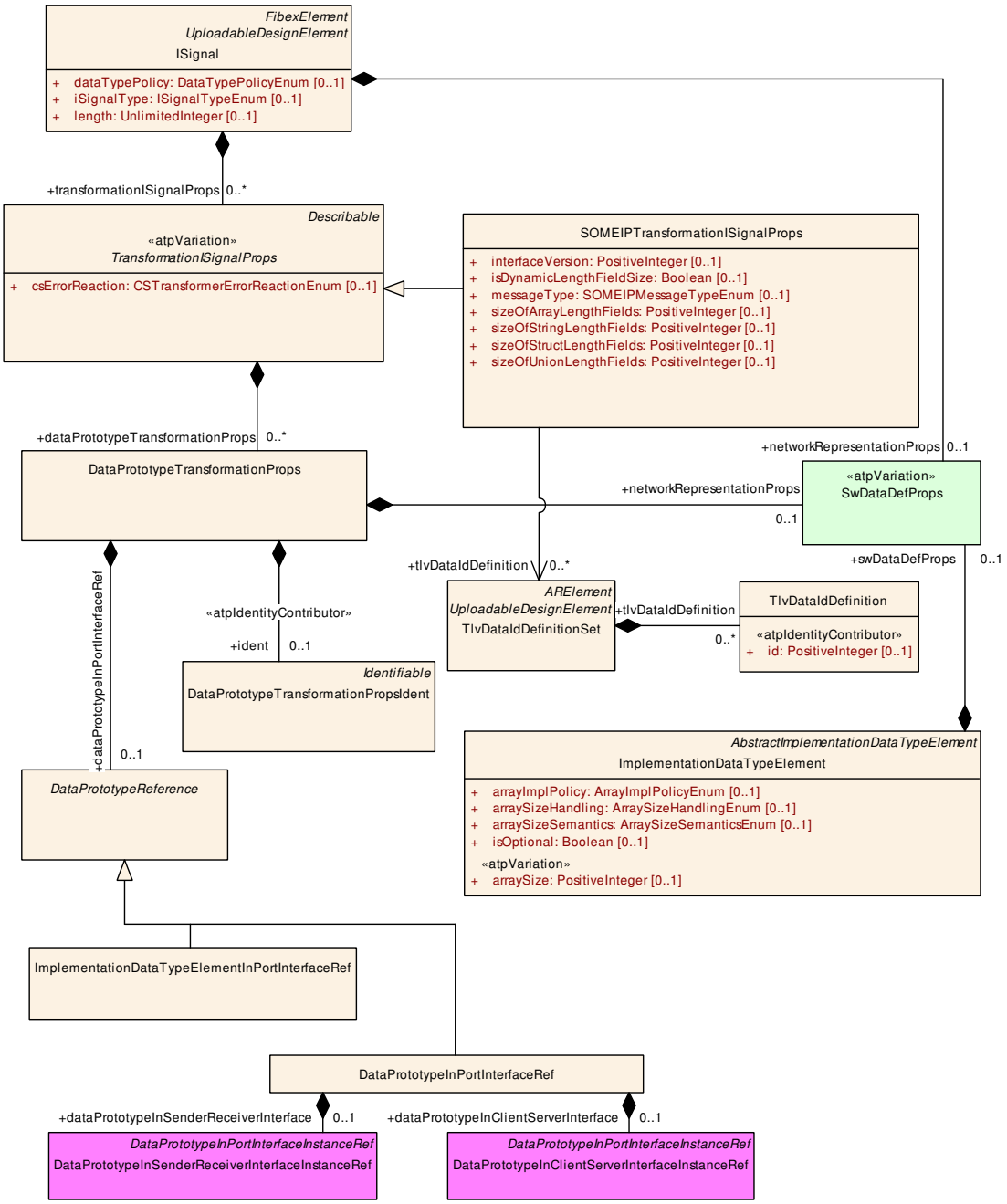

Figure 7.13: Transformer Network Representation

7.3.2.2.1 Example - Serialization based on the network representation

An example with concrete methodological steps in a common OEM-Tier 1 development process for the serialization based on the network representation is presented in Figure 7.14. The steps are as follows:

1. OEM decides on a common `SwBaseType` and `CompuMethod` for each bus, as part of the network representation, and `physicalProps` used for serialization/deserialization of one concrete complex data type.

2. OEM provides an ApplicationDataType (with physical semantics) on the PortPrototype on the RootSwCompositionPrototypes of the communicating EcuInstances (sender and possibly multiple receivers), along with a DataTypeMap that associates the ApplicationDataType to a matching ImplementationDataType that in turn refers to an SwBaseType.

3. Tier 1s are free to define arbitrary ImplementationDataType (with SwBaseType) and an ApplicationDataType (with physical semantics) in the application software components. If this SwBaseType is different than the one used for the serialization/deserialization, RTE is responsible for the type conversion together with possible scaling defined by the CompuMethod referenced from the ApplicationDataType of the PortPrototype of the RootSwCompositionPrototype and the CompuMethod referenced from the physicalProps of the SystemSignal.

   A second type conversion and scaling may be performed between the PortPrototype of the RootSwCompositionPrototype and the PortPrototype of the ApplicationSwComponentType.

   Please note that on the receiver side it is possible that the SwComponentPrototype that is typed by ApplicationSwComponentType receives only a subset of data defined on the RootSwCompositionPrototype. In this case, this needs be described by the PortInterfaceMapping.

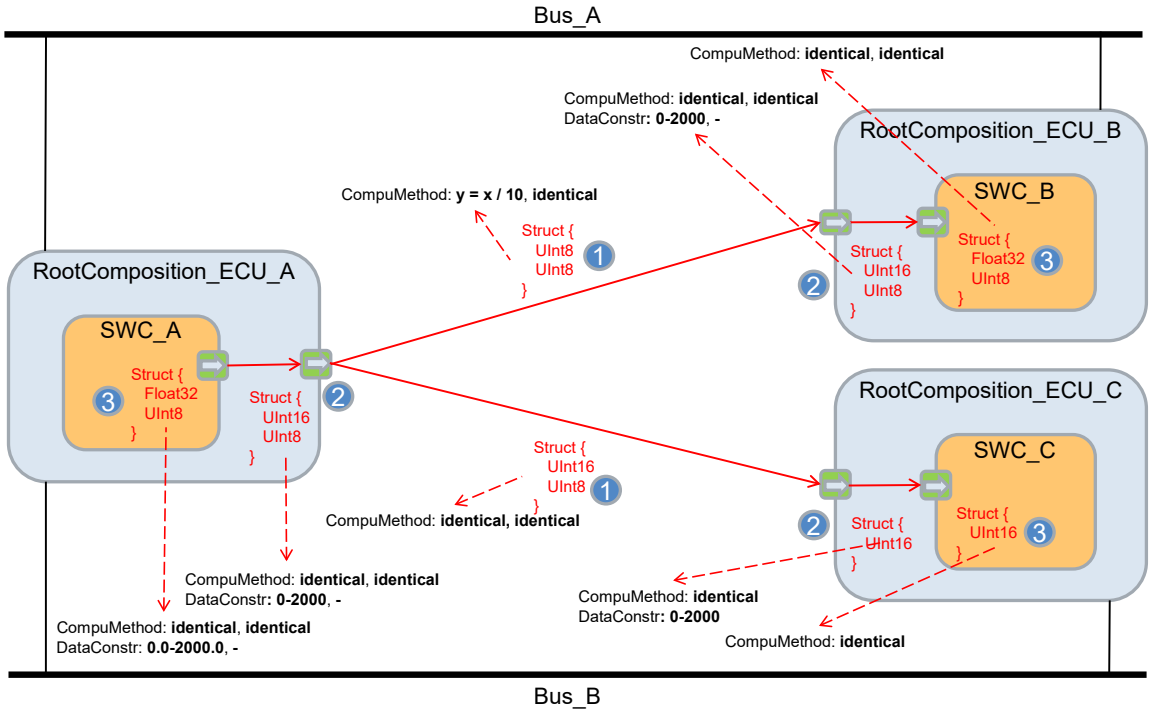

Figure 7.14: Serialization based on the network representation

The actual steps that need to be performed at runtime are presented in Figure 7.15 and they are as follows:

1. Application of the sending software component provides the data to be transmitted to the RTE and stores it in SWC internal buffer.
2. If SwBaseType defined by the ImplementationDataType on the PortPrototype of the ApplicationSwComponentType is different then the one optionally defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype, the RTE performs type conversion, and scaling if CompuMethods are also defined on these PortPrototypes, and stores the values internally in the RTE.
3. If network representation defines a SwBaseType that is different from the one optionally defined by the ImplementationDataType on the PortPrototype on the RootSwCompositionPrototype, the RTE performs another type conversion, and scaling if CompuMethod is also defined as part of the physicalProps, and stores the value internally in the RTE. This internal value is used for the serialization.
4. On the receiver side, the RTE stores the serialized data in the RTE internal buffer. When the receiver SWC wants to read the data, the RTE first de-serializes the values received from the bus whose type is specified by the SwBaseType that is part of the network representation. If the SwBaseType is different then the one optionally defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype, the RTE performs type conversion, and scaling if CompuMethods are also defined on the PortPrototype and physicalProps, and stores the values internally in the RTE.
5. If SwBaseType defined by the ImplementationDataType on the PortPrototype of the SwComponentPrototype that is typed by ApplicationSwComponentType is different then the one defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype, the RTE performs another type conversion, and scaling if CompuMethods are also defined on these PortPrototype, and stores the final values internally in the buffer of the application. The RTE is also able to deliver only a subset of data to the SwComponentPrototype that is typed by ApplicationSwComponentType, if that is required by the description of the PortInterfaceMapping.

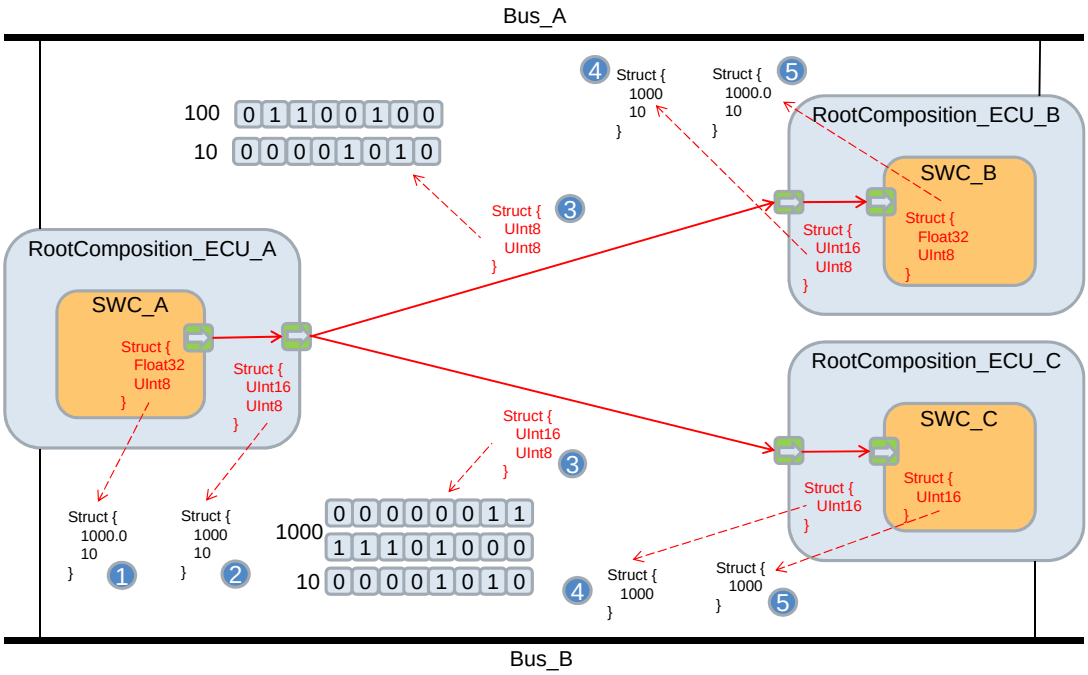

Figure 7.15: Serialization based on the network representation

7.3.2.2.1.1 Necessity of data conversion based on CompuMethods

CompuMethods are used to define how the information processed by AUTOSAR (the internal view) is to be interpreted in the physical domain. As an example the BatteryVoltage may be processed by AUTOSAR in a 1/10th scaling, thus an Integer value of 485 represents 48,5 Volt.

It is important to recognize that the value 48,5V is never handled by the AUTOSAR platform software. AUTOSAR only handles the internal representation (485).

Of course there are use-cases where the physical representation is required, e.g. monitoring tools show the measured values in the physical domain, thus have to perform a conversion. Another use-case is the dashboard where the BatteryVoltage might be displayed. But in this case it is the task of the displaying application software to perform the conversion into the physical domain, or rather into the visualization.

The AUTOSAR infrastructure (respectively the RTE) may still be required to perform a conversion of data. This is required if two CompuMethods - applying to the same data path - are not equal.

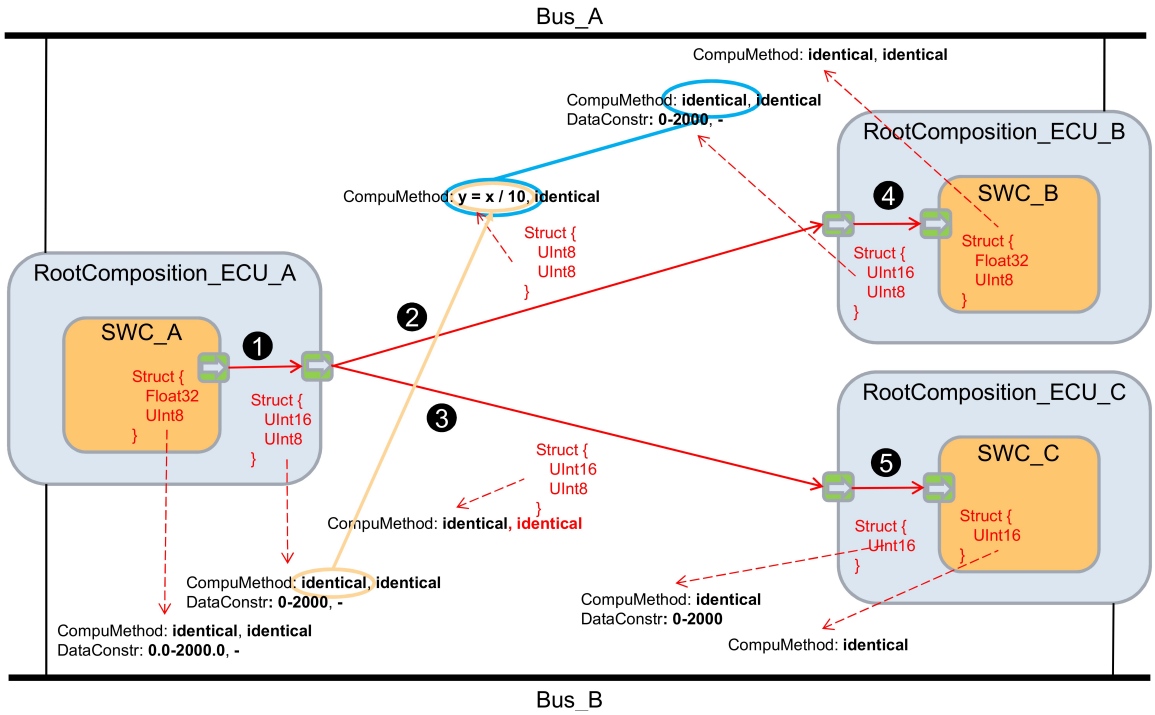

Figure 7.16: Data conversion based on CompuMethods

In the example figure 7.16 there are several CompuMethods defined for various occurrences of data: A value is produced in SWC_A with a BaseType Float32 and an "identical" CompuMethod ($y = x$). The value is then delegated to the RootComposition_ECU_A, where the BaseType is an uint16 and the CompuMethod is again "identical". For the network transport the CompuMethod is defined as $y = x/10$. As the two CompuMethods are not equal the RTE will have to convert the "identical" representation to the $y = x/10$ representation. On the reception side there is again an "identical" CompuMethod defined, thus another conversion by the receiving RTE is required.

7.3.2.2.2 Example - Serialization based on the ImplementationDataTypes

An example with concrete methodological steps in a common OEM-Tier 1 development process for the serialization based on the ImplementationDataTypes is presented in Figure 7.17. The steps are as follows:

1. OEM provides an ApplicationDataType (with physical semantics) on the PortPrototype on the RootSwCompositionPrototypes of the communicating EcuInstances (sender and possibly multiple receivers), along with a DataTypeMap that associates the ApplicationDataType to a matching ImplementationDataType that in turn refers to an SwBaseType.

2. Tier 1s are free to define arbitrary ImplementationDataType (with SwBaseType) and an ApplicationDataType (with physical semantics) in the application software components. If this SwBaseType is different than the one

used for the serialization/deserialization, RTE is responsible for the type conversion together with possible scaling defined by CompuMethods referenced from the ApplicationDataType of the PortPrototypes of the RootSwCompositionPrototype and AtomicSwComponentTypes.

Please note that on the receiver side it is possible that the SwComponentPrototype that is typed by ApplicationSwComponentType receives only a subset of data defined on the RootSwCompositionPrototype. In this case, this needs be described by the PortInterfaceMapping.

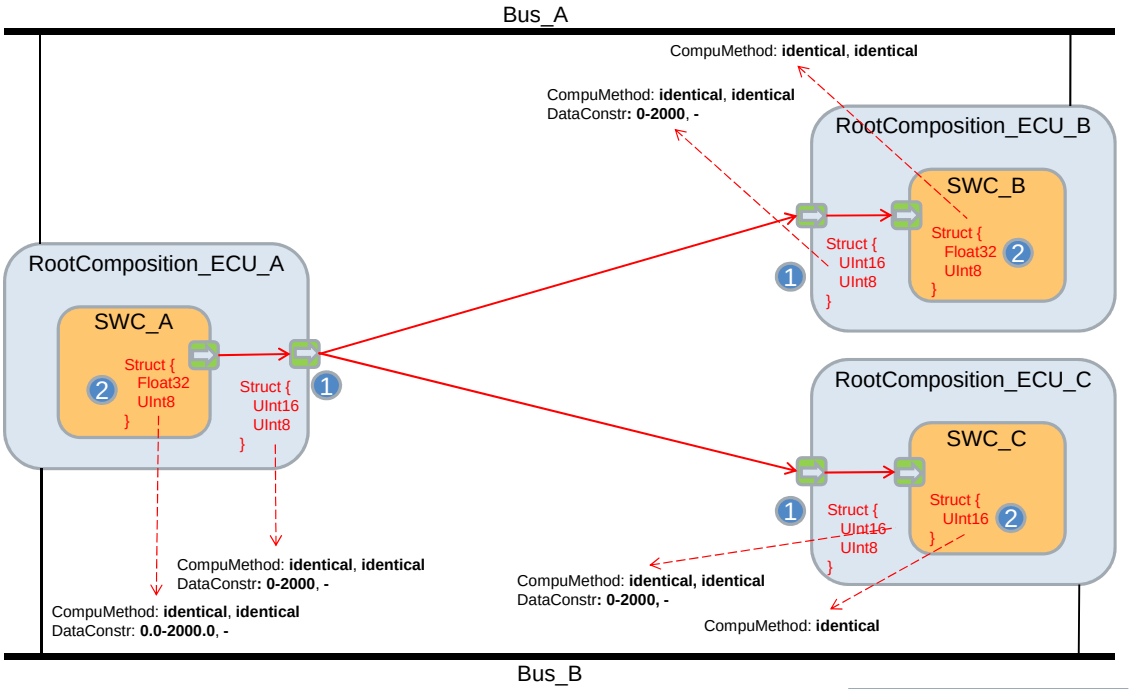

Figure 7.17: Serialization based on the ImplementationDataTypes

The actual steps that need to be performed at runtime are presented in Figure 7.18 and they are as follows:

1. Application of the sending software component provides the data to be transmitted to the RTE and stores it in SWC internal buffer.
2. If SwBaseType defined by the ImplementationDataType on the PortPrototype of the SwComponentPrototype that is typed by ApplicationSwComponentType is different then the one defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype, the RTE performs type conversion, and scaling if CompuMethods are also defined on these PortPrototypes, and stores the values internally in the RTE.
3. As no network representation is provided, the internal value from step 2 is used for the serialization and transmission on the bus.
4. On the receiver side, the RTE stores the serialized data received from the bus in the RTE internal buffer. When the receiver SWC wants to read the data, the

RTE de-serializes these values as defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype.

5. If SwBaseType defined by the ImplementationDataType on the PortPrototype of the SwComponentPrototype that is typed by ApplicationSwComponentType is different then the one defined by the ImplementationDataType on the PortPrototype of the RootSwCompositionPrototype, the RTE performs another type conversion, and scaling if CompuMethods are also defined on these PortPrototype, and stores the final values internally in the buffer of the application. The RTE is also able to deliver only a subset of data to the SwComponentPrototype that is typed by ApplicationSwComponentType, if that is required by the description of the PortInterfaceMapping.

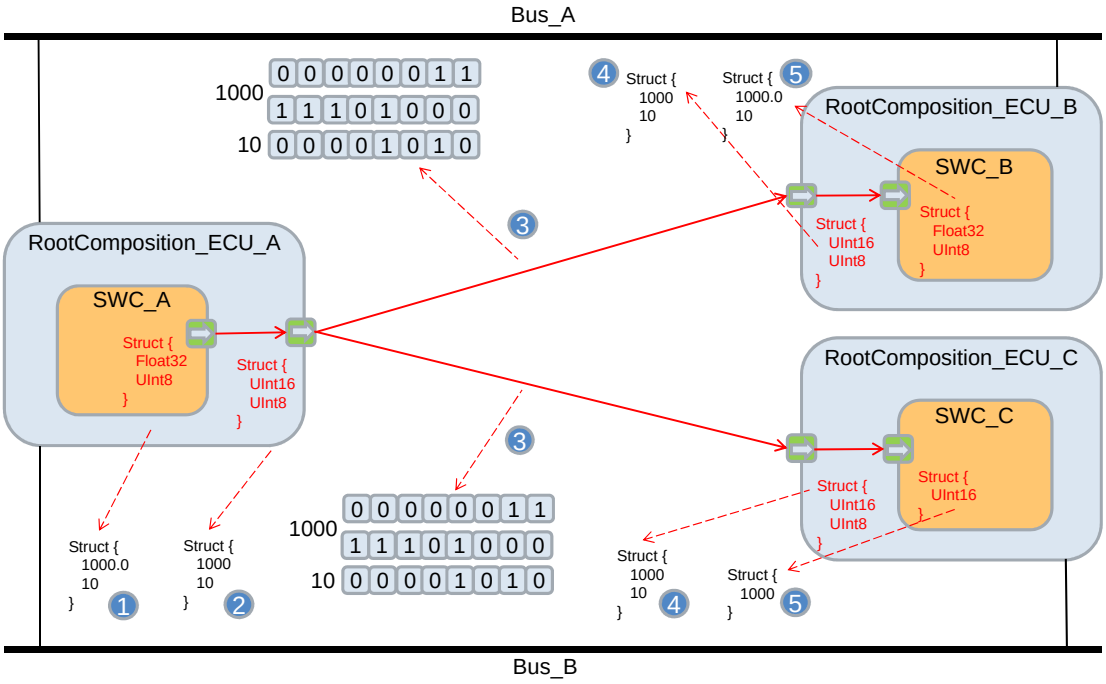

Figure 7.18: Serialization based on the ImplementationDataTypes

7.3.3 COM Based Transformer

In order to support the signal group based interaction of the transformer with the COM module as defined in the COM Based Transformer specification [17] a further modeling is supported:
In case the array based signal group API of Com shall be used the ISignalGroup has a reference to the DataTransformation element in the role comBasedSignalGroupTransformation. This defines that the RTE shall use the array based signal group API of Com [25] in order to transport the transformed data.

[TPS_SYST_02058] Usage of COM Based Transformer ⌈If

1. an ISignalGroup references a DataTransformation in the role comBasedSignalGroupTransformation and
2. this ISignalGroup references a SystemSignalGroup and
3. the referenced SystemSignalGroup is referenced by a SenderReceiverToSignalGroupMapping in the role signalGroup

then the VariableDataPrototype referenced by the SenderReceiverToSignalGroupMapping shall be transformed using the COM Based Transformer [17].⌋

[constr_5050] VariableDataPrototype of COM Based Transformer

Imposition time: CP: IT_SysDesc

⌈The VariableDataPrototype of [TPS_SYST_02058] shall be typed by an ApplicationRecordDataType or an ImplementationDataType of category STRUCTURE.⌋

Please note that according to [SWS_Rte_03867] the RTE calculates the InputBufferLength that is used for the output buffer calculation for Sender/Receiver communication needed for the VariableDataPrototype of the dataElement of the SenderReceiverInterface that shall be transformed.

[constr_3132] Required COM Based Transformation for comBasedSignalGroupTransformation

Imposition time: CP: IT_SysDesc

⌈If a ISignalGroup has a reference to the DataTransformation element in the role comBasedSignalGroupTransformation then this DataTransformation shall be the handled by the COM Based Transformer [17].⌋

Note that the SystemSignalGroup (and the corresponding ISignalGroup) in this case not only contains the application data element signals mapped by the SenderReceiverToSignalGroupMapping but also the data which has been added by the transformers (e.g. crc, sequence counter, ...). This is also shown in the example in Figure 7.20.

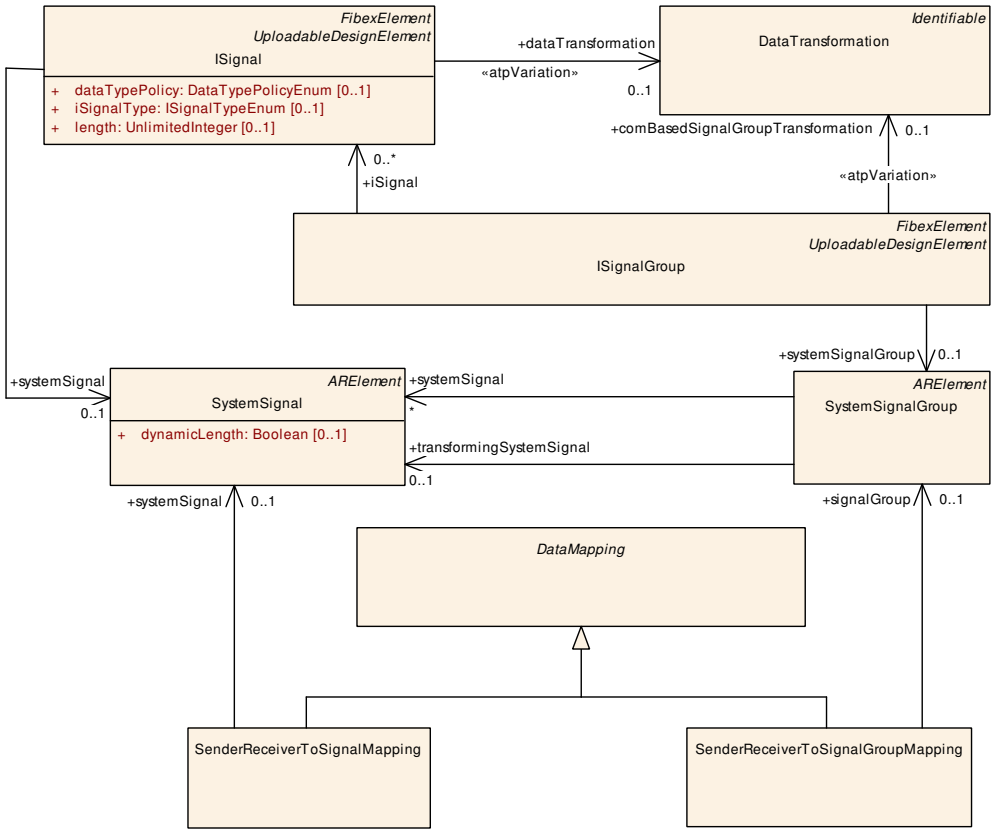

Figure 7.19: Transformer Data Mapping

[constr_3183] `ISignalGroup` with `transformationISignalProps`

Imposition time: CP: `IT_SysDesc`

⌈An `ISignalGroup` that aggregates `transformationISignalProps` shall reference the `DataTransformation` in the role `comBasedSignalGroup-Transformation`.⌋

[TPS_SYST_02068] Transformer header field representation in an `ISignalGroup`

⌈In case `ISignalGroup` has a reference to a `DataTransformation` in the role `comBasedSignalGroupTransformation` and the `DataTransformation` has further `TransformationTechnology`s defined in the role `transformerChain` then space for the individual headers shall be allocated by defining one `ISignal` per header part that is member of the `ISignalGroup`.⌋

[constr_3152] `BufferProperties.headerLength` settings for any transformer used in combination with a COM Based transformer

Imposition time: CP: `IT_SysDesc`

⌈A transformer used in a transformer chain with a COM Based transformer shall be configured with the following values:

- `BufferProperties.headerLength` = 0

⌋

This is because the space for the transformer headers (e.g. CRC and Sequence-Counter for E2E) needs to be allocated by a proper `ISignalGroup` layout according to [TPS_SYST_02068].

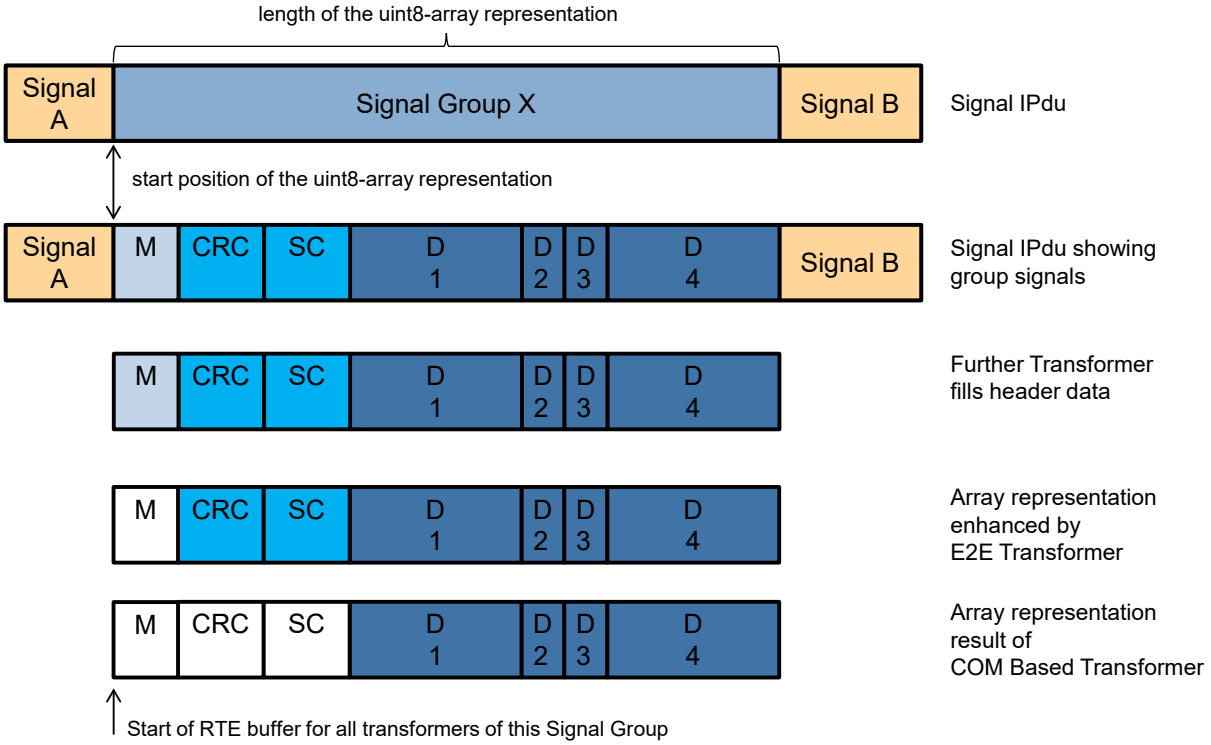

Figure 7.20: Example of COM Based Transformer buffer layout

As shown in figure 7.20 the example illustrates that for the E2E header ('CRC' and 'SC') and the 'M' header three further `ISignal`s are defined within the `ISignalGroup` in order to compensate the space required by the additional transformers.

7.3.4 DDS Transformer

The DDS communication stack is special in the sense that the actual data transformation (RTPS serialization) does not happen in scope of the RTE and transformers, but rather in the DDS basic software module itself. The DDS basic software module is located below the RTE in the layered software architecture [39].

The DDS transformer configuration has the mere purpose to configure the pass-through of the communicated data from the application software, via the RTE and Com to the DDS basic software module and vice versa. The classes `DdsTransformationDescription` and `DdsTransformationISignalProps` have no further attributes defined.

[constr_3821] DDS transformer configuration

Imposition time: CP: `IT_SysDesc`

⌈For each `TransformationDescription` variant that is a `DdsTransformationDescription`

- attribute `protocol` of `TransformationTechnology` shall be set to `Dds`
- attribute `version` of `TransformationTechnology` shall be set to `1.0.0`
- attribute `transformerClass` of `TransformationTechnology` shall be set to `serializer`
- attribute `hasInternalState` of `TransformationTechnology` shall be set to `false`
- attribute `needsOriginalData` of `TransformationTechnology` shall be set to `true`
- attribute `dataTransformationKind` of `DataTransformation` shall be set to `DataTransformationKindEnum.symmetric`
- attribute `executeDespiteDataUnavailability` of `DataTransformation` shall be set to `true`
- attribute `headerLength` of `BufferProperties` shall be set to `0` (bits)

⌋

As the DDS transformer is a pass-through transformer, no further transformers shall be defined in combination with the `DdsTransformationDescription`. The handling of safety and security is performed in scope of the DDS basic software module.

[constr_3822] No further transfomers in scope of the DDS transformer

Imposition time: CP: `IT_SysDesc`

⌈If a `DataTransformation` references a `TransformationTechnology` in the role `transformerChain` and that `TransformationTechnology` aggregates a `DdsTransformationDescription` in the role `transformationDescription`,

then that DataTransformation shall not reference any other TransformationTechnology.

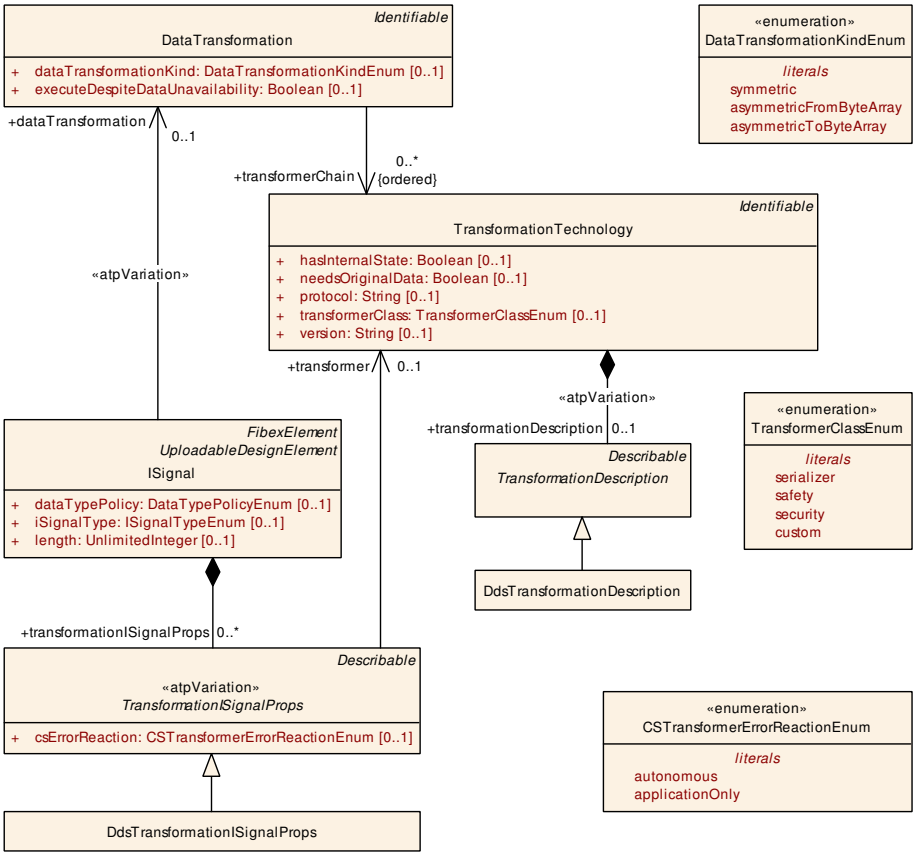

Figure 7.21: DDS Transformer

Class	**DdsTransformationDescription**			
Note	The DdsTransformationDescription is used to specify the DDS transformer specific attributes. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.25: DdsTransformationDescription

Class	«atpVariation» **DdsTransformationISignalProps**			
Note	The class DdsTransformationISignalProps specifies ISignal specific configuration properties for the DDS transformer. **Tags:** atp.Status=candidate			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 7.26: DdsTransformationISignalProps

7.3.5 E2E Transformer

This section specifies the configuration of the E2E protection that is invoked "out-of-box" by RTE and realized by E2E Transformer [47], E2E Library [48] and CRC Library [49].

The specific configuration for an E2E transformer takes place in EndToEndTransformationDescription and EndToEndTransformationISignalProps shown in Figure 7.22 and in EndToEndTransformationComSpecProps (see more details in [4])

[TPS_SYST_02275] Relation between EndToEndTransformationDescription and EndToEndTransformationComSpecProps ⌈It is possible to overwrite the ISignal specific E2E settings that are defined in EndToEndTransformationDescription with settings available in the EndToEndTransformationComSpecProps defined at the PortPrototype of a SwComponentType. With this approach it is possible to define Port-Prototype specific configuration options for the E2E data transformer.⌋

[constr_3763] Allowed value for maxDeltaCounter in the context of a profileName

Imposition time: CP: IT_SysDesc

⌈An EndToEndTransformationComSpecProps that is associated with an EndToEndTransformationDescription as described in [TPS_SYST_02275] shall not contain a maxDeltaCounter value that is outside the value range imposed by the profile defined in EndToEndTransformationDescription.profileName.

The profile specific value ranges are listed in [constr_3158], [constr_3195], [constr_3159], [constr_3196], [constr_3197], and [constr_3316].⌋

[constr_9331] E2E protection of a ClientServerOperation

Imposition time: CP: IT_SysDesc

⌈If an ISignal aggregates EndToEndTransformationISignalProps and references a SystemSignal that in turn is referenced by a ClientServerToSignalMapping in the role callSignal, then the EndToEndTransformationISignalProps settings and the EndToEndTransformationDescription settings defined in the TransformationTechnology that is referenced by the EndToEndTransformationISignalProps shall have the same values for the following attributes:

- EndToEndTransformationDescription.profileName
- EndToEndTransformationDescription.offset
- EndToEndTransformationISignalProps.sourceId

as for the ISignal that is referenced by the SystemSignal that in turn is referenced by the same ClientServerToSignalMapping in the role returnSignal.⌋

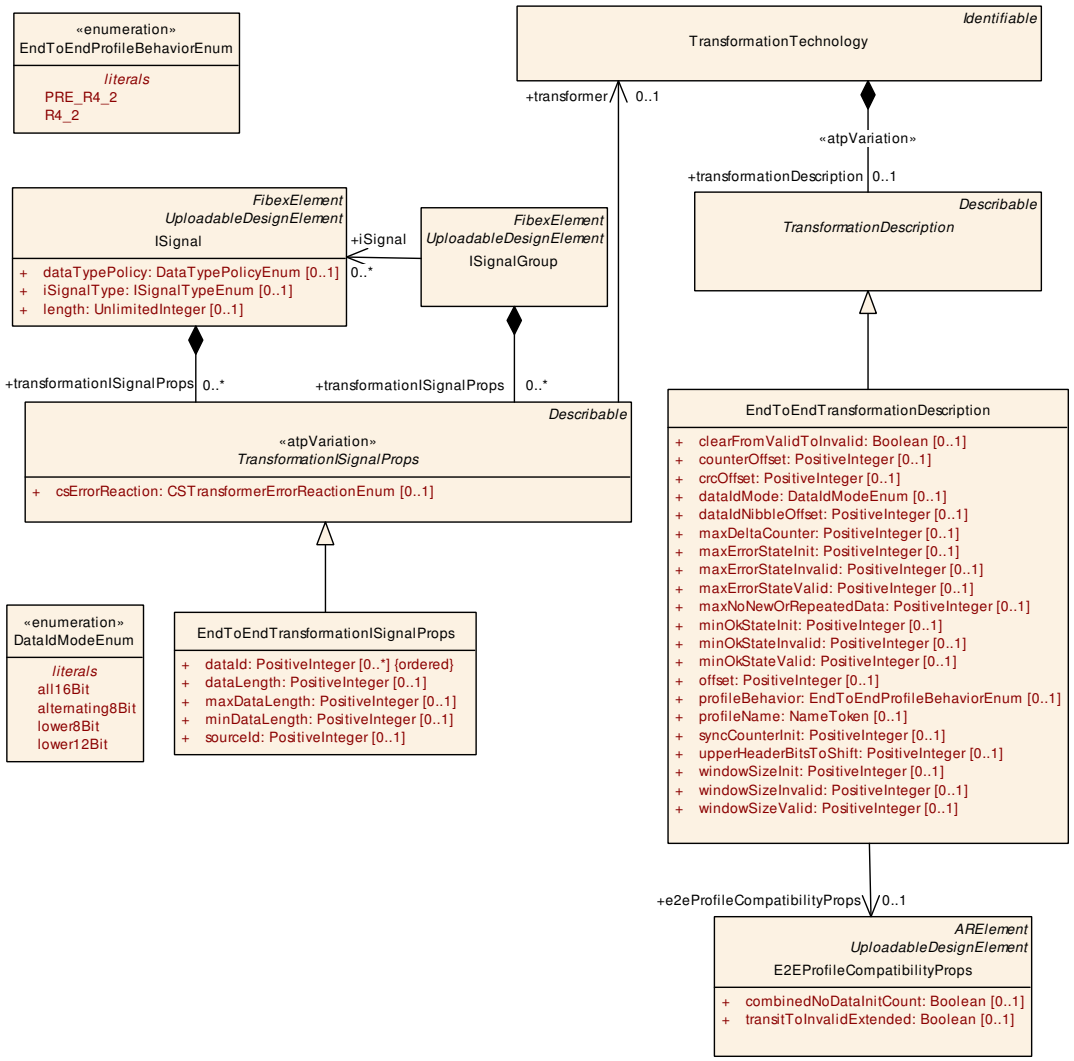

Figure 7.22: E2E Transformer Configuration

Class	**EndToEndTransformationDescription**			
Note	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clearFromValid ToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
counterOffset	PositiveInteger	0..1	attr	Offset of the counter in the Data[] array in bits.
crcOffset	PositiveInteger	0..1	attr	Offset of the CRC in the Data[] array in bits.
dataIdMode	DataIdModeEnum	0..1	attr	This attribute describes the inclusion mode that is used to include the implicit two-byte Data ID in the one-byte CRC.
dataIdNibble Offset	PositiveInteger	0..1	attr	Offset of the Data ID nibble in the Data[] array in bits.

△

Class	**EndToEndTransformationDescription**			
e2eProfile Compatibility Props	E2EProfileCompatibility Props	0..1	ref	Reference to additional settings for the E2E state machine. **Tags:** atp.Splitkey=e2eProfileCompatibilityProps
maxDelta Counter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and Max DeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorState Init	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INIT.
maxErrorState Invalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INVALID.
maxErrorState Valid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_VALID.
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	The maximum allowed amount of consecutive failed counter checks.
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT.
minOkState Invalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID.
minOkState Valid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID.
offset	PositiveInteger	0..1	attr	Offset of the E2E header in the Data[] array in bits.
profileBehavior	EndToEndProfile BehaviorEnum	0..1	attr	Behavior of the check functionality
profileName	NameToken	0..1	attr	Definition of the E2E profile.
syncCounterInit	PositiveInteger	0..1	attr	Number of checks required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
upperHeader BitsToShift	PositiveInteger	0..1	attr	This attribute describes the number of upper-header bits to be shifted. value = 0 or not present: shift of upper header is NOT performed. value > 0: the E2E Transformer on the protect-side, takes the first upperHeaderBitsToShift bits from the upper buffer (e.g. SOME/IP header part generated by SOME/IP transformer) and shifts them towards the lower bytes and bits within the Data[] for the length of the E2E header (e.g. 12 bytes in case of E2E Profile 4). This means the shift distance is fixed - it depends on the E2E header size - what is configured here is the number of bits that are to be shifted. This option is defined because the Some/IP header generated by SOME/IP transformer shall be, due to compatibility between non-protected and E2 E-protected communication, at the same position, which is before E2E header.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.

▽

△

Class	**EndToEndTransformationDescription**			
windowSize Invalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSize Valid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table 7.27: EndToEndTransformationDescription

Enumeration	**DataIdModeEnum**
Note	Supported inclusion modes to include the implicit two-byte Data ID in the one-byte CRC.
Aggregated by	E2EProfileConfiguration.dataIdMode, EndToEndTransformationDescription.dataIdMode
Literal	***Description***
all16Bit	Two bytes are included in the CRC (double ID configuration). **Tags:** atp.EnumerationLiteralIndex=0
alternating8Bit	One of the two bytes byte is included, alternating high and low byte, depending on parity of the counter (alternating ID configuration). For even counter low byte is included; For odd counters the high byte is included. **Tags:** atp.EnumerationLiteralIndex=1 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.
lower12Bit	The low byte is included in the implicit CRC calculation, the low nibble of the high byte is transmitted along with the data (i.e. it is explicitly included), the high nibble of the high byte is not used. This is applicable for the IDs up to 12 bits. **Tags:** atp.EnumerationLiteralIndex=2
lower8Bit	Only low byte is included, high byte is never used. This is applicable if the IDs in a particular system are 8 bits. **Tags:** atp.EnumerationLiteralIndex=3 This EnumerationLiteral is only used by the AUTOSAR Classic Platform.

Table 7.28: DataIdModeEnum

Class	**E2EProfileCompatibilityProps**			
Note	This meta-class collects settings for configuration of the E2E state machine. **Tags:** atp.recommendedPackage=E2EProfileCompatibilityPropsCollection			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
combinedNo DataInitCount	Boolean	0..1	attr	E2E State machine behavior concerning counting of detected counter errors and missing messages in states NODATA and INIT • value = 0 (false) or not defined: counting of detected counter errors and missing messages in states NODATA and INIT are counted per state separated (Autosar R23-11 or former behavior) • value = 1 (true): counting of detected counter errors and missing messages in states NODATA and INIT are counted in total

▽

△

Class	**E2EProfileCompatibilityProps**			
transitToInvalid Extended	Boolean	0..1	attr	E2E State machine behavior concerning transition from NODATA/INIT to INVALID value=0 (false): no direct transition from NODATA to INVALID, no transition from INIT to INVALID due to counter-related faults (Autosar R19-11 or former behavior) value=1 (true): direct transition from NODATA to INVALID covered, transition from INIT to INVALID due to counter-related faults covered (state machine extended)

Table 7.29: E2EProfileCompatibilityProps

Enumeration	**EndToEndProfileBehaviorEnum**
Note	Behavior of the check functionality
Aggregated by	EndToEndTransformationDescription.profileBehavior
Literal	*Description*
PRE_R4_2	Check has the legacy behavior, before AUTOSAR Release 4.2. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=PRE--R-4--2
R4_2	Check behaves like new P4/P5/P6 profiles introduced in AUTOSAR Release 4.2. **Tags:** atp.EnumerationLiteralIndex=1 xml.name=R-4--2

Table 7.30: EndToEndProfileBehaviorEnum

Class	«atpVariation» **EndToEndTransformationISignalProps**			
Note	Holds all the ISignal specific attributes for the EndToEndTransformer.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
dataId (ordered)	PositiveInteger	*	attr	This represents a unique numerical identifier. Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection.
dataLength	PositiveInteger	0..1	attr	Length of payload and E2E header in bits.
maxDataLength	PositiveInteger	0..1	attr	Maximum length of payload and E2E header in bits.
minDataLength	PositiveInteger	0..1	attr	Minimum length of payload and E2E header in bits.
sourceId	PositiveInteger	0..1	attr	This attribute represents a unique numerical identifier identifying the source of a certain transmission. In case of C/S communication, this ID uniquely identifies the client. Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection.

Table 7.31: EndToEndTransformationISignalProps

[constr_5229] Existence of attribute E2EProfileCompatibilityProps.transitToInvalidExtended is mandatory for each EndToEndTransformationDescription

Imposition time: CP: IT_SysDesc

⌈For each EndToEndTransformationDescription, a reference to E2EProfileCompatibilityProps in the role e2eProfileCompatibilityProps shall exist and the referenced E2EProfileCompatibilityProps shall define a value for the attribute transitToInvalidExtended.⌋

[constr_3313] E2E transformer configuration

Imposition time: CP: IT_SysDesc

⌈For each TransformationDescription variant that is a EndToEndTransformationDescription

- attribute protocol of TransformationTechnology shall be set to E2E
- attribute version of TransformationTechnology shall be set to 1.0.0
- attribute transformerClass of TransformationTechnology shall be set to safety

⌋

[TPS_SYST_02067] E2E profile ⌈The E2E profile is defined by EndToEndTransformationDescription.profileName.⌋

[constr_9287] Existence of EndToEndTransformationDescription.profileName

Imposition time: CP: IT_SysDesc

⌈For each EndToEndTransformationDescription, the attribute profileName shall exist.⌋

[TPS_SYST_02073] EndToEndTransformationDescription.profileName ⌈EndToEndTransformationDescription.profileName can have the following values: PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_08m, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_44, PROFILE_44m and PROFILE_76.⌋

[TPS_SYST_02072] profileName of EndToEndTransformationDescription ⌈The values for the profileName of EndToEndTransformationDescription mentioned in [TPS_SYST_02073] are standardized and reserved for being used in the way the AUTOSAR standard foresees. In addition, it is positively possible to use other than the standardized values for the profileName.⌋

The setting of the EndToEndTransformationDescription.profileName has an influence on the upper- and lower multiplicities of certain attributes of EndToEndTransformationDescription and EndToEndTransformationISignalProps.

[constr_3185] Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the EndToEndTransformationDescription.dataIdMode attribute shall be 1.⌋

[constr_3186] Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.dataIdMode attribute shall be 0.⌋

[constr_3326] Allowed values for EndToEndTransformationDescription.dataIdMode in PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_11 then the value of the EndToEndTransformationDescription.dataIdMode attribute shall be set to all16Bit or lower12Bit.⌋

[constr_3187] Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the EndToEndTransformationDescription.counterOffset attribute shall be 1.⌋

[constr_3188] Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.counterOffset attribute shall be 0.⌋

[constr_3189] Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the EndToEndTransformationDescription.crcOffset attribute shall be 1.⌋

[constr_3190] Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.crcOffset attribute shall be 0.⌋

[constr_3193] Multiplicity of EndToEndTransformationDescription.offset in PROFILE_01 and PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the EndToEndTransformationDescription.offset attribute shall be 0.⌋

[constr_3194] Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the EndToEndTransformationDescription.offset attribute shall be 1.⌋

[constr_3191] Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_01, PROFILE_11 and dataIdMode equal to lower12Bit

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01 or PROFILE_11 and the value of the EndToEndTransformationDescription.dataIdMode attribute is set to lower12Bit then the multiplicity

of the EndToEndTransformationDescription.dataIdNibbleOffset attribute shall be 1.⌋

[constr_3192] Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76 or dataIdMode different from lower12Bit

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 or the EndToEndTransformationDescription.dataIdMode attribute is set to value different from lower12Bit then the multiplicity of the EndToEndTransformationDescription.dataIdNibbleOffset attribute shall be 0.⌋

[constr_3148] executeDespiteDataUnavailability setting in case an E2E Transformer is used

Imposition time: CP: IT_SysDesc

⌈A transformer chain using E2E shall be configured with DataTransformation.executeDespiteDataUnavailability = TRUE.⌋

[constr_3149] TransformationTechnology.needsOriginalData settings for E2E Transformer

Imposition time: CP: IT_SysDesc

⌈The TransformationTechnology.needsOriginalData attribute of a TransformationTechnology element of an E2E transformer shall be set to FALSE.⌋

[constr_3151] BufferProperties.headerLength settings for an E2E transformer used in combination with a SOME/IP transformer

Imposition time: CP: IT_SysDesc

⌈The BufferProperties.headerLength for an E2E transformer located in a transformer chain with a SOME/IP transformer shall be configured with the following values depending on the value of the EndToEndTransformationDescription.profileName attribute:

1. PROFILE_01: BufferProperties.headerLength = 16 bits
2. PROFILE_02: BufferProperties.headerLength = 16 bits
3. PROFILE_04: BufferProperties.headerLength = 96 bits
4. PROFILE_05: BufferProperties.headerLength = 24 bits
5. PROFILE_06: BufferProperties.headerLength = 40 bits

6. PROFILE_07: `BufferProperties.headerLength` = 160 bits
7. PROFILE_08: `BufferProperties.headerLength` = 128 bits
8. PROFILE_11: `BufferProperties.headerLength` = 16 bits
9. PROFILE_22: `BufferProperties.headerLength` = 16 bits
10. PROFILE_04m: `BufferProperties.headerLength` = 128 bits
11. PROFILE_07m: `BufferProperties.headerLength` = 192 bits
12. PROFILE_08m: `BufferProperties.headerLength` = 160 bits
13. PROFILE_44: `BufferProperties.headerLength` = 96 bits
14. PROFILE_44m: `BufferProperties.headerLength` = 128 bits

⌋

This means that the E2E header in profiles 1 and 2 use 2 bytes when using SOME/IP transformer. This yields four unused bits in case of some recommended configuration settings of Profile 1 and 2. Those unused bits are set to 0xF by the E2E transformer on the sender side.

[constr_3153] E2E header field reservation required by COM Based transformer

Imposition time: CP: `IT_SysDesc`

⌈A COM Based transformer that is used in a transformer chain with an E2E transformer requires that the following amount of space is allocated for the E2E header fields using a proper `ISignalGroup` layout according to [TPS_SYST_02068]:

PROFILE_01: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_01: if `dataIdMode` != `lower12Bit`: 12 bits

PROFILE_02: 16 bits

PROFILE_04: 96 bits

PROFILE_05: 24 bits

PROFILE_06: 40 bits

PROFILE_07: 160 bits

PROFILE_08: 128 bits

PROFILE_11: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_11: if `dataIdMode` == `all16Bit`: 12 bits

PROFILE_22: 16 bits

PROFILE_04m: 128 bits

PROFILE_07m: 192 bits

PROFILE_08m: 160 bits

PROFILE_44: 96 bits

PROFILE_44m: 128 bits

PROFILE_76: 40 bits

⌋

[constr_3184] Only one `EndToEndTransformationISignalProps.dataId` element in PROFILE_01 and PROFILE_11

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationISignalProps.dataId` attribute shall be 1.⌋

[constr_3156] Allowed values for `EndToEndTransformationISignalProps.dataId` in PROFILE_01 and PROFILE_11

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 0-65535.⌋

[constr_3157] Allowed values for `EndToEndTransformationISignalProps.dataId` in PROFILE_01 and PROFILE_11 in case `dataIdMode` is set to `lower12Bit`

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 and the value of `EndToEndTransformationDescription.dataIdMode` attribute has a value of `lower12Bit` then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 256-65535.⌋

[constr_3158] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_01 and PROFILE_11

Imposition time: CP: `IT_SysDesc`

⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the attribute `maxDeltaCounter` shall be in the range 1-14.⌋

[constr_3195] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02 and PROFILE_22

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02 or PROFILE_22 then the attribute maxDeltaCounter shall be in the range 1-15.⌋

[constr_3159] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_04, PROFILE_04m PROFILE_44 and PROFILE_44m

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_04m, PROFILE_44, or PROFILE_44m the value of maxDeltaCounter attribute shall be in the range 1-65535.⌋

[constr_3196] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_05

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_05 then the attribute maxDeltaCounter shall be in the range 1-255.⌋

[constr_3197] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_06

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_06 then the attribute maxDeltaCounter shall be in the range 1-255.⌋

[constr_3316] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07, PROFILE_08, PROFILE_07m and PROFILE_08m

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_07, PROFILE_08, PROFILE_07m, or PROFILE_08m the value of maxDeltaCounter attribute shall be in the range 1-4‘294‘967‘295.⌋

[constr_3160] EndToEndTransformationISignalProps.dataId in PROFILE_02 and PROFILE_22

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02 or PROFILE_22 then the multiplicity of the dataId attribute shall be 16 and the value of each instance shall be in the range 0..255.⌋

[constr_5220] Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_04m, PROFILE_07m, PROFILE_08m and PROFILE_44m

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_04m, PROFILE_07m, PROFILE_08m, or PROFILE_44m then the multiplicity of the EndToEndTransformationISignalProps.sourceId attribute shall be 1.⌋

[constr_5221] Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute is set to PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, or PROFILE_76 then the multiplicity of the EndToEndTransformationISignalProps.sourceId attribute shall be 0.⌋

[constr_3161] EndToEndTransformationISignalProps.dataLength in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, or PROFILE_22 then the multiplicity of the EndToEndTransformationISignalProps.dataLength attribute shall be 1.⌋

[constr_3162] EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, or PROFILE_22 then the multiplicity of the attributes EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength shall be 0.⌋

[constr_3163] EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then

the multiplicity of the attributes EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength shall be 1.⌋

[constr_3164] EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then the multiplicity of the attribute EndToEndTransformationISignalProps.dataLength shall be 0.⌋

[constr_3533] EndToEndTransformationISignalProps.dataLength shall be a multiple of 8

Imposition time: CP: IT_SysDesc

⌈The value of EndToEndTransformationISignalProps.dataLength, EndToEndTransformationISignalProps.maxDataLength, and EndToEndTransformationISignalProps.minDataLength shall be a multiple of 8.⌋

[constr_3155] Allowed values for EndToEndTransformationDescription.upperHeaderBitsToShift

Imposition time: CP: IT_SysDesc

⌈The value of of the EndToEndTransformationDescription.upperHeaderBitsToShift attribute depends on the used serializing transformer:

COM based transformer: 0 (no bits are shifted)

SOME/IP transformer: 64 (to support the header shift of SOME/IP).

Custom transformer: no restriction (depends on header length and placement of custom transformer)

⌋

[constr_3165] Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01, PROFILE_11

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_01 or PROFILE_11 and the serializing transformer is different than the ComBasedTransformer then:

1. EndToEndTransformationDescription.crcOffset shall be set to the same value of upperHeaderBitsToShift.

2. EndToEndTransformationDescription.counterOffset shall be set to the value of upperHeaderBitsToShift + 8.
3. (if used) EndToEndTransformationDescription.dataIdNibbleOffset shall be set to the value of upperHeaderBitsToShift + 12.

⌋

This means that the E2E header of profile 1 and profile 11, when used with SOME/IP Transformer or a Custom Transformer, is not spread across application data, but is a consecutive block of bytes. The layout flexibility available with these E2E protection profiles is therefore only supported if the ComBasedTransformer is used in combination with the E2E Transformer.

[constr_3166] EndToEndTransformationDescription.upperHeaderBitsToShift in PROFILE_02

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02 then the value of the upperHeaderBitsToShift attribute shall be 0.⌋

[constr_3169] EndToEndTransformationDescription.offset value in PROFILE_02, PROFILE_22 and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02, PROFILE_22, or PROFILE_76 then the value of the EndToEndTransformationDescription.offset attribute shall be 0.⌋

The E2E profiles PROFILE_02, PROFILE_22 and PROFILE_76 are not intended to be used with the SOME/IP Transformer.

[constr_3167] Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m the value of the EndToEndTransformationDescription.offset attribute shall be equal to the value of the EndToEndTransformationDescription.upperHeaderBitsToShift attribute.⌋

[TPS_SYST_02194] Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m ⌈If the EndToEndTransformationDescription.profileName attribute has a value of PPROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44m or PROFILE_44 the E2E protected data is identified by a EndToEndTransformationISignalProps.dataId.⌋

In other words if a SystemSignal defines the E2E protected data and a fanout is described by several ISignals that point to this SystemSignal, the dataId in each of those ISignals may have the same value.

[constr_3172] Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_01

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_01 and the value of the profileBehavior attribute is R4_2 then:

- the value of the EndToEndTransformationDescription.maxNoNewOrRepeatedData attribute shall be 14.
- the value of the EndToEndTransformationDescription.syncCounterInit attribute shall be 1.

⌋

[constr_3173] Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_02

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_02 and the value of the profileBehavior attribute is R4_2 then:

- the value of the EndToEndTransformationDescription.maxNoNewOrRepeatedData attribute shall be 15.
- the value of the EndToEndTransformationDescription.syncCounterInit attribute shall be 1.

⌋

[constr_3174] EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76

Imposition time: CP: IT_SysDesc

⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08,

PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, or PROFILE_76 then:

1. the multiplicity of the EndToEndTransformationDescription.maxNoNewOrRepeatedData attribute shall be 0.
2. the multiplicity of the EndToEndTransformationDescription.syncCounterInit attribute shall be 0.
3. the multiplicity of the EndToEndTransformationDescription.profileBehavior attribute shall be 0.

⌋

The EndToEndTransformationDescription may be differently chosen for a given ISignal or ISignalGroup depending on selected variant, with the following exceptions:

[constr_3182] Restriction on TransformationTechnology.transformationDescription VariationPoint

Imposition time: CP: IT_SysDesc

⌈The EndToEndTransformationDescription.profileName attribute shall not be subject to variability for a given ISignal / ISignalGroup, i.e., the value of the EndToEndTransformationDescription.profileName attribute shall be the same in all different variants.⌋

In other words, it is not possible that in one variant PROFILE_04 is used, and in another variant PROFILE_05 is used for the same ISignal or ISignalGroup.

7.3.5.1 E2E state machine settings

E2E state machine settings are set in EndToEndTransformationDescription and a subset of them can be overridden in EndToEndTransformationComSpecProps. The E2E state machine is described in more detail in the E2E Protocol specification [50].

Please note that the configuration of the E2E state machines with the configuration attributes available in EndToEndTransformationDescription is restricted by [PRS_E2E_01457], [PRS_E2E_01460], [PRS_E2E_01461], [PRS_E2E_01462], [PRS_E2E_01463], [PRS_E2E_01464] defined in [50].

[TPS_SYST_02419] Semantics of E2EProfileCompatibilityProps.combinedNoDataInitCount ⌈The E2EProfileCompatibilityProps.combinedNoDataInitCount defines the behavior of the E2E state machine for all E2E profiles.⌋

7.3.5.2 E2E recommended configuration settings

This chapter provides different configuration settings for particular E2E Profiles. Please note that in future additional recommended configuration settings might be added.

7.3.5.2.1 E2E Profile 1 configuration setting C

Caveat: Since the E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and since it is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR), support for the E2E wrapper approach will eventually be discontinued by AUTOSAR. Thus those AUTOSAR artifacts (e.g., specification items, meta classes) are to be considered as obsolete according to [TPS_STDT_00064] and will be removed from AUTOSAR with R25-11. New projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach.

The E2E Profile 1 configuration setting C is foreseen for CAN/FlexRay messages that are serialized by the COM based transformer and should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using Rte_Send / Rte_Receive.

[TPS_SYST_02069] Recommended configuration settings for E2E Profile 1 configuration setting C ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_01	Profile 1
EndToEndTransformationDescription.crcOffset	0	CRC offset
EndToEndTransformationDescription.counterOffset	8	Counter offset
EndToEndTransformationDescription.dataIdNibble-Offset	12	Data Id high nibble offset
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2, i.e. one lost message.
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Init	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSize-Invalid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSizeInit	2	Last 2 messages are considered
EndToEndTransformationDescription.minOkStateValid	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Valid	1	One error allowed
EndToEndTransformationDescription.minOkState-Invalid	2	At least two OK messages
EndToEndTransformationDescription.maxErrorState-Invalid	1	One error allowed

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.upperHeaderBits-ToShift	0	no bits are shifted
BufferProperties.headerLength	16	16 bits is the length of E2E profile 1C header.
EndToEndTransformationDescription.profileBehavior	R4_2	Behavior of Profile P1 adjusted for the state machine.
EndToEndTransformationDescription.maxNoNewOr-RepeatedData	14	Behavior of Profile P1 adjusted for the state machine.
EndToEndTransformationDescription.syncCounterInit	1	Behavior of Profile P1 adjusted for the state machine.

⌋

7.3.5.2.2 E2E Profile 4 configuration setting A

The E2E Profile 4 configuration setting A is foreseen for long messages that are serialized by the SOME/IP transformer. The configuration setting 4A should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using Rte_Send / Rte_Receive.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1.
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02070] Recommended configuration settings for E2E Profile 4 configuration setting A ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_04	Profile 4
EndToEndTransformationDescription.offset	64	To support the fixed location of Some/IP header
EndToEndTransformationDescription.maxDeltaCounter	1	Maximum jump considered to be OK is 1
EndToEndTransformationDescription.minOkStateInit	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Init	0	No errors allowed
EndToEndTransformationDescription.windowSizeValid	1	Only the last message is considered
EndToEndTransformationDescription.windowSize-Invalid	1	Only the last message is considered
EndToEndTransformationDescription.windowSizeInit	1	Only the last message is considered
EndToEndTransformationDescription.minOkStateValid	1	received message shall be OK.

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.maxErrorState-Valid	0	No errors allowed
EndToEndTransformationDescription.minOkState-Invalid	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Invalid	0	No errors allowed
EndToEndTransformationDescription.upperHeaderBits-ToShift	64	64 bits from Some/IP header to be shifted
BufferProperties.headerLength	96	96 bits is the length of E2E profile 4 header.

⌋

7.3.5.2.3 E2E Profile 4 configuration setting B

The E2E Profile 4 configuration setting B is foreseen for long messages that are serialized by the SOME/IP transformer. The configuration setting 4B should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using Rte_Send / Rte_Receive.

This configuration setting requires having within the monitoring window the following properties:

1. At least one OK message
2. At most one error not related to counters (e.g. CRC, data ID, length)
3. the remaining data in the monitoring window may be
   - repetitions or
   - jumps above 1.

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02071] Recommended configuration settings for E2E Profile 4 configuration setting B ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_04	Profile 4
EndToEndTransformationDescription.offset	64	To support the fixed location of Some/IP header
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2, i.e. one lost message
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Init	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	3	Last 3 messages are considered

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.windowSize-Invalid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSizeInit	2	Last two last messages are considered
EndToEndTransformationDescription.minOkStateValid	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Valid	1	One error allowed
EndToEndTransformationDescription.minOkState-Invalid	2	At least two OK messages
EndToEndTransformationDescription.maxErrorState-Invalid	1	One error allowed
EndToEndTransformationDescription.upperHeaderBits-ToShift	64	64 bits from Some/IP header to be shifted
BufferProperties.headerLength	96	96 bits is the length of E2E profile 4 header.

⌋

7.3.5.2.4 E2E Profile 5

The E2E Profile 5 configuration setting is foreseen for legacy communication and for messages that are serialized by the SOME/IP transformer. The configuration setting should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using Rte_Send / Rte_Receive.

This configuration setting requires having within the monitoring window the following properties:

1. At least one OK message in a window of 2
2. At most one error not related to counters (e.g. wrong CRC, data ID, or length)
3. At most one repetition or jump above 2

[TPS_SYST_02379] Recommended configuration settings for E2E Profile 5 configuration setting ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_05	Profile 5
EndToEndTransformationDescription.offset	64	To support the fixed location of Some/IP header
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Init	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	2	Last 2 messages are considered
EndToEndTransformationDescription.windowSize-Invalid	2	Last 2 messages are considered

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.windowSizeInit	2	Last 2 messages are considered
EndToEndTransformationDescription.minOkStateValid	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateValid	1	One error allowed
EndToEndTransformationDescription.minOkStateInvalid	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateInvalid	1	One error allowed
EndToEndTransformationDescription.upperHeaderBitsToShift	64	64 bits from Some/IP header to be shifted
BufferProperties.headerLength	24	24 bits is the length of E2E profile 5 header.

⌋

7.3.5.2.5 E2E Profile 7 configuration setting A

The E2E Profile 7 configuration setting A is foreseen for long messages (up to 4 MB) that are serialized by the SOME/IP transformer.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1.
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02134] Recommended configuration settings for E2E Profile 7 configuration setting A ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_07	Profile 7
EndToEndTransformationDescription.offset	64	To support the fixed location of Some/IP header
EndToEndTransformationDescription.maxDeltaCounter	1	Maximum jump considered to be OK is 1
EndToEndTransformationDescription.minOkStateInit	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorStateInit	0	No errors allowed
EndToEndTransformationDescription.windowSizeValid	1	Only the last message is considered
EndToEndTransformationDescription.windowSizeInvalid	1	Only the last message is considered
EndToEndTransformationDescription.windowSizeInit	1	Only the last message is considered
EndToEndTransformationDescription.minOkStateValid	1	received message shall be OK.

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.maxErrorState-Valid	0	No errors allowed
EndToEndTransformationDescription.minOkState-Invalid	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Invalid	0	No errors allowed
EndToEndTransformationDescription.upperHeaderBits-ToShift	64	64 bits from Some/IP header to be shifted
BufferProperties.headerLength	160	160 bits is the length of E2E profile 7 header.

⌋

7.3.5.2.6 E2E Profile 7 configuration setting B

The E2E Profile 7 configuration setting B is foreseen for long messages (up to 4 MB) that are serialized by the SOME/IP transformer.

This configuration setting requires having within the monitoring window the following properties:

1. At least one OK message
2. At most one error not related to counters (e.g. CRC, data ID, length)
3. the remaining data in the monitoring window may be
   - repetitions or
   - jumps above 1.

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02135] Recommended configuration settings for E2E Profile 7 configuration setting B ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_07	Profile 7
EndToEndTransformationDescription.offset	64	To support the fixed location of Some/IP header
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2, i.e. one lost message
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorState-Init	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSize-Invalid	3	Last 3 messages are considered

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.windowSizeInit	2	Last two messages are considered
EndToEndTransformationDescription.minOkStateValid	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateValid	1	One error allowed
EndToEndTransformationDescription.minOkStateInvalid	2	At least two OK messages
EndToEndTransformationDescription.maxErrorStateInvalid	1	One error allowed
EndToEndTransformationDescription.upperHeaderBitsToShift	64	64 bits from Some/IP header to be shifted
BufferProperties.headerLength	160	160 bits is the length of E2E profile 7 header.

⌋

7.3.5.2.7 E2E Profile 11 configuration setting C

The E2E Profile 11 configuration setting C is foreseen for CAN/FlexRay messages that are serialized by the COM based transformer and should be used with RTE event-based communication, i.e. queued communication with queue size = 1, using Rte_Send / Rte_Receive.

[TPS_SYST_02155] Recommended configuration settings for E2E Profile 11 configuration setting C ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_11	Profile 11
EndToEndTransformationDescription.crcOffset	0	CRC offset
EndToEndTransformationDescription.counterOffset	8	Counter offset
EndToEndTransformationDescription.dataIdNibbleOffset	12	Data Id high nibble offset
EndToEndTransformationDescription.maxDeltaCounter	2	Maximum jump considered to be OK is 2, i.e. one lost message
EndToEndTransformationDescription.minOkStateInit	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateInit	1	One error allowed
EndToEndTransformationDescription.windowSizeValid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSizeInvalid	3	Last 3 messages are considered
EndToEndTransformationDescription.windowSizeInit	2	Last two messages are considered
EndToEndTransformationDescription.minOkStateValid	1	At least one OK message
EndToEndTransformationDescription.maxErrorStateValid	1	One error allowed
EndToEndTransformationDescription.minOkStateInvalid	2	At least two OK messages

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.maxErrorState-Invalid	1	One error allowed
EndToEndTransformationDescription.upperHeaderBits-ToShift	0	no bits are shifted
BufferProperties.headerLength	16	16 bits is the length of E2E profile 1C header.

⌋

7.3.5.2.8 E2E Profile 4m configuration setting A

The E2E Profile 4m configuration setting A is foreseen for long messages that are serialized by e.g. the SOME/IP transformer.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1 at the source side
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02349] Recommended configuration settings for E2E Profile 4m configuration setting A ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_04m	Profile 4m
EndToEndTransformationDescription.offset	0	Header added by the serializing transformer (e.g., SOME/IP transformer) is not included in CRC calculation
EndToEndTransformationDescription.maxDeltaCounter - source (client)	1	Maximum jump considered to be OK is 1 for the source side
EndToEndTransformationDescription.maxDeltaCounter - sink (server)	$2^{16}-1$	No counter-based checks on the sink side
EndToEndTransformationDescription.minOkStateInit	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Init	0	No errors allowed
EndToEndTransformationDescription.windowSizeValid	8	Last 8 messages are considered
EndToEndTransformationDescription.windowSize-Invalid	8	Last 8 messages are considered
EndToEndTransformationDescription.windowSizeInit	8	Last 8 messages are considered
EndToEndTransformationDescription.minOkStateValid	2	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Valid	0	No errors allowed
EndToEndTransformationDescription.minOkState-Invalid	1	received message shall be OK.

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.maxErrorState-Invalid	0	No errors allowed
EndToEndTransformationDescription.upperHeaderBits-ToShift	x	depends on the length of the header added by the serializing transformer (e.g., 64 in case of the SOME/IP transformer)
BufferProperties.headerLength	128	128 bits is the length of E2E profile 4m header.

⌋

Please note that the ISignal that represents the ClientServerOperation request shall refer an EndToEndTransformationDescription that defines the maxDeltaCounter for the sink. The ISignal that represents the ClientServerOperation response shall refer an EndToEndTransformationDescription that defines the maxDeltaCounter for the source.

7.3.5.2.9 E2E Profile 7m configuration setting A

The E2E Profile 7m configuration setting A is foreseen for long messages (up to 4 MB) that are serialized by e.g. the SOME/IP transformer.

This configuration setting is quite strict as it does not allow any errors, i.e.:

1. Repetitions
2. Counter jumps bigger than 1 at the source side
3. Errors not related to counters (e.g. CRC, data ID, length)

As soon as any error is detected, there is a transition to invalid state.

[TPS_SYST_02350] Recommended configuration settings for E2E Profile 7m configuration setting A ⌈

Attribute	Allowed value	Comment
EndToEndTransformationDescription.profileName	PROFILE_07m	Profile 7m
EndToEndTransformationDescription.offset	0	Header added by the serializing transformer (e.g., SOME/IP transformer) is not included in CRC calculation
EndToEndTransformationDescription.maxDeltaCounter - source (client)	1	Maximum jump considered to be OK is 1 for the source side
EndToEndTransformationDescription.maxDeltaCounter - sink (server)	$2^{32}-1$	No counter-based checks on the sink side
EndToEndTransformationDescription.minOkStateInit	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorState-Init	0	No errors allowed
EndToEndTransformationDescription.windowSizeValid	6	Last 6 messages are considered

▽

△

Attribute	Allowed value	Comment
EndToEndTransformationDescription.windowSizeInvalid	6	Last 6 messages are considered
EndToEndTransformationDescription.windowSizeInit	6	Last 6 messages are considered
EndToEndTransformationDescription.minOkStateValid	2	received message shall be OK.
EndToEndTransformationDescription.maxErrorStateValid	0	No errors allowed
EndToEndTransformationDescription.minOkStateInvalid	1	received message shall be OK.
EndToEndTransformationDescription.maxErrorStateInvalid	0	No errors allowed
EndToEndTransformationDescription.upperHeaderBitsToShift	x	depends on the length of the header added by the serializing transformer (e.g., 64 in case of the SOME/IP transformer)
BufferProperties.headerLength	192	192 bits is the length of E2E profile 4m header.

⌋

Please note that the ISignal that represents the ClientServerOperation request shall refer an EndToEndTransformationDescription that defines the maxDeltaCounter for the sink. The ISignal that represents the ClientServerOperation response shall refer an EndToEndTransformationDescription that defines the maxDeltaCounter for the source.

7.3.6 UserDefined Transformer

Autosar allows to describe custom Transformers that are not standardized by AUTOSAR. This is done by the usage of the following elements:

- UserDefinedTransformationDescription
- UserDefinedTransformationISignalProps
- UserDefinedTransformationProps

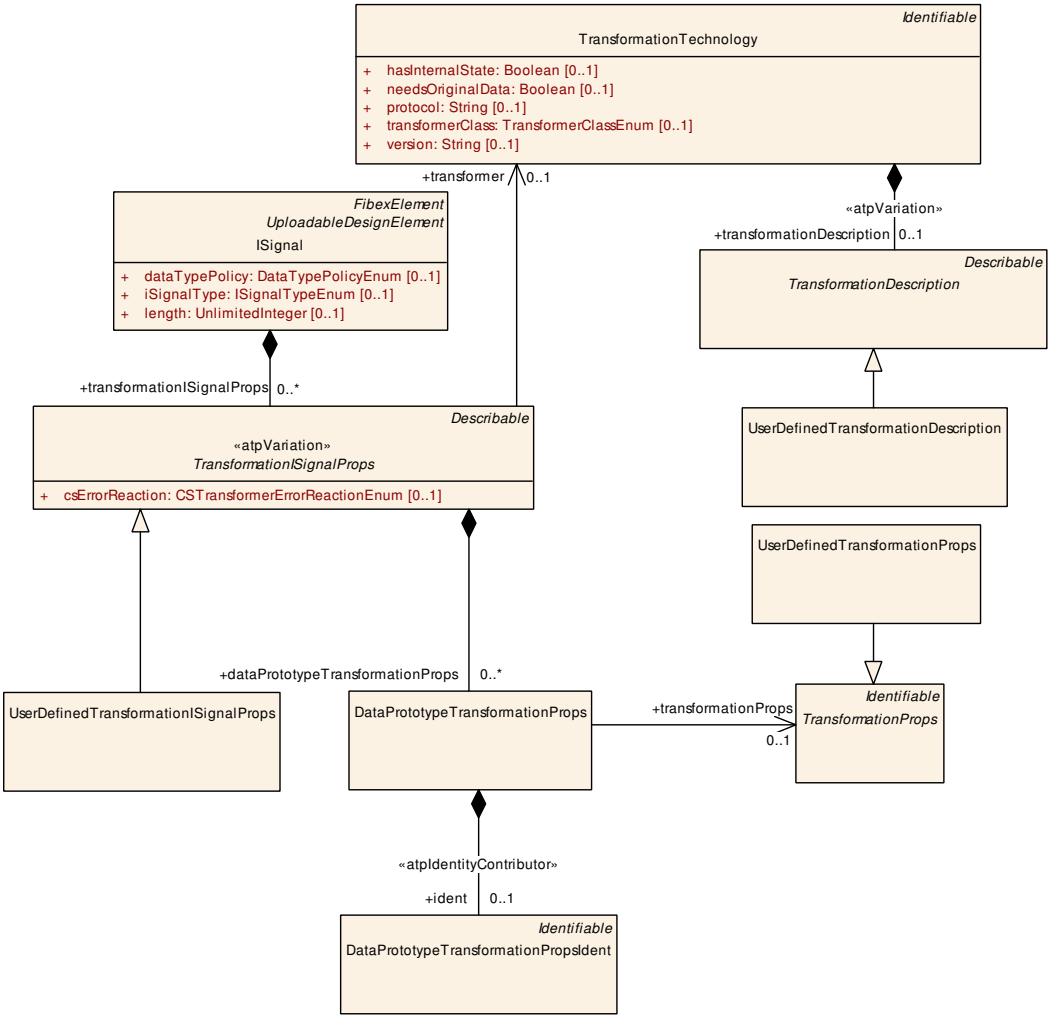

Figure 7.23: User Defined Transformation configuration

Please note that all these UserDefined classes are Identifiable or Describable and therefore are able to describe special data (sdg) which is not represented by the standard model.

Class	«atpVariation» **UserDefinedTransformationISignalProps**			
Note	The UserDefinedTransformationISignalProps is used to specify ISignal specific configuration properties for custom transformers.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 7.32: UserDefinedTransformationISignalProps

Class	**UserDefinedTransformationProps**			
Note	The class UserDefinedTransformationProps specifies specific configuration properties of a user defined serializer.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TransformationProps*			
Aggregated by	TransformationPropsSet.transformationProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 7.33: UserDefinedTransformationProps

7.3.7 Support for TLV Encoding

AUTOSAR supports the usage of the so-called *Tag-Length-Value* (TLV) encoding. The following sub-sections explain the details and the extent of the support for TLV encoding.

7.3.7.1 Assignment of TLV Data Ids

[TPS_SYST_05016] Assignment of TLV data ids ⌈The assignment of TLV data ids is done in the context of the specification of `SOMEIPTransformationISignalProps`, namely by means of the attribute `SOMEIPTransformationISignalProps.tlvDataIdDefinition.tlvDataIdDefinition.id`.⌋

This approach takes benefit from the fact that the `TlvDataIdDefinition` is able to create references to relevant model elements.

The assignment of the TLV data id is therefore done by creating such a reference and assigning a TLV data id to it by means of the attribute `TlvDataIdDefinition.id`.

Please note that the assignment of TLV data ids is compulsory for an entire data structure that has at least one optional member. In a nutshell, this conclusion (that is also backed by [PRS_SOMEIP_00230], see [18]) is the motivation for the existence of [constr_1641] and [constr_1642].

Please note further that the assignment of TLV data ids is not restricted to data structures with optional members. There is also a use case to support sending the elements of a specific data structure in arbitrary order even if none of the elements is considered optional.

Moreover, TLV data ids can also be assigned to arguments of a ClientServerOperation. Using TLV data ids for arguments supports that arguments can be sent in arbitrary order and that new arguments can be added at arbitrary positions during the evolution of the interface. Note that optional arguments are not supported.

[TPS_SYST_02378] Optional method arguments ⌈AUTOSAR Classic platform does not support the existence of optional method arguments.⌋

The reason for the restriction in [TPS_SYST_02378] is that the RTE does not have an API to handle optional method arguments.



Figure 7.24: Definition of data ids for the TLV encoding inside a SOME/IP message

To sum it up: the usage of TLV data ids and optional members are two different features. Optional members require the usage of TLV data ids, but TLV data ids can also be used without having optional members.

Class	**TlvDataIdDefinitionSet**			
Note	This meta-class acts as a container of TlvDataIdDefinitions to be used in a given context **Tags:** atp.recommendedPackage=TlvDataDefinitionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
tlvDataId Definition	TlvDataIdDefinition	*	aggr	This aggregation represents the collection of TlvDataTid Definitions aggregated by the TlvDataIdDefinitionSet **Tags:** atp.Splitkey=tlvDataIdDefinition, tlvDataId Definition.id

Table 7.34: TlvDataIdDefinitionSet

Class	**TlvDataIdDefinition**			
Note	This meta-class represents the ability to define the tlvDataId.			
Base	*ARObject*			
Aggregated by	TlvDataIdDefinitionSet.tlvDataIdDefinition			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
id	PositiveInteger	0..1	attr	This attribute represents the definition of the value of the TlvDataId **Stereotypes:** atpIdentityContributor
tlvArgument	ArgumentDataPrototype	0..1	ref	This reference assigns a tlvDataId to a given argument of a ClientServerOperation. **Tags:** atp.Splitkey=tlvArgument
tlv Implementation DataType Element	AbstractImplementation DataTypeElement	0..1	ref	This reference associates the definition of a TLV data id with a given AbstractImplementationDataTypeElement. **Tags:** atp.Splitkey=tlvImplementationDataTypeElement
tlvRecord Element	ApplicationRecord Element	0..1	ref	This reference associates the definition of a TLV data id with a given ApplicationRecordElement. **Tags:** atp.Splitkey=tlvRecordElement

Table 7.35: TlvDataIdDefinition

[constr_9288] Existence of `TlvDataIdDefinition.id`

Imposition time: CP: `IT_SysDesc`

⌈For each `TlvDataIdDefinition`, the attribute `id` shall exist.⌋

[TPS_SYST_02211] Reference from `SOMEIPTransformationISignalProps` to `TlvDataIdDefinitionSet` ⌈The reference from `SOMEIPTransformationISignalProps` to `TlvDataIdDefinitionSet` means that it is in the hand of the creator of a model to decide whether a global scope should be assumed or whether the definition needs to be customized for a specific case.⌋

[constr_1641] Consistent assignment of TLV data ids to `ApplicationRecordDataType`

Imposition time: CP: `IT_SysDesc`

⌈For every `ApplicationRecordDataType` where direct members set the attribute `ApplicationRecordElement.isOptional` to the value `True` references to all direct members of this `ApplicationRecordDataType` shall be created on the basis of the definition of `TlvDataIdDefinition`.⌋

[constr_1642] Consistent assignment of TLV data ids to `ImplementationDataType` or `ImplementationDataTypeElement`

Imposition time: CP: `IT_SysDesc`

⌈For every `ImplementationDataType` or `ImplementationDataTypeElement` of `category STRUCTURE` where direct members set the attribute `ImplementationDataTypeElement.isOptional` to the value `True` references to all direct members of this `ImplementationDataType` resp `ImplementationDataTypeElement` shall be created on the basis of the definition of `TlvDataIdDefinition`.⌋

The definition of a TlvDataIdDefinition that refers to an eligible model element is not limited to scenarios where optional elements are defined. It is also possible to define TlvDataIdDefinition for arbitrary methods or data structures.

A typical use case could be to prepare the argument list or sub-elements for future extensions. However, if one argument or sub-element is referenced then it is necessary to define references from TlvDataIdDefinitions to all other arguments or sub-elements as well.

[constr_5111] Existence of references TlvDataIdDefinition.tlvArgument, TlvDataIdDefinition.tlvRecordElement, and TlvDataIdDefinition.tlvImplementationDataTypeElement

Imposition time: CP: IT_SysDesc

⌈For each TlvDataIdDefinition, only one out of the following references shall exist:

- reference to ArgumentDataPrototype in the role tlvArgument
- reference to ApplicationRecordElement in the role tlvRecordElement
- reference to ImplementationDataTypeElement in the role tlvImplementationDataTypeElement.

⌋

[constr_1643] Completeness of the existence of a set of TlvDataIdDefinition.tlvArguments

Imposition time: CP: IT_SysDesc

⌈If the reference TlvDataIdDefinition.tlvArguments exists for one argument of a given ClientServerOperation then further TlvDataIdDefinition.tlvArguments shall exist for all arguments of the given ClientServerOperation and all affected TlvDataIdDefinitions shall be referenced by the same SOMEIPTransformationISignalProps via TlvDataIdDefinitionSet.⌋

[constr_1644] Completeness of the existence of a set of TlvDataIdDefinition.tlvRecordElements

Imposition time: CP: IT_SysDesc

⌈If the reference TlvDataIdDefinition.tlvRecordElement exists for one element of a given ApplicationRecordDataType then further TlvDataIdDefinition.tlvRecordElement shall exist for all elements of the given ApplicationRecordDataType and all affected TlvDataIdDefinitions shall be referenced by the same SOMEIPTransformationISignalProps via TlvDataIdDefinitionSet.⌋

[constr_1645] Completeness of the existence of a set of TlvDataIdDefinition.tlvImplementationDataTypeElements

Imposition time: CP: IT_SysDesc

⌈Completeness of the existence of a set of TlvDataIdDefinition.tlvImplementationDataTypeElements If the reference TlvDataIdDefinition.tlvImplementationDataTypeElement exists for one subElement of a given ImplementationDataType or ImplementationDataTypeElement then further TlvDataIdDefinition.tlvImplementationDataTypeElement shall exist for all subElements of the given ImplementationDataType or ImplementationDataTypeElement and all affected TlvDataIdDefinitions shall be referenced by the same SOMEIPTransformationISignalProps via TlvDataIdDefinitionSet.⌋

The definition of a TlvDataIdDefinition.id has the purpose to provide means to unambiguously identify the argument or sub-element. For this purpose, the value of the id needs to be unique in the respective context.

[constr_1646] Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ArgumentDataPrototype

Imposition time: CP: IT_SysDesc

⌈For all TlvDataIdDefinition that are referencing ArgumentDataPrototypes of a given ClientServerOperation in the role tlvArgument the attribute TlvDataIdDefinition.id shall exist and have a unique value in the context of respective arguments of the enclosing ClientServerOperation where attribute direction is set to the value in/inout or out/inout.

Note: an argument where attribute direction is set to the value in may have the same data id as an argument where attribute direction is set to the value out since the two are transferred in separate messages.⌋

[constr_1647] Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ApplicationRecordElement

Imposition time: CP: IT_SysDesc

⌈For all TlvDataIdDefinition that are referencing ApplicationRecordElements of a given ApplicationDataType in the role tlvRecordElement the attribute TlvDataIdDefinition.id shall exist and have a unique value in the context of respective enclosing ApplicationRecordDataType.⌋

[constr_1648] Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ImplementationDataTypeElement

Imposition time: CP: IT_SysDesc

⌈For all TlvDataIdDefinition that are referencing ImplementationDataTypeElements of a given ImplementationDataType/ImplementationDataTypeElement in the role tlvImplementationDataTypeElement the attribute TlvDataIdDefinition.id shall exist and have a unique value in the context of respective enclosing ImplementationDataType or ImplementationDataTypeElement.⌋

Obviously, it is necessary to avoid ambiguity with respect to the definition of TLV data ids. Each model element that can be assigned such an id shall only be assigned one id.

[constr_1649] TlvDataIdDefinition referencing ArgumentDataPrototype

Imposition time: CP: IT_SysDesc

⌈Each ArgumentDataPrototype shall be referenced at most once in the role tlvArgument in the context of the same SOMEIPTransformationISignalProps.⌋

[constr_1650] TlvDataIdDefinition referencing ApplicationRecordElement

Imposition time: CP: IT_SysDesc

⌈Each ApplicationRecordElement shall be referenced at most once in the role tlvRecordElement in the context of the same SOMEIPTransformationISignalProps.⌋

[constr_1651] TlvDataIdDefinition referencing ImplementationDataTypeElement

Imposition time: CP: IT_SysDesc

⌈Each ImplementationDataTypeElement shall be referenced at most once in the role tlvImplementationDataTypeElement in the context of the same SOMEIPTransformationISignalProps.⌋

As depicted in Figure 7.24, the meta-model supports the TlvDataIdDefinition to refer both to an ApplicationRecordElement as well as an ImplementationDataTypeElement.

In a typical case either the one or the other reference will be used and there is intentionally no constraint to explicitly use both references in a concrete model.

It would mean a significant markup in real-world AUTOSAR models to explicitly require that TlvDataIdDefinitions shall exist that assign concrete ids to both a given ApplicationRecordDataType as well as the mapped ImplementationDataType.

However, scenarios are conceivable that the assignment of TLV data ids may be done based on ApplicationDataType plus networkRepresentationProps on one end of the communication and based on ImplementationDataType on the other end.

In this case, a constraint to keep TLV data ids in sync between ApplicationDataType and ImplementationDataType would not even be helpful because either side might not know about the actual data type used as the basis of the creation of the Transformer at the other end.

Nevertheless, if both an ApplicationDataType and the mapped ImplementationDataType are annotated with TLV data Ids within the same model then the associated values shall obviously be identical for corresponding sub-elements.

7.3.7.2 Assignment of TLV Wire Type

The TLV encoding supports the definition of a so-called wire type that controls how the information about the length of length fields shall be interpreted. The meaning of specific settings of the wire type is defined in [18].

[TPS_SYST_05017] Definition of the applicable wire type attribute SOMEIPTransformationISignalProps.isDynamicLengthFieldSize shall be used to define the applicable wire type ⌈If the value of attribute SOMEIPTransformationISignalProps.isDynamicLengthFieldSize is set to True then **wire type 5-7** shall be used.

If the value of attribute SOMEIPTransformationISignalProps.isDynamicLengthFieldSize does not exist or is set to False then **wire type 4** shall be used.⌋

[constr_1652] Definition of static length fields sizes in case of TLV usage

Imposition time: CP: IT_SysDesc

⌈If TlvDataIdDefinitions are defined for a SOMEIPTransformationISignalProps, the attributes sizeOfArrayLengthFields, sizeOfStructLengthFields, sizeOfStringLengthFields and sizeOfUnionLengthFields shall be greater than 0.⌋

Rationale for the existence of [constr_1652]: The TLV serialization requires the usage of length fields. If wire type 4 is used the length field size shall be statically configured.

If wire types 5-7 (dynamic length field size) are used the static configuration of the length field size shall also be present since not all length fields are preceded by a tag, e.g. structures contained in an array or the top-level struct contained in a SOME/IP event. Not using length fields here would result in ambiguities.

[constr_1653] Identical values for length fields sizes in case of TLV usage

Imposition time: CP: IT_SysDesc

⌈If TlvDataIdDefinitions are defined for a SOMEIPTransformationISignalProps, the attributes sizeOfArrayLengthFields, sizeOfStructLengthFields, sizeOfStringLengthFields and sizeOfUnionLengthFields shall have an identical value.⌋

Rationale for the existence of [constr_1653]: if an unknown member or argument is encountered the deserializer cannot determine the actual datatype of the member/argument when wire type 4 is used.

[constr_1654] No definition of length field sizes on DataPrototype level in case of TLV usage

Imposition time: CP: IT_SysDesc

⌈If TlvDataIdDefinitions are defined for a SOMEIPTransformationISignalProps, the attributes sizeOfArrayLengthFields, sizeOfStructLengthFields and sizeOfUnionLengthFields shall not be defined on DataPrototype level but only on ISignal level.⌋

Rationale for the existence of [constr_1654]: if an unknown member or argument is encountered the deserializer needs to know the size of the length field when wire type 4 is used. The easiest way is that the size of the length field is then only defined at the top-level element.

8 Gateways

A gateway is a function within an EcuInstance that performs as a FrameMapping, IPduMapping or ISignalMapping function between two or more CommunicationClusters.

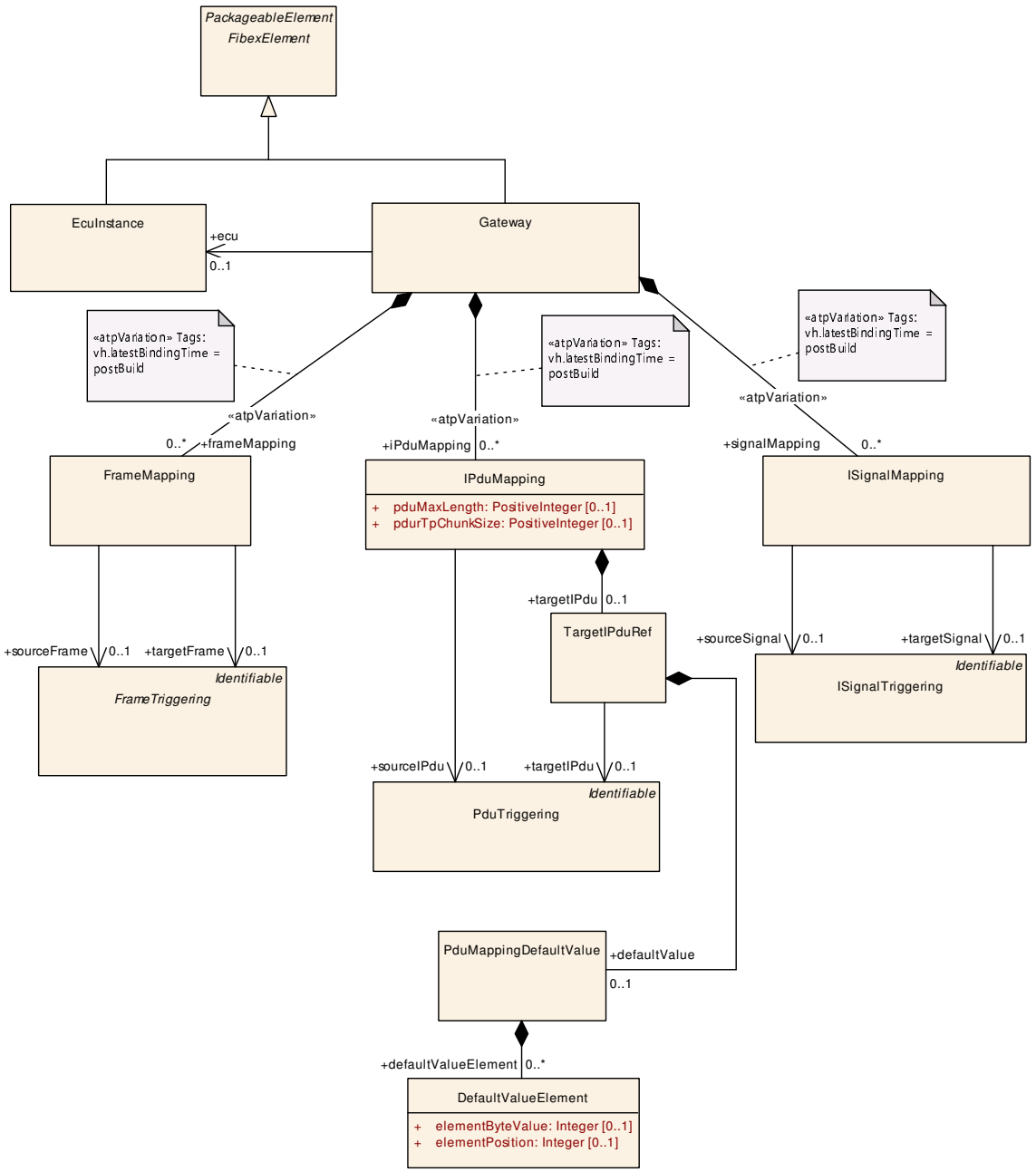

Figure 8.1: Communication Overview (Fibex4Multiplatform: Gateway)

Figure 8.1 shows the meta-model for the Gateway description in the System Template.

Class	**Gateway**			
Note	A gateway is an ECU that is connected to two or more clusters (channels, but not redundant), and performs a frame, Pdu or signal mapping between them. **Tags:** atp.recommendedPackage=Gateways			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecu	EcuInstance	0..1	ref	Reference to one ECU instance that implements the gateway. **Tags:** atp.Splitkey=ecu
frameMapping	FrameMapping	*	aggr	Frame Gateway: The entire source frame is mapped as it is onto the target frame (what in general is only possible inside of a common platform). In this case source and target frame should be the identical object. atpVariation: If frames are variable in clusters, the gateway frame mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=frameMapping, frameMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
iPduMapping	IPduMapping	*	aggr	IPdu Gateway: Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. atpVariation: If PDUs are variable in clusters, the gateway PDU mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=iPduMapping, iPduMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
signalMapping	ISignalMapping	*	aggr	Signal Gateway: Arranges those signals that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. atpVariation: If signals are variable in clusters, the gateway signal mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=signalMapping, signalMapping.variation Point.shortLabel vh.latestBindingTime=postBuild

Table 8.1: Gateway

[constr_9291] Existence of `Gateway.ecu`

Imposition time: CP: `IT_SysDesc`

⌈For each `Gateway`, the reference to `EcuInstance` in the role `ecu` shall exist.⌋

8.1 Frame Mapping

[TPS_SYST_03130] Frame Mapping support

Status: DRAFT

⌈The FrameMapping arranges those FrameTriggerings that are transferred by the Gateway from one PhysicalChannel to the other in pairs and defines the mapping between them. Each pair consists of a sourceFrame and a targetFrame referencing to a FrameTriggering.⌋

The FrameMapping supports the routing of frames between identical bus systems.

[constr_3792] FrameMapping between identical bus systems

Status: DRAFT
Imposition time: CP: IT_SysDesc

⌈The FrameTriggerings referenced in the role sourceFrame and targetFrame shall be owned by PhysicalChannels which in turn are owned by CommunicationClusters of the same kind.⌋

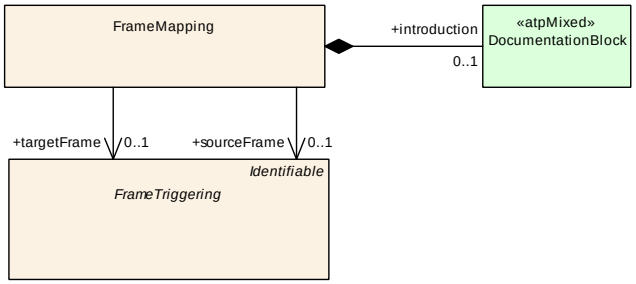

Figure 8.2: Frame Mapping (Fibex4Multiplatform: FrameMapping)

Class	**FrameMapping**			
Note	Maps the source frame to the target frame.			
Base	*ARObject*			
Aggregated by	Gateway.frameMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the frame mapping. **Tags:** atp.Splitkey=introduction
sourceFrame	FrameTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceFrame
targetFrame	FrameTriggering	0..1	ref	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetFrame

Table 8.2: FrameMapping

[constr_9289] Existence of `FrameMapping.sourceFrame`

Imposition time: CP: `IT_SysDesc`

⌈For each `FrameMapping`, the reference to `FrameTriggering` in the role `sourceFrame` shall exist.⌋

[constr_9290] Existence of `FrameMapping.targetFrame`

Imposition time: CP: `IT_SysDesc`

⌈For each `FrameMapping`, the reference to `FrameTriggering` in the role `targetFrame` shall exist.⌋

8.2 IPdu Mapping

[TPS_SYST_01117] Pdu Gateway support ⌈The IPduMapping arranges those IPdus that are transferred by the Gateway from one PhysicalChannel to the other (or the same) PhysicalChannel in pairs and defines the mapping between them. Each pair consist of a sourceIPdu and a targetIPdu referencing to a PduTriggering.⌋

For FlexRay: If a Pdu is gatewayed to more than one PhysicalChannel of the same CommunicationCluster, all of this gateway relationships shall be specified. Therefore, all affected PduTriggerings shall be referenced in the gateway mappings.

[TPS_SYST_01118] Support of Multicast Pdu routing ⌈The 1:n multicast routing is supported with the definition of several IPduMappings where the sourceIPdu refers to the same PduTriggering.⌋

[TPS_SYST_02143] Support of Multisource Pdu routing ⌈The n:1 routing is supported with the definition of several IPduMappings where the targetIPdu refers to the same PduTriggering.⌋

Please note that in case of n:1 routing by a local module (e.g. COM, Dcm) it shall be enforced at run-time that *at most one* routing path is active (i.e., enabled via PduR_EnableRouting()). In case of n:1 routing by a pure gateway routing (either TP or IF) all routing paths can be active at run time.

[TPS_SYST_02207] Routing on the fly ⌈If routing on the fly is not possible in case:

- there is more than one destination routing path or
- the routing path uses the Interface (IF) API instead of Transport Protocol (TP) API

then IPduMapping.pdurTpChunkSize will be ignored in the Ecu configuration.⌋

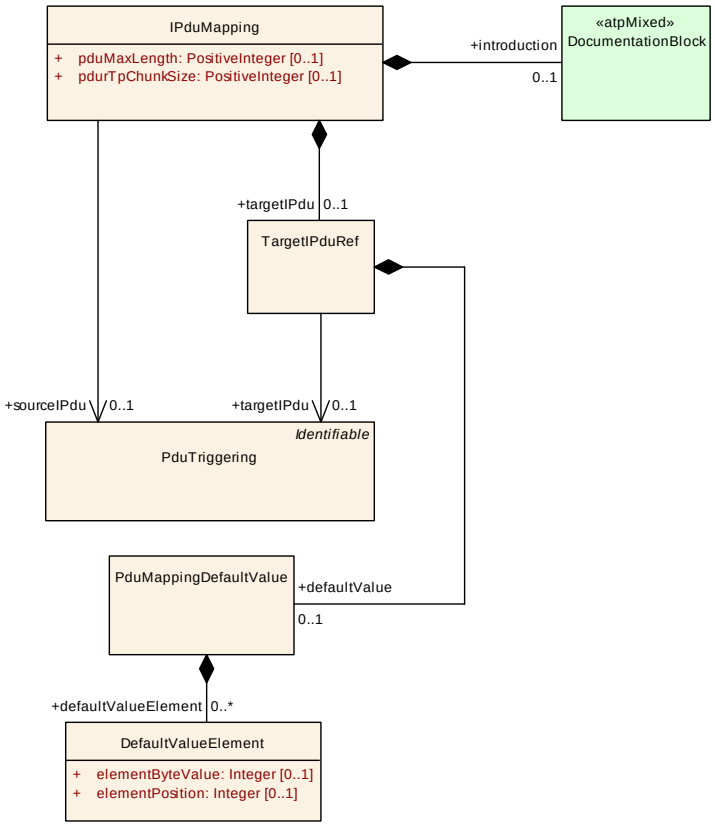

Figure 8.3: I-Pdu Mapping (Fibex4Multiplatform: IPduMapping)

Class	**IPduMapping**			
Note	Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.			
Base	*ARObject*			
Aggregated by	Gateway.iPduMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the IPdu mapping. **Tags:** atp.Splitkey=introduction
pduMaxLength	PositiveInteger	0..1	attr	Define the maximum length in bytes which limits the length of the Pdu during gateway operation if the runtime length of the received Pdu exceeds this limit.
pdurTpChunk Size	PositiveInteger	0..1	attr	Optionally defines the to be configured Pdu Router Tp ChunkSize for this routing relation.
sourceIPdu	PduTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceIPdu
targetIPdu	TargetIPduRef	0..1	aggr	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetIPdu

Table 8.3: IPduMapping

[constr_9292] Existence of IPduMapping.sourceIPdu

Imposition time: CP: IT_SysDesc

⌈For each IPduMapping, the reference to PduTriggering in the role sourceIPdu shall exist.⌋

[constr_9293] Existence of IPduMapping.targetIPdu

Imposition time: CP: IT_SysDesc

⌈Each IPduMapping shall aggregate a TargetIPduRef in the role targetIPdu.⌋

Class	**TargetIPduRef**			
Note	Target destination of the referencing mapping.			
Base	*ARObject*			
Aggregated by	IPduMapping.targetIPdu			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultValue	PduMappingDefaultValue	0..1	aggr	If no I-Pdu has been received a default value will be distributed. **Tags:** atp.Splitkey=defaultValue
targetIPdu	PduTriggering	0..1	ref	IPdu Reference **Tags:** atp.Splitkey=targetIPdu

Table 8.4: TargetIPduRef

[constr_9294] Existence of TargetIPduRef.targetIPdu

Imposition time: CP: IT_SysDesc

⌈For each TargetIPduRef, the reference to PduTriggering in the role target-IPdu shall exist.⌋

Class	**PduMappingDefaultValue**			
Note	Default Value which will be distributed if no I-Pdu has been received since last sending.			
Base	*ARObject*			
Aggregated by	TargetIPduRef.defaultValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
defaultValue Element	DefaultValueElement	*	aggr	The default value consists of a number of elements. Each default value element is represented by the element and the position in an array. **Tags:** atp.Splitkey=defaultValueElement

Table 8.5: PduMappingDefaultValue

[constr_9295] Existence of PduMappingDefaultValue.defaultValueElement

Imposition time: CP: IT_SysDesc

⌈For each PduMappingDefaultValue, at least one DefaultValueElement shall be aggregated by PduMappingDefaultValue in the role defaultValueElement.⌋

Class	**DefaultValueElement**			
Note	The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.			
Base	*ARObject*			
Aggregated by	PduMappingDefaultValue.defaultValueElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
elementByte Value	Integer	0..1	attr	The integer value of a freely defined data byte.
elementPosition	Integer	0..1	attr	This attribute specifies the byte position of the element within the default value

Table 8.6: DefaultValueElement

[constr_9296] Existence of `DefaultValueElement.elementPosition`

Imposition time: CP: `IT_SysDesc`

⌈For each `DefaultValueElement`, the attribute `elementPosition` shall exist.⌋

[constr_9297] Existence of `DefaultValueElement.elementByteValue`

Imposition time: CP: `IT_SysDesc`

⌈For each `DefaultValueElement`, the attribute `elementByteValue` shall exist.⌋

8.2.1 Examples of IPdu Gateway configurations

In figure 8.4 an `ISignalIPdu` is received on a Can network: the *PduTriggeringCan* has a reference to an `IPduPort` for this `EcuInstance` with `communicationDirection` = `in`.

For illustration purposes just one `ISignal` is shown as part of the `ISignalIPdu`. This `ISignal` has also an `ISignalTriggering` defined. However, this `ISignalTriggering` has no reference to an `ISignalPort` for this `EcuInstance`. Therefore, there is no interaction of the payload `ISignal` of the `ISignalIPdu` defined in the scope of this `EcuInstance`.

The *PduTriggeringCan* is referenced by an `IPduMapping` in the role `sourceIPdu`. This defines a gateway operation where *PduTriggeringFr* is the target of the gateway (for simplicity the illustration of the `TargetIPduRef` has been left out). The *PduTriggeringFr* is a `PduTriggering` which also refers to the same `ISignalIPdu` as the *PduTriggeringCan*.

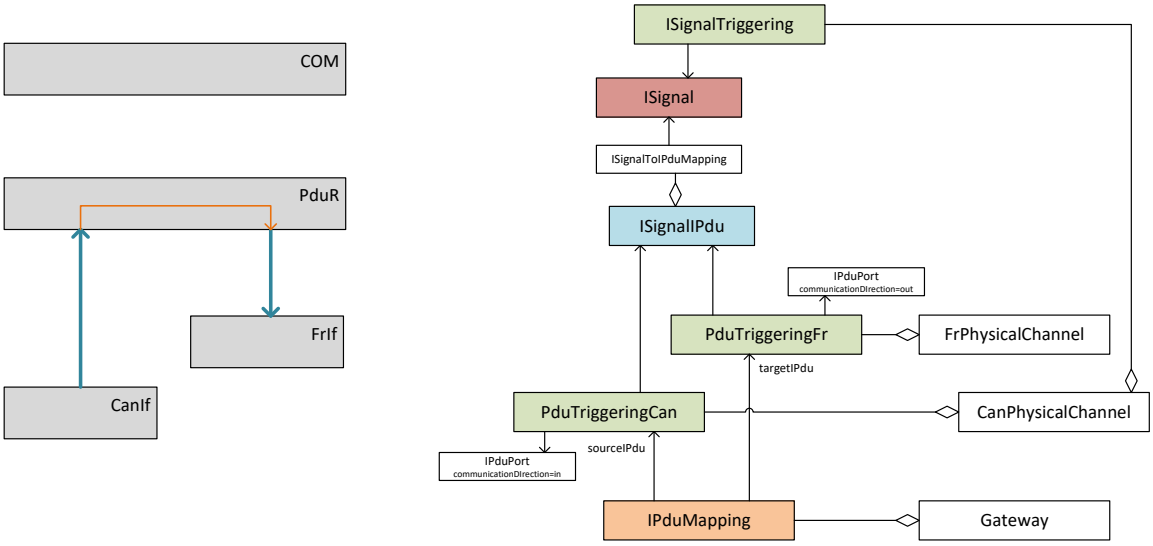

Figure 8.4: Simple IPdu Gateway definition

An alternative modeling of `ISignalIPdu`s and `ISignal`s is to have their definition be network focused and maintained in `ARPackage`s representing one vehicle network each.

In such a setup there would be an `ISignalIPdu` and `ISignal` definition for each network. In the example of Figure 8.5 this means that there is an `ISignalIPdu` and `ISignal` for *Can* and another `ISignalIPdu` and `ISignal` for *Fr*.

For the sake of simplicity the following examples do not clutter the figures with such duplicate `ISignalIPdu`s and `ISignal`s, in real projects however such a duplication is quite often used.

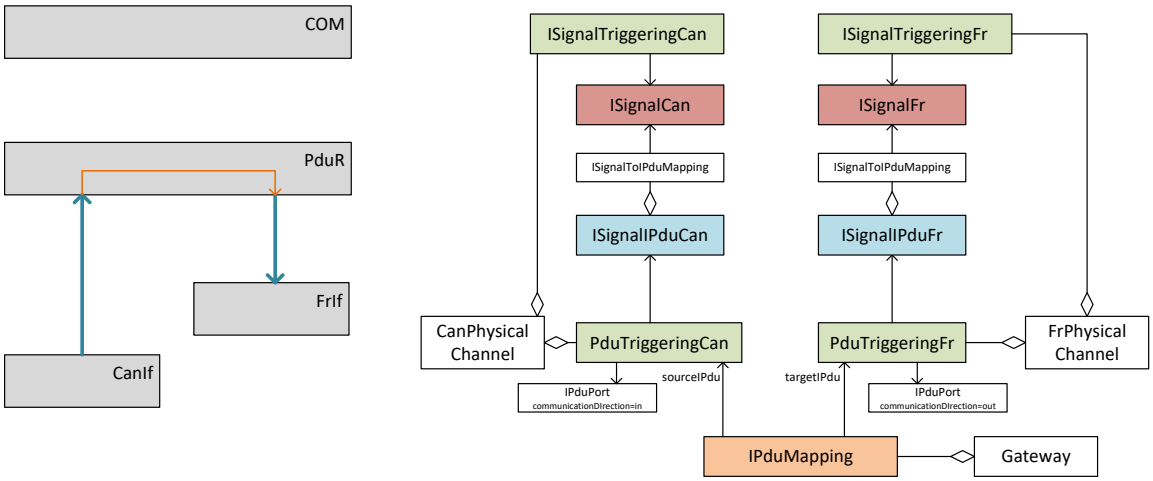

Figure 8.5: Simple IPdu Gateway with duplicated `ISignalIPdu` definition

In figure 8.6 the same gateway operation as in figure 8.4 is defined using the IPduMapping. Additionally, the ISignalTriggering has a reference to an ISignalPort with communicationDirection = in for this EcuInstance.

The example configuration in figure 8.6 not only performs the gateway operation of the ISignalIPdu, but also receives the ISignalIPdu locally and processes the payload in the Com module. In this scenario the Pdu router module performs a "fan-out" where the received *PduTriggeringCan* (coming from the CanIf module) is forwarded to the Com module and forwarded to the FrIf module. More details on Pdu router fan-out can be found in chapter 6.11.2.1.

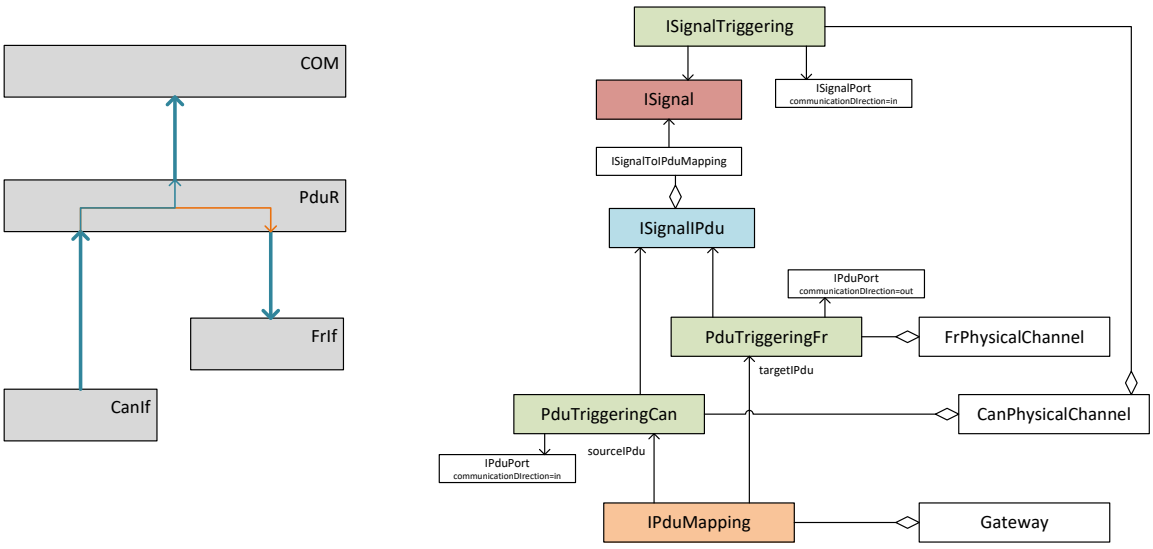

Figure 8.6: IPdu Gateway definition with local reception

One further refinement is shown in figure 8.7 where the received *PduTriggeringCan* is processed locally by the Com module and additionally defined as contained part of a ContainerIPdu. In this example the gateway operation is not immediately visible, because the received *PduTriggeringCan* is first gatewayed into the *PduTriggeringSig* which is then taken as the ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering for the ContainerIPdu.

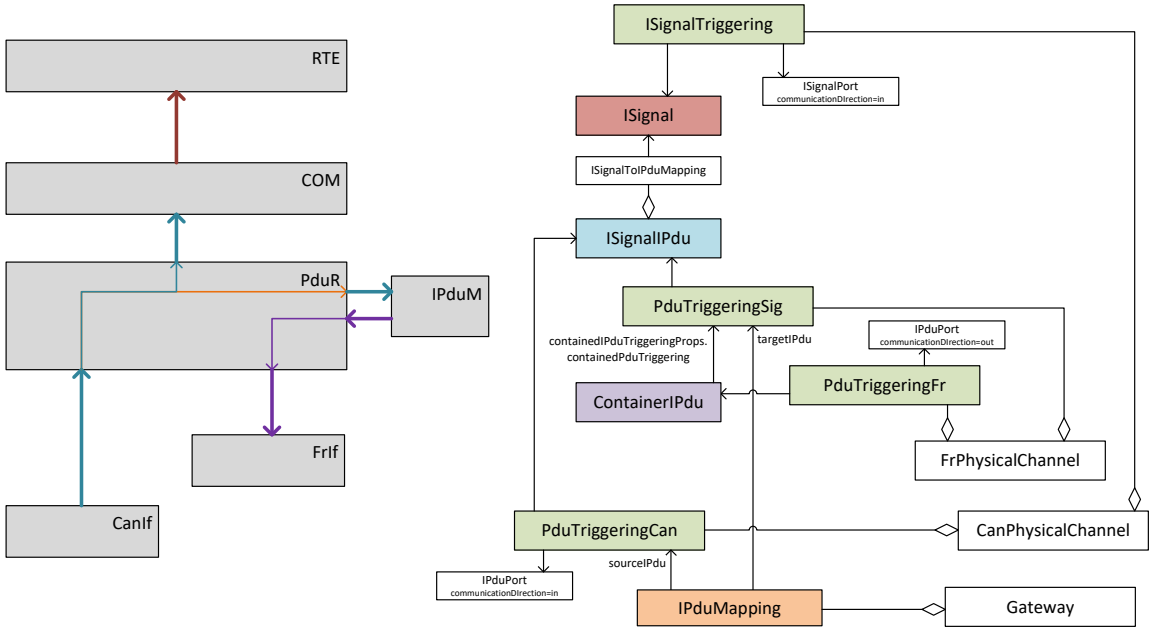

Figure 8.7: IPdu Gateway definition with local reception and container

8.2.2 Usage of `IPduMapping.pduMaxLength`

The Pdu gateway can be configured to use `IPduMapping.pduMaxLength` as a specific value per gateway operation. This value (if defined) will be used for the length configuration of the involved `Pdu`s in the Ecu Configuration of the COM-Stack. There is no direct 1:1 correspondence of `IPduMapping.pduMaxLength` in EcuC, it is reflected in the value of EcuC `PduLength` only.

The rational for the existence of `IPduMapping.pduMaxLength` is that in the system template the length of a `Pdu` is defined at the `Pdu.length` attribute. The `Pdu` can be used in the definition of several `PduTriggering`s. The use-case is to have the possibility to define different ECU Configuration `PduLength`s for each routing operation (`IPduMapping`).

[TPS_SYST_02310] Pdu routing with `IPduMapping.pduMaxLength` ⌈The attribute `IPduMapping.pduMaxLength` defines a maximum length which shall be forwarded to the destination module by the PduR, if the runtime length of the actually received `Pdu` exceeds `IPduMapping.pduMaxLength`.⌋

If the attribute `IPduMapping.pduMaxLength` is defined the value will be derived into the EcuC `PduLength` configuration parameter.

- If the runtime length of the received `Pdu` does not exceed `PduLength` (now configured to the value of `IPduMapping.pduMaxLength`) then the PduR will forward the runtime length of the received `Pdu` to the corresponding `targetIPdu`.

- If the runtime length of the received Pdu does exceed PduLength (now configured to the value of IPduMapping.pduMaxLength) then the PduR will forward the PduLength of the received Pdu to the corresponding targetIPdu.

If the attribute IPduMapping.pduMaxLength is not defined then the Pdu.length is used to derive the PduLength configuration parameter. This corresponds to the normal routing operation.

[constr_5166] Existence of IPduMapping.pduMaxLength

Imposition time: CP: IT_SysDesc

⌈If several IPduMappings refer to the same PduTriggering in IPduMapping.sourceIPdu, then all of these IPduMappings shall provide either no IPduMapping.pduMaxLength value, or the same IPduMapping.pduMaxLength value.⌋

[TPS_SYST_02311] IPduMapping.pduMaxLength relying on the environment length configuration ⌈IPduMapping.pduMaxLength shall not exceed the available amount of free payload bytes in the surrounding Pdus or frames of routedPdu, where routedPdu is the sourceIPdu or the targetIPdu, e.g. if routedPdu is contained in a ContainerIPdu with (ContainerIPdu.headerType = shortHeader), then IPduMapping.pduMaxLength shall not exceed (ContainerIPdu.length - 4) (4 being the byte length of the short HeaderId/Length field)⌋

Example: Pdu_A is transmitted on CAN_1 network in a CAN-FD frame and forwarded by a gateway as part of a ContainerIPdu to CAN_2 network.

Pdu_A in the transmitting ECU_A (CAN_1) has the following configuration:
Pdu.length = 9 bytes.

The routing path to route Pdu_A as Contained IPdu to CAN_2 has the following configuration:

- IPduMapping.pduMaxLength = 60 byte
- IPduMapping.sourceIPdu = PduTriggering referring to PDU_A
- IPduMapping.targetIPdu = refers to PduTriggering referring to PDU_B (PDU_B as part of a ContainerIPdu)

The length information in the system description are:

- sourceIPdu.length = 9 byte
- targetIPdu.length = 60 byte (due to pduMaxLength = 60 byte).

The gateway has no data length check configured.

Because this is an IPduMapping with pduMaxLength defined, the specific upstream mapping for the EcuC PduLength applies:

In case IPduMapping is used:
1:1 (sourceIPdu:targetIPdu) routing: When the SysTPduToPduTriggeringRef

PduTriggering is referenced by an IPduMapping in the role sourceIPdu or targetIPdu, respectively, and that IPduMapping has a pduMaxLength defined then IPduMapping.pduMaxLength shall be used as PduLength for the derived PduRSrcPdu and PduRDestPdu, respectively.

The PduLength in Ecu Configuraton of the received and sent Pdu is configured to 60 byte.

Thus both, the receiving and the sending path in the gateway Ecu Configuration are prepared to handle a Pdu with 60 byte.

- ECU_A transmits PDU_A within a CAN-FD frame with a length of actually 12 byte (9 byte payload + 3 byte padding).
- ECU_B (gateway) receives the CAN-L-PDU and CanIf forwards 12 byte to the upper layer as received length.
- PduR forwards 12 byte (as the received length is smaller than IPduMapping.pduMaxLength of 60 byte)
- IpduM packs the received 12 byte payload in the container.

Update scenario: Communication matrix of ECU_A is updated and payload of PDU_A is extended to 60 Byte. ECU_B (gateway) is not updated.

- ECU_A transmits the extended PDU_A within a CAN-L-PDU with a length of 64 byte (60 byte payload + 4 byte padding).
- ECU_B (gateway) receives the CAN-L-PDU, CanIf forwards 64 byte to PduR.
- PduR forwards 60 byte (as the received length is bigger than IPduMapping.pduMaxLength of 60 byte) as received length to the IpduM.
- IPduM packs the shortened 60 byte Pdu to the configured ContainerIPdu.

[constr_5235] Maximum Frame.frameLength of the used bus protocol shall not be exceeded

Imposition time: CP: IT_SysDesc

⌈The Pdu.length used for an IPdu and the IPduMapping.pduMaxLength used for a targetIPdu shall not exceed the limitation of the maximum Frame.frameLength of the used bus protocol (e.g. CAN2.0 max. Frame.frameLength == 8Byte, CAN-FD Frame.frameLength == 64byte).⌋

[constr_5236] Restriction of IPduMapping.pduMaxLength

Imposition time: CP: IT_SysDesc

⌈IPduMapping.pduMaxLength shall be equal or greater than the maximum Pdu.length of sourceIPdu and targetIPdu. For a N:1 routing and 1:N routing, respectively, the maximum Pdu.length of all involved Pdus shall be used to evaluate a proper IPduMapping.pduMaxLength.⌋

8.2.3 Routing and processing of Diagnostics Pdus

An EcuInstance routes a source DcmIPdu to a destination DcmIPdu if there is an IPduMapping in place that is configured according to [TPS_SYST_01117]. The EcuInstance also processes the DcmIPdu locally if the source DcmIPdu is assigned a functional destination address.

8.3 Signal Mapping

[TPS_SYST_01119] Signal Gateway support ⌈The ISignalMapping defines the mapping between ISignals and ISignalGroups that are transferred by the Gateway from one PhysicalChannel to the other (or the same) PhysicalChannel. Each mapping pair consists of a sourceSignal and a targetSignal referencing an ISignalTriggering. Each ISignalTriggering points to either an ISignal or an ISignalGroup. The ISignal refers to the to be routed SystemSignal, the ISignalGroup refers to the to be routed SystemSignalGroup.⌋

[constr_3051] Restriction of ISignalMapping references

Imposition time: CP: IT_SysDesc

⌈If the sourceSignal references an ISignal then the targetSignal shall also reference an ISignal.⌋

[TPS_SYST_01155] Routing of ISignalGroups ⌈If the sourceSignal references an ISignalGroup then the targetSignal can reference either an ISignalGroup or an ISignal.⌋

[constr_3052] Complete ISignalMapping of ISignalGroup signals

Imposition time: CP: IT_SysDesc

⌈If an ISignalMapping to an ISignal that is a member of a ISignalGroup exists then (see [TPS_SYST_01120]) an ISignalMapping to the enclosing ISignalGroup shall exist as well.⌋

[TPS_SYST_02162] Routing of ISignals of ISignalGroups ⌈When performing a signal group routing two approaches are supported for the pairing of the included ISignals:

- implicit mapping: the ISignalMapping points in the sourceSignal role to an ISignalTriggering of an ISignalGroup and no ISignalMappings are defined for the included ISignals. Identical shortNames of ISignal elements identify correlating ISignals between the source and the target in the scope of the ISignalMapping.
- explicit mapping: the ISignalMapping points in the sourceSignal role to an ISignalTriggering of an ISignalGroup and in addition explicitly specified ISignalMappings define which ISignals correlate to each other.

⌋

Please note that SWS_COM [25] does not support the “implicit mapping” of [TPS_SYST_02162]. Thus it is required in the upstream mapping to derive individual ISignalMappings for all the members of a to be routed ISignalGroup.

[TPS_SYST_01120] Precedence of ISignalMappings ⌈If a dedicated ISignal-Mapping for at least one ISignal within an ISignalGroup exists the implicit mapping on the basis of shortNames is no longer applicable for any ISignal within that ISignalGroup.⌋

[TPS_SYST_01121] Support of Mulicast signal routing ⌈The 1:n multicast routing is supported with the definition of several ISignalMappings. See also the COM Signal Gateway fan-out description in [TPS_SYST_01110].⌋

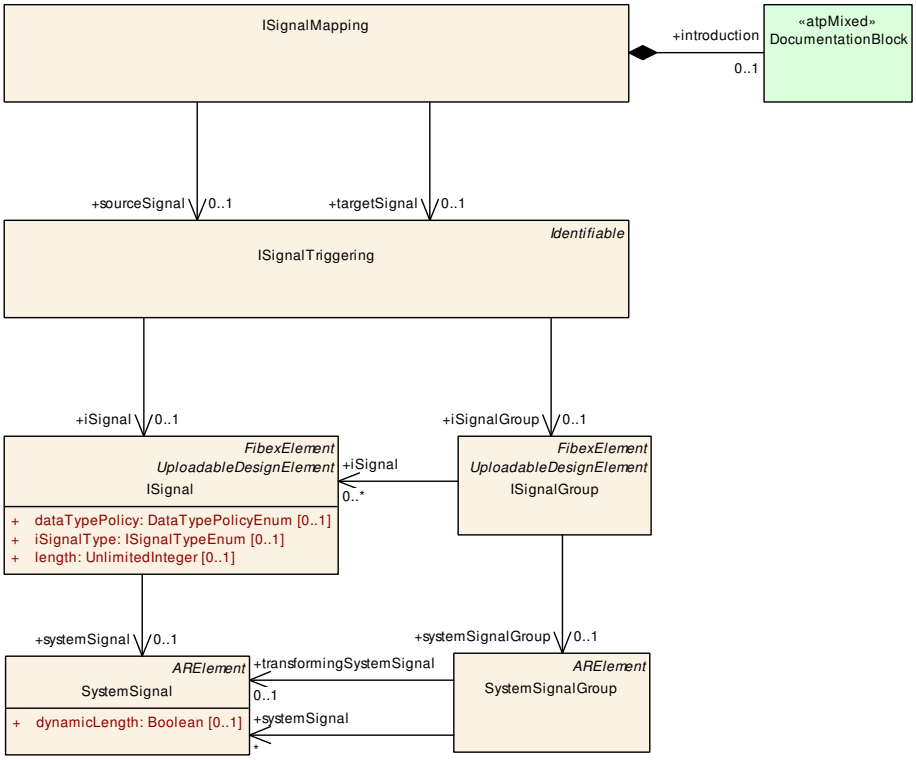

Figure 8.8: Signal Mapping (Fibex4Multiplatform: Signal Mapping)

Class	**ISignalMapping**			
Note	Arranges those signals (or SignalGroups) that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a source and a target referencing to a ISignalTriggering.			
Base	*ARObject*			
Aggregated by	Gateway.signalMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ISignal mapping. **Tags:** atp.Splitkey=introduction
sourceSignal	ISignalTriggering	0..1	ref	Source destination of the referencing mapping. **Tags:** atp.Splitkey=sourceSignal
targetSignal	ISignalTriggering	0..1	ref	Target destination of the referencing mapping. **Tags:** atp.Splitkey=targetSignal

Table 8.7: ISignalMapping

[constr_9298] Existence of `ISignalMapping.sourceSignal`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignalMapping`, the reference to `ISignalTriggering` in the role `sourceSignal` shall exist.⌋

[constr_9299] Existence of `ISignalMapping.targetSignal`

Imposition time: CP: `IT_SysDesc`

⌈For each `ISignalMapping`, the reference to `ISignalTriggering` in the role `targetSignal` shall exist.⌋

8.3.1 Partial Signal Group Mapping

[TPS_SYST_01122] partial routing between `ISignalGroup`s ⌈The `ISignalMapping` supports partial routing between `ISignalGroup`s which have not identical set of `ISignal`s within an `ISignalGroup`.⌋

[constr_3053] Complete `ISignalMapping` of target `ISignalGroup`

Imposition time: CP: `IT_SysDesc`

⌈If an `ISignalGroup` is referenced by a `targetSignal` then [TPS_SYST_02162] applies for each of the contained `ISignal` of that `ISignalGroup`.⌋

Figure 8.9 shows an example for a partial signal group mapping with explicit mappings for the `GroupSignal`s.

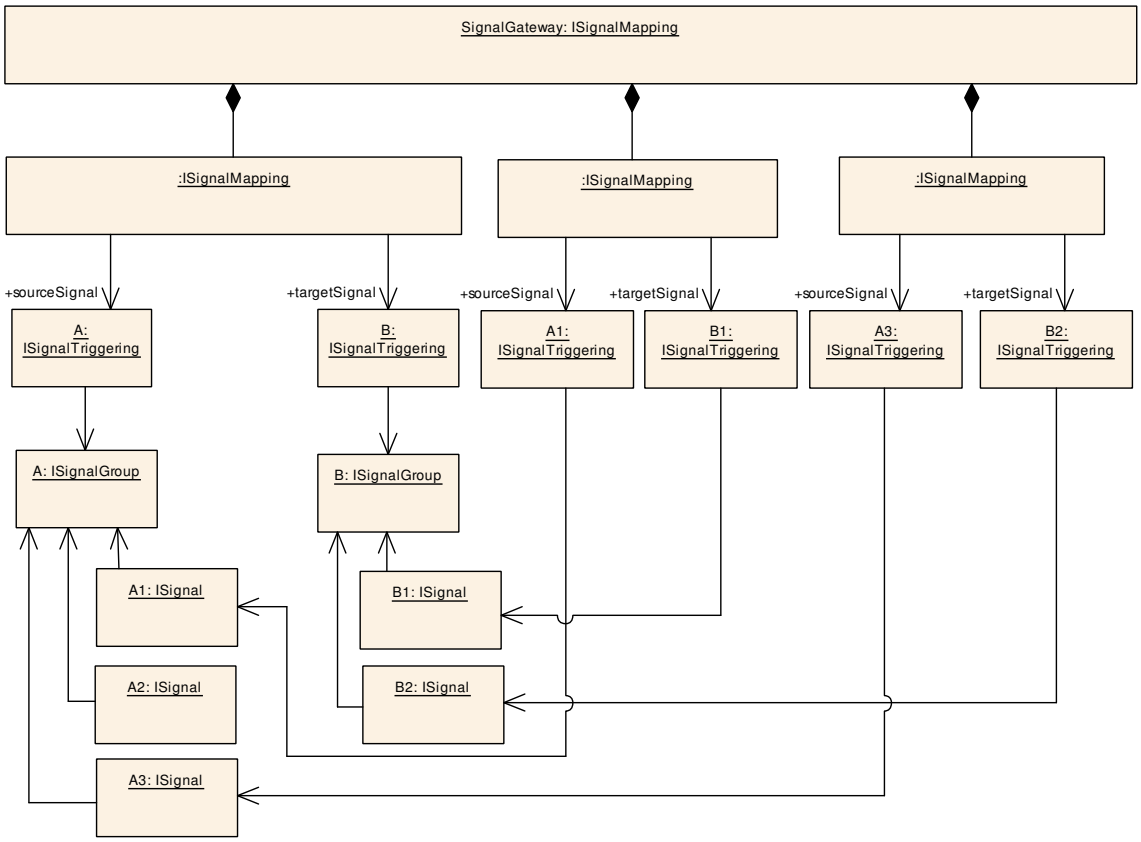

Figure 8.9: Partial Signal Group Mapping Example

9 Global Time Synchronization

9.1 Introduction

This chapter describes the modeling of how a global time synchronization in an AUTOSAR system can be achieved.

[constr_3519] Value of `category` of `GlobalTimeDomain`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `category` of `GlobalTimeDomain` can have the following value:

- SYNCHRONIZED: this time base does not depend on the existence of another time base

⌋

There are several use cases for implementing a system-wide global time in an vehicle:

- In case of an accident it may be necessary to post-mortem analyze whether the vehicle ECUs performed according to specification. This implies that it shall be possible to unambiguously determine the sequence of activities before a crash. This sequence can only be determined if all components in the distributed system depend on a reliable global time basis.
- It may be necessary that several ECUs in the distributed system need to act in concert with respect to the time that a specific activity is executed. A very trivial example for this requirement is the activation of turn indicators in a car. These are rarely connected to a single ECU (which could take care of synchronously flashing the turn indicators) but their synchronized execution is still very essential for the vehicle operation.
- The distribution of several global time bases shall be possible (e.g. a vehicle local time based on the runtime of the car and a GPS-based time).

It is obvious that the distribution of global time within a vehicle requires a system-wide context and therefore, the AUTOSAR System Template defines relevant meta-classes and their relations for this purpose.

Of course, the actual implementation of global time distribution is done in a couple of basic-software modules that need to be configured in the context of integrating a particular ECU. The purpose of the meta-model described in chapter 9 is to support the configuration of these basic-software modules.

The modeling of how the distribution of global time is supposed to work can roughly be distributed into two parts, the discussion of the *big picture* (see 9.2) and the description of the details that eventually will support the configuration of the corresponding basic-software modules. The latter can be found in chapter 9.3.

9.2 The big Picture

The central part of the formalization of global time synchronization is the existence of a *global time domain*, formalized as GlobalTimeDomain.

However, the fragment *global* in *global time domain* primarily stresses the fact that it is supposed to support the distribution of a *global* time rather than implying an information about the scope or visibility of a GlobalTimeDomain[1].

In other words, there is typically more than a single GlobalTimeDomain available in the System.

Please note that the concept of the GlobalTimeDomain roughly corresponds to the existence of a CommunicationCluster, i.e. it takes at least one CommunicationCluster to implement a *global time domain*.

[TPS_SYST_05006] Chaining of GlobalTimeDomains ⌈It is possible to extend the *global time domain* to several CommunicationClusters that are interconnected by means of a Gateway.

In other words, the global time base is routed from one CommunicationCluster to another, whereas the Time Slave resp. Time Slave Port updates its local time base by using the received global time base and takes into account, whether a time base correction has to be considered or not.

There are certainly use-cases for implementing a GlobalTimeDomain that extends to several CommunicationClusters, but in many (if not in the majority of) cases it will be necessary to update the time information for the sake of precision.

In this case, however, two separate GlobalTimeDomains rather than a single GlobalTimeDomain exist. The GlobalTimeDomain relate to each other such that one GlobalTimeDomain refers to the other in the role globalTimeSubDomain.⌋

In order to understand the way how GlobalTimeDomains refer to each other, it is important to understand that the concept of a *global time domain* has an underlying asymmetric approach of how the time information is distributed.

That is, not all participants in the communication of global time information are able and/or entitled to update the time information and send it around for others to consume.

The identification of a global time domain on the system level is primarily done via the GlobalTimeDomain instance.

[TPS_SYST_02103] Semantics of GlobalTimeDomain ⌈GlobalTimeDomain represents a specific time source, e.g. GPS time.⌋

[1] For the intents and purposes of this chapter, always make sure to read **global-time domain** rather than **global time-domain**.

In AUTOSAR Releases prior to R23-11 there was also the restriction that all network representations of the same global time domain had to have the identical GlobalTimeDomain.domainId:

In a chain of GlobalTimeDomain.globalTimeSubDomain the value of the attribute GlobalTimeDomain.domainId had to be identical.

This restriction is no longer applied. If however a network compatibility between pre AUTOSAR R23-11 ECUs and current ECUs has to be achieved, then this restriction still needs to be respected (refer to previous versions of this document for details how to configure GlobalTimeDomain.domainId for compatibility).

The modeling of GlobalTimeDomains and SubDomains describes the propagation of time values of a time source through the networks. Note that these SubDomains might use different GlobalTimeDomain.domainId values on different networks.

[constr_9302] Existence of GlobalTimeDomain.domainId

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈If a GlobalTimeDomain defines a GlobalTimeDomain.globalTimeMaster or GlobalTimeDomain.slave, then the attribute GlobalTimeDomain.domainId shall exist.⌋

[TPS_SYST_05007] separation of roles within a GlobalTimeDomain ⌈Within a single *global time domain*, There is a strict separation of roles into a single *global time master* (formalized by the meta-class GlobalTimeMaster) and a collection of so-called *global time slaves* (formalized by means of the meta-class GlobalTimeSlave).

The role of the GlobalTimeMaster is to provide the global time information and the role of the collection of GlobalTimeSlaves is to consume the information. The chaining of GlobalTimeDomains needs to be understood as the intention to implement the following information flow:

1. from the GlobalTimeMaster of one GlobalTimeDomain to the GlobalTimeSlaves of the same GlobalTimeDomain
2. via the GlobalTimeMaster of the GlobalTimeDomain referenced in the role globalTimeSubDomain to the GlobalTimeSlaves of the globalTimeSubDomain

⌋

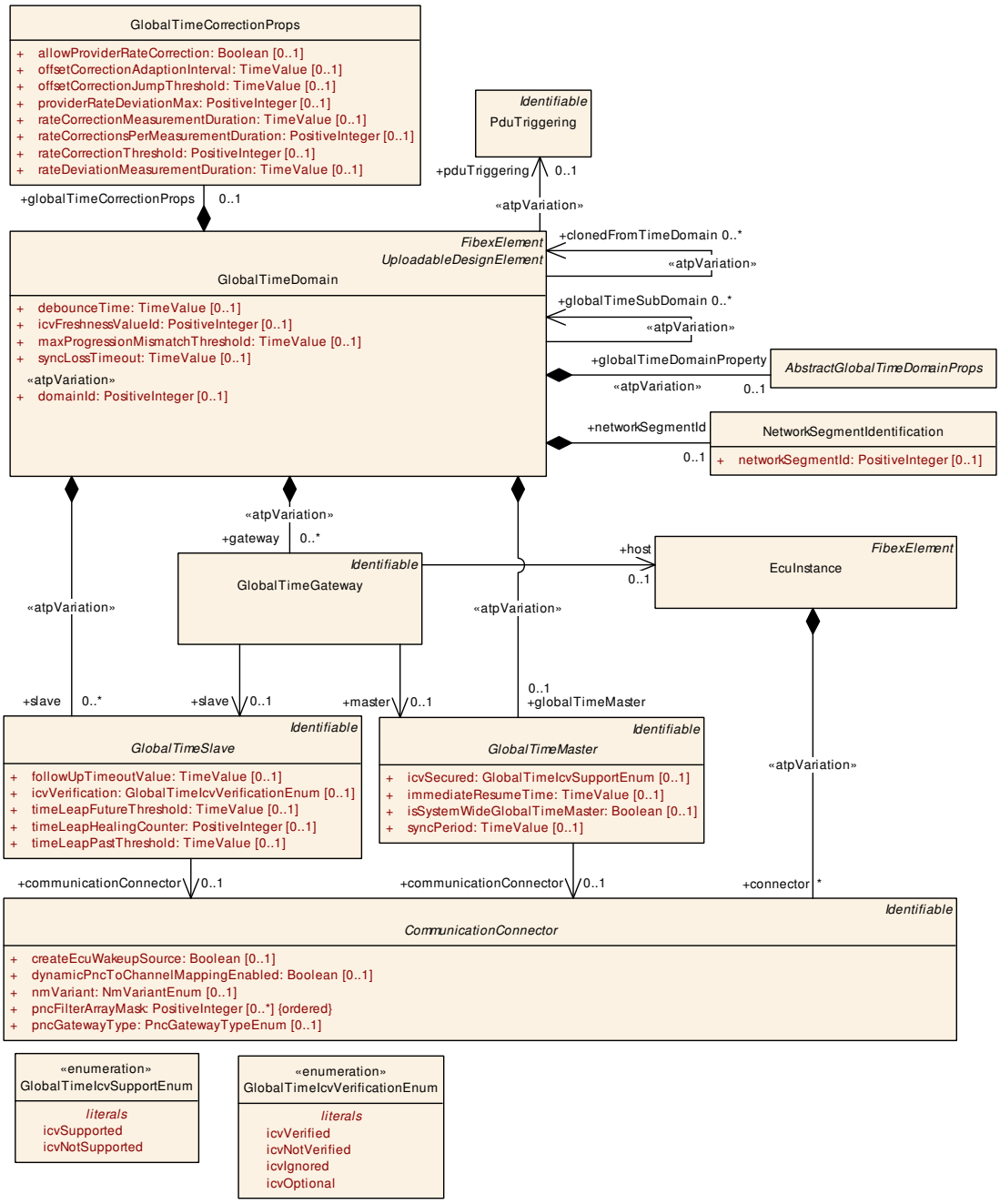

Figure 9.1: Big Picture of AUTOSAR global time synchronization

[TPS_SYST_05008] Semantics of a `GlobalTimeGateway` ⌈In order to achieve the flow of information between a `GlobalTimeSlave` of a given `GlobalTimeDomain` to the `GlobalTimeMaster` of another `GlobalTimeDomain`, it is necessary to establish the existence of a so-called `GlobalTimeGateway`.

In terms of functionality, a `GlobalTimeGateway` complements the functionality of the underlying `Gateway` such that, on top of the mere routing from one `Communication-Cluster` to another, the time information is actively updated in the process of passing it from one `GlobalTimeDomain` to the other.⌋

[TPS_SYST_05009] GlobalTimeDomain.pduTriggering for transmitting global time information ⌈The flow of global time information is unidirectional, i.e. the GlobalTimeSlaves consume the information without providing any form of feedback to the corresponding GlobalTimeMaster.

Thanks to this conceptual detail, there is only the need for **one** dedicated Pdu for the transmission of the actual global time information in the context of one GlobalTimeDomain.

The characteristics of accessing the information contained in this Pdu do make any requirements on the nature of the Pdu. Therefore, it is sufficient and applicable to use the GeneralPurposePdu for this use case.

To make this possible, it is necessary to include the global time use case in the set of standardized values of the attribute GeneralPurposePdu.category. In other words, [constr_3081] applies.⌋

[constr_3261] GlobalTimeDomain.pduTriggering category

Imposition time: CP: IT_SysDesc

⌈The Pdu that is referenced by the PduTriggering that in turn is referenced by GlobalTimeDomain in the role pduTriggering shall be a GeneralPurposePdu of category GLOBAL_TIME.⌋

[TPS_SYST_05010] GlobalTimeDomain.pduTriggering is not required on Ethernet ⌈The Pdu for transmitting global time information is not required on the Ethernet bus. Here, the information is accessed directly from the Ethernet Interfaces, i.e. the hardware already keeps track of the global time.⌋

[constr_1369] CommunicationConnectors shall be attached to the same CommunicationCluster

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈All CommunicationConnectors referenced from GlobalTimeMaster and GlobalTimeSlaves aggregated in one GlobalTimeDomain shall be referenced in the role commConnector by the same PhysicalChannel aggregated by the same CommunicationCluster.⌋

[constr_1370] Consistency of GlobalTimeDomain

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈The GlobalTimeSlave referenced in the role GlobalTimeGateway.slave and the GlobalTimeMaster referenced in the role GlobalTimeGateway.master shall **not** be aggregated by the same GlobalTimeDomain.⌋

The background of [constr_1370] is that the GlobalTimeGateway is supposed to connect two GlobalTimeDomains it is hardly possible that the GlobalTimeGateway.slave and the GlobalTimeMaster can be aggregated by the same GlobalTimeDomain.

[TPS_SYST_05011] Ownership of GlobalTimeGateway ⌈Since the existence of a GlobalTimeGateway is only justified if a GlobalTimeDomain exists that is referenced by a GlobalTimeDomain in the role globalTimeSubDomain it seems appropriate to aggregate the GlobalTimeGateway at the GlobalTimeDomain referenced in the role globalTimeSubDomain.⌋

In other words, the GlobalTimeGateway shall be aggregated at the GlobalTimeDomain that also aggregates the master.

Please note that GlobalTimeDomain.gateway effectively has a 0..1 multiplicity since no more than one globalTimeMaster is allowed per GlobalTimeDomain.

[constr_1371] Consistency of attribute host

Imposition time: CP: IT_SysDesc

⌈Within the context of an aggregating GlobalTimeDomain, the CommunicationConnectors referenced in the role GlobalTimeGateway.master.communicationConnector and GlobalTimeGateway.slave.communicationConnector shall be aggregated by the same EcuInstance that is referenced in the role GlobalTimeGateway.host.⌋

[constr_1372] Consistency of attribute pduTriggering

Imposition time: CP: IT_SysDesc

⌈Within the context of an aggregating GlobalTimeDomain, the pduTriggering shall be owned by PhysicalChannel that is also referencing the CommunicationConnectors referenced in the roles GlobalTimeSlave.communicationConnector and GlobalTimeMaster.communicationConnector.⌋

[TPS_SYST_05013] Semantics of GlobalTimeMaster.isSystemWideGlobalTimeMaster ⌈The attribute GlobalTimeMaster.isSystemWideGlobalTimeMaster indicates whether a given GlobalTimeMaster is considered an independent (i.e. [constr_1373] applies) source of global time information.⌋

[constr_1373] GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE shall not be referenced in the role GlobalTimeGateway.master.⌋

[TPS_SYST_05014] GlobalTimeMaster.isSystemWideGlobalTimeMaster ⌈There is no limitation regarding the number of GlobalTimeMasters that have attribute isSystemWideGlobalTimeMaster set to TRUE. The attribute does not imply that there can only be one GlobalTimeMaster within the context of a System.⌋

[constr_1374] Only fan-out possible for `GlobalTimeGateway`

Imposition time: CP: `IT_SysDesc`

⌈For all `GlobalTimeGateway`s that refer to the same `EcuInstance` the condition applies that no two `GlobalTimeGateway`s shall refer to the same `GlobalTimeMaster`.⌋

In other words, a fan-in of time information such that time information is received from several sources is not supported.

GlobalTime sub domains are associated with specific `PhysicalChannel`s (via the `PduTriggering`s of `GeneralPurposePdu`s with `category` GLOBAL_TIME, see [constr_3081]).

In order to identify the `PhysicalChannel` on a system scope, the `NetworkSegmentIdentification.networkSegmentId` is used.

The `networkSegmentId` is derived into the ECU Configuration of every GlobalTime sub domain participant and is available to the TimeSync modules.

One specific use-case is the identification of dedicated `PhysicalChannel`s for the Time Validation (see [51]), where a central entity receives time validation notifications from network nodes of several GlobalTime subdomains and needs to match the respective notifications according to the `networkSegmentId`.

[constr_3620] `GlobalTimeDomain.networkSegmentId` only applicable to GlobalTime sub domains

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The aggregation `GlobalTimeDomain.networkSegmentId` shall only be defined if the `GlobalTimeDomain` is itself referenced in the role `GlobalTimeDomain.globalTimeSubDomain`.⌋

Rational: There is a GlobalTime sub domain defined for each network the GlobalTime is distributed to, thus only for GlobalTime sub domains a definition of a `networkSegmentId` makes sense.

Note that the `GlobalTimeDomain.networkSegmentId` is currently not taken to the ECU Configuration of the EthTSyn module, as there are other means used for the identification of EthTSyn network segments (see [52]).

[constr_3621] Value range of `GlobalTimeDomain.networkSegmentId`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈If defined, the value of `GlobalTimeDomain.networkSegmentId` shall be in the range 0..255.⌋

It is possible to define that a `GlobalTimeDomain` is secured for the transport using the Integrity Check Value (ICV). This requires a configuration of the properties used for the calculation and verification of the ICV.

The authentication algorithm and its configuration are defined at the GlobalTimeDomain using the reference GlobalTimeDomain.icvSecureComProps. Additionally the attribute GlobalTimeDomain.icvFreshnessValueId is provided to configure the freshness value calculation and verification.

Class	**SecOcSecureComProps**			
Note	Configuration of AUTOSAR SecOC. **Tags:** atp.recommendedPackage=SecureComProps			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *SecureComProps*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
authentication	CryptoServicePrimitive	0..1	ref	This reference defines the authentication or authenticated encryption algorithm used for MAC generation and verification or AEAD encryption and decryption. **Tags:** atp.Splitkey=authentication
authentication BuildAttempts	PositiveInteger	0..1	attr	This attribute defines the additional number of authentication build attempts that are to be carried out when the generation of the authentication information failed for a given message. If zero is set then only one authentication attempt is done.
authentication VerifyAttempts	PositiveInteger	0..1	attr	This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given message. If zero is set than only one authentication attempt is done.
authInfoTx Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Message.
freshnessValue Length	PositiveInteger	0..1	attr	This attribute defines the complete length in bits of the Freshness Value.
freshnessValue TxLength	PositiveInteger	0..1	attr	This attribute defines the length in bits of the Freshness Value to be included in the payload of the secured message.
jobRequirement	SecOcJobRequirement	*	aggr	Collection of cryptographic job requirements. **Tags:** atp.Splitkey=jobRequirement, job Requirement.shortName
use Authenticated Encryption	Boolean	0..1	attr	A Boolean value that indicates if the payload shall be handled as confidential. If it set to TRUE, Authenticated Encryption with Associated Data (AEAD) shall be applied.

Table 9.1: SecOcSecureComProps

Each GlobalTimeMaster of the GlobalTimeDomain can individually define whether the Integrity Check Value (ICV) shall be calculated and transported for that GlobalTimeMaster's network.

[TPS_SYST_03125] Enabling sending of GlobalTimeDomain Integrity Check Value (ICV) ⌈The attribute GlobalTimeMaster.icvSecured setting to GlobalTimeIcvSupportEnum.icvSupported enables the calculation and sending of the ICV on the network that GlobalTimeMaster is responsible for.⌋

Each GlobalTimeSlave of the GlobalTimeDomain can individually define whether the Integrity Check Value (ICV) shall be verified (if received).

[TPS_SYST_03126] Enabling verification of `GlobalTimeDomain` Integrity Check Value (ICV) ⌈The attribute `GlobalTimeSlave.icvVerification` defines the reception behavior of the ICV on the network that `GlobalTimeSlave` is responsible for.⌋

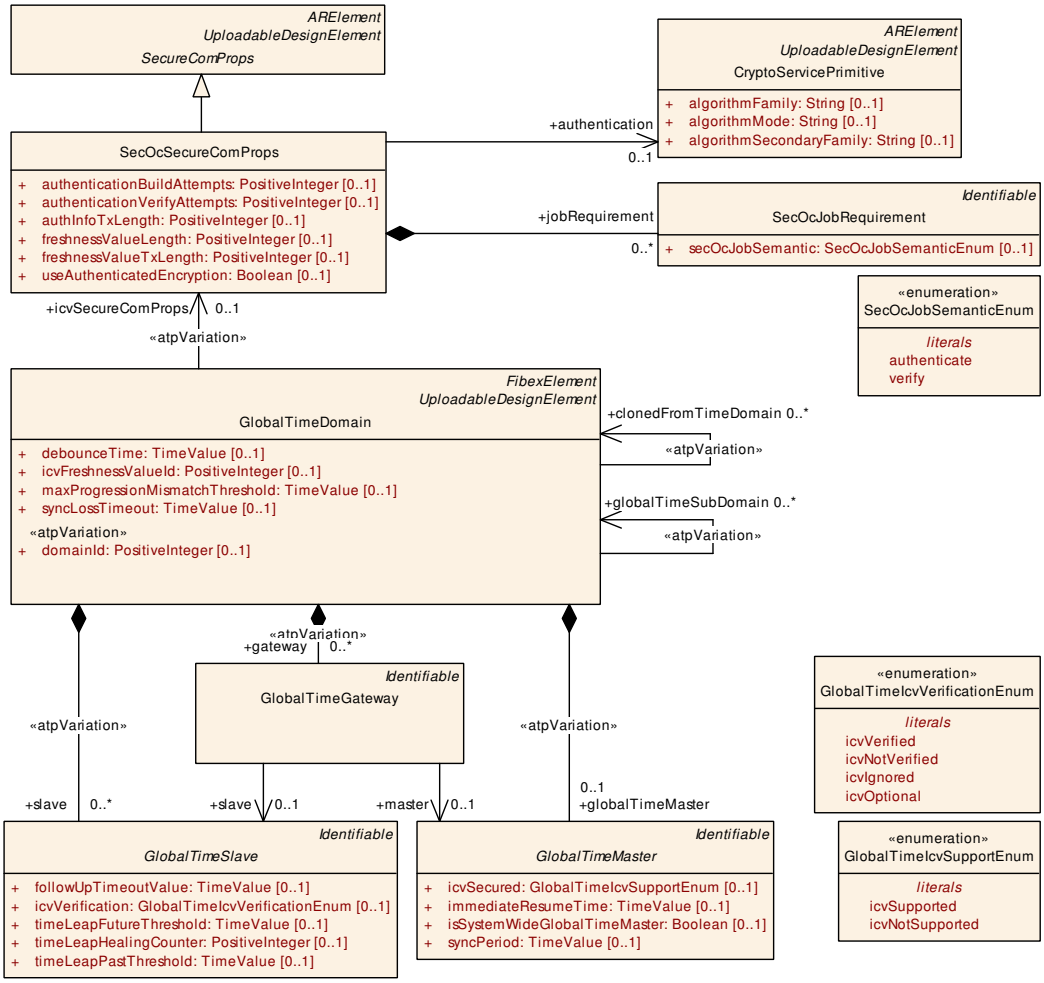

Figure 9.2: Global Time Sync with Integrity Check Value (ICV)

In figure 9.3 an example of a Global Time Sync setup is shown. The *Global time master* ECU creates the *TimeDomain* and provides it to several `globalTimeSubDomain`s. The `GlobalTimeMaster`s for the `globalTimeSubDomain`s take the *TimeDomain* and distribute it to their networks.

The time for the `GlobalTimeMaster`s *TM11* and *TM21* is based on the *TimeDomain* and therefore they have the attribute `isSystemWideGlobalTimeMaster` set to *true*.

The time for the `GlobalTimeMaster`s *TM12* and *TM22* are based on a `GlobalTimeGateway` and therefore they have the attribute `isSystemWideGlobalTimeMaster` set to *false*.

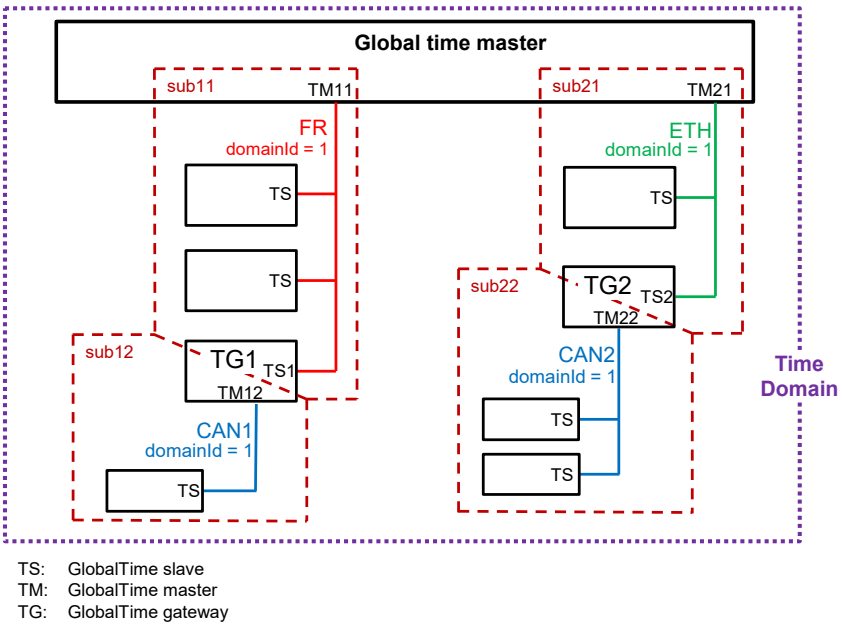

Figure 9.3: Example Global Time Sync topology

A partial outline of the example system description structure is shown in figure 9.4.

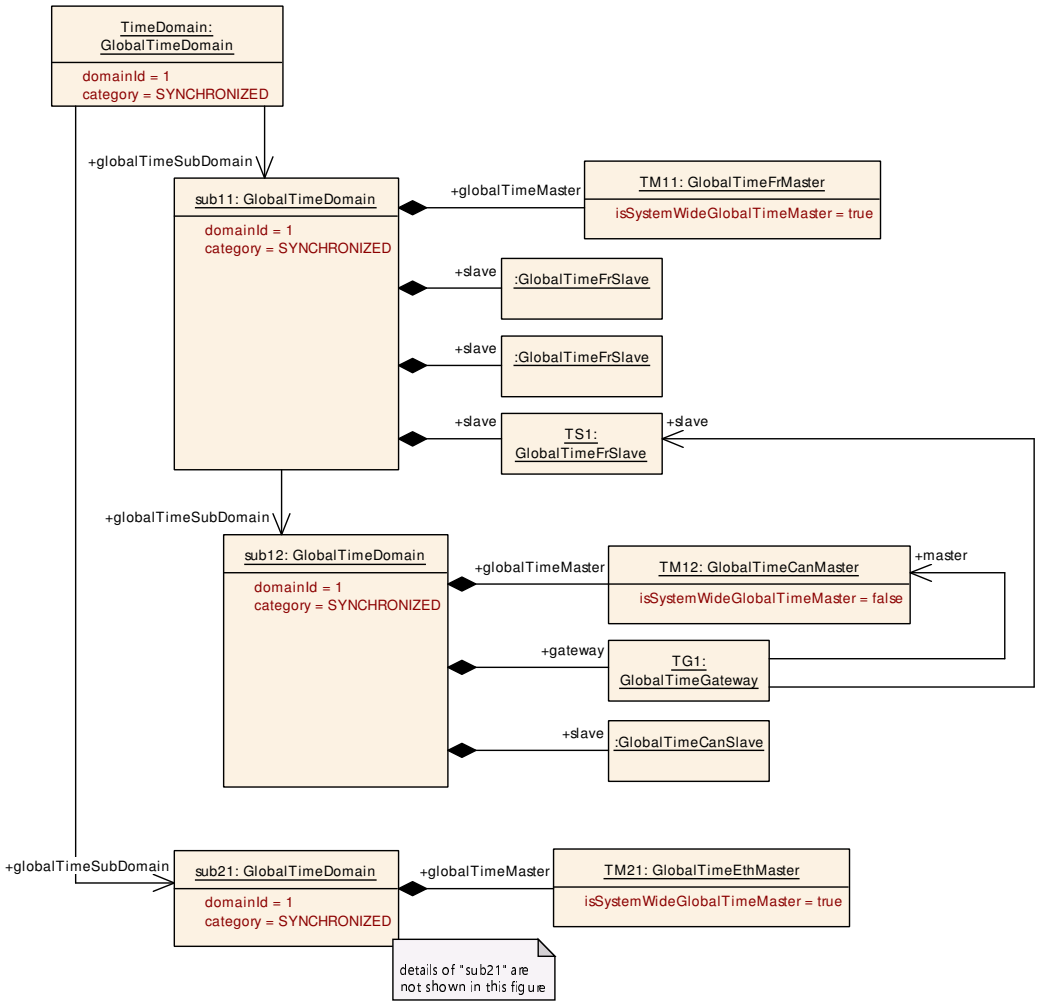

Figure 9.4: System Description of Global Time Sync example

In figure 9.5 an example a Global Time Sync setup is shown where different `GlobalTimeDomain.domainId`s are used for different `GlobalTimeDomain.globalTimeSubDomain`s.

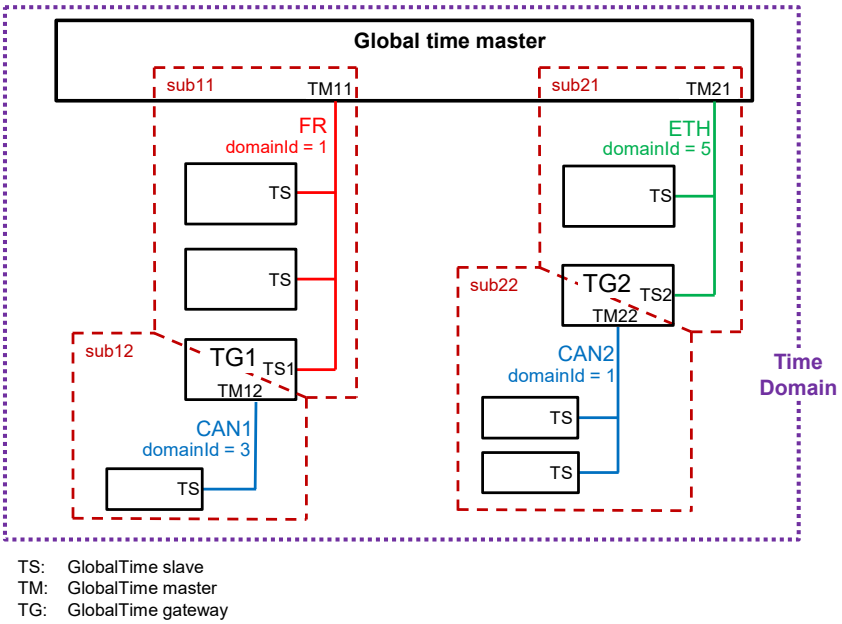

Figure 9.5: Example Global Time Sync topology using different `domainId`s

A partial outline of the example system description structure is shown in figure 9.6.

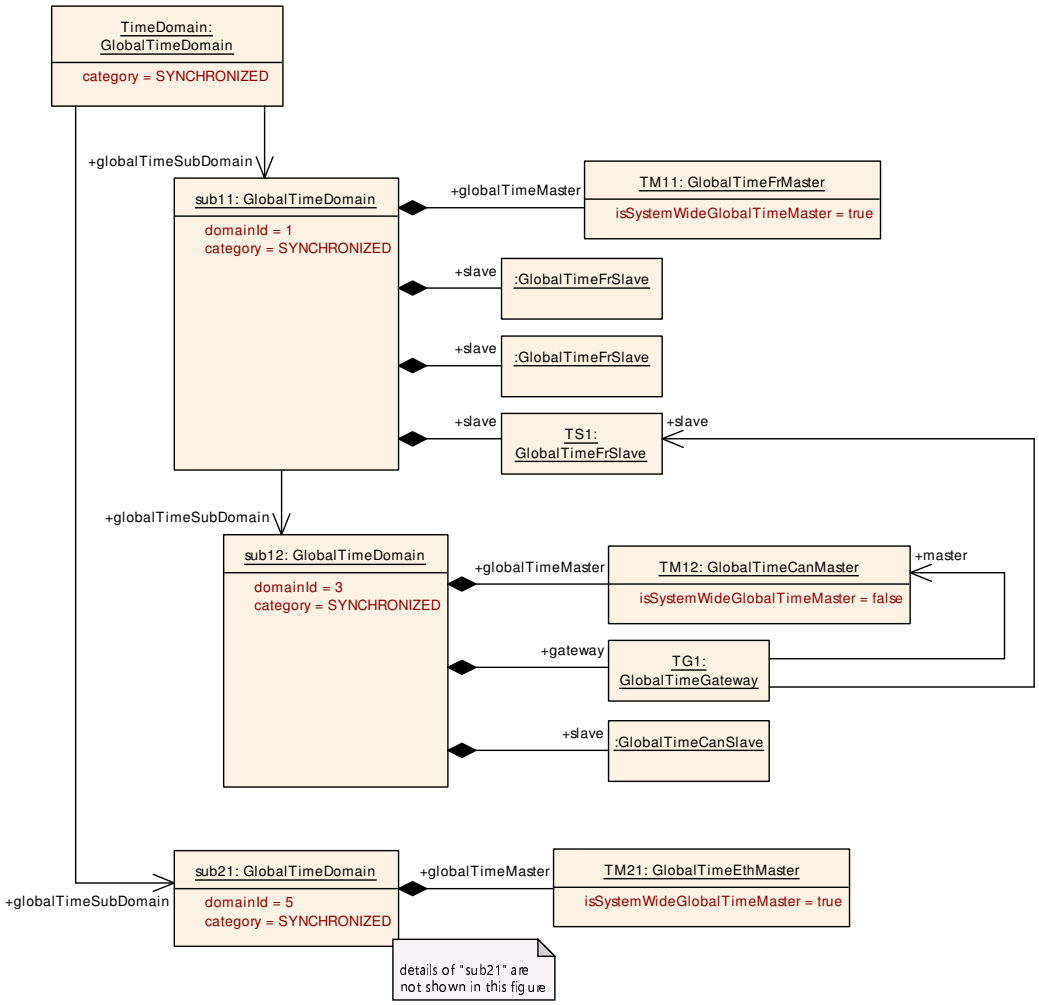

Figure 9.6: System Description of Global Time Sync example

Class	**GlobalTimeDomain**			
Note	This represents the ability to define a global time domain. **Tags:** atp.recommendedPackage=GlobalTimeDomains			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clonedFrom TimeDomain	GlobalTimeDomain	*	ref	Reference to a set of GlobalTimeDomains from which the time information of the reference source GlobalTime Domain is cloned. This allows to model the cloning source of global time information. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clonedFromTimeDomain, clonedFromTime Domain.globalTimeDomain, clonedFromTime Domain.variationPoint.shortLabel vh.latestBindingTime=postBuild
debounceTime	TimeValue	0..1	attr	Defines the minimum amount of time between two time sync messages are transmitted.

△

Class	**GlobalTimeDomain**			
domainId	PositiveInteger	0..1	attr	This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
gateway	GlobalTimeGateway	*	aggr	A GlobalTimeGateway may exist in the context of a GlobalTimeDomain to actively update the global time information as it is routed from one GlobalTimeDomain to another. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=gateway, gateway.shortName, gateway.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTime CorrectionProps	GlobalTimeCorrection Props	0..1	aggr	Defintion of attributes for rate and offset correction. **Tags:** atp.Splitkey=globalTimeCorrectionProps
globalTime Domain Property	AbstractGlobalTime DomainProps	0..1	aggr	Additional properties of the GlobalTimeDomain. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeDomainProperty, globalTime DomainProperty.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTime Master	GlobalTimeMaster	0..1	aggr	This represents the single master of a GlobalTime Domain. A GlobalTimeDomain may have no GlobalTime Domain.master, e.g. when it gets its time from a GPS receiver. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeMaster, globalTimeMaster.short Name, globalTimeMaster.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTimeSub Domain	GlobalTimeDomain	*	ref	By this means it is possible to create a hierarchy of sub Domains where one global time domain can declare one or more other global time domains as its subDomains. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=globalTimeSubDomain, globalTimeSub Domain.globalTimeDomain, globalTimeSub Domain.variationPoint.shortLabel vh.latestBindingTime=postBuild
icvFreshness ValueId	PositiveInteger	0..1	attr	This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.
icvSecureCom Props	SecOcSecureCom Props	0..1	ref	Reference to a SecureComProps definition to be used for the Integrity Check Value (ICV) calculation and verification. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=icvSecureComProps, icvSecureCom Props.secOcSecureComProps, icvSecureCom Props.variationPoint.shortLabel vh.latestBindingTime=postBuild
maxProgression Mismatch Threshold	TimeValue	0..1	attr	This attribute defines the maximum allowed difference between local time and fallback time of the time base in seconds.

▽

△

Class	**GlobalTimeDomain**			
network SegmentId	NetworkSegment Identification	0..1	aggr	Defines the numerical identification of a GlobalTime sub domain. **Tags:** atp.Splitkey=networkSegmentId
pduTriggering	PduTriggering	0..1	ref	This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=pduTriggering, pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
slave	GlobalTimeSlave	*	aggr	This represents the collections of slaves of the Global TimeDomain. A GlobalTimeDomain may have no Global TimeDomain.slaves, e.g. when it propagates its time directly to sub domains. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=slave, slave.shortName, slave.variation Point.shortLabel vh.latestBindingTime=postBuild
syncLoss Timeout	TimeValue	0..1	attr	This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain.

Table 9.2: GlobalTimeDomain

Class	***AbstractGlobalTimeDomainProps*** (abstract)			
Note	This abstract class enables a GlobalTimeDomain to specify additional properties.			
Base	*ARObject*			
Subclasses	CanGlobalTimeDomainProps, EthGlobalTimeDomainProps, FrGlobalTimeDomainProps			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 9.3: AbstractGlobalTimeDomainProps

Class	**NetworkSegmentIdentification**			
Note	This meta-class represents the ability to identify the PhysicalChannel on a system scope in a numerical way. One possible application of this approach is the Time Validation.			
Base	*ARObject*			
Aggregated by	GlobalTimeDomain.networkSegmentId			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
network SegmentId	PositiveInteger	0..1	attr	This attribute represents the numerical identifier of a PhysicalChannel on system level scope.

Table 9.4: NetworkSegmentIdentification

Class	***GlobalTimeMaster*** (abstract)			
Note	This represents the generic concept of a global time master.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	GlobalTimeCanMaster, GlobalTimeEthMaster, GlobalTimeFrMaster, UserDefinedGlobalTimeMaster			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	Communication Connector	0..1	ref	The GlobalTimeMaster is bound to the Communication Connector. **Tags:** atp.Splitkey=communicationConnector
icvSecured	GlobalTimeIcvSupport Enum	0..1	attr	Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages. This Attribute is only used by the AUTOSAR Classic Platform.
immediate ResumeTime	TimeValue	0..1	attr	Defines the minimum time between an "immediate" message and the next periodic message.
isSystemWide GlobalTime Master	Boolean	0..1	attr	If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.
syncPeriod	TimeValue	0..1	attr	This represents the period. Unit: seconds

Table 9.5: GlobalTimeMaster

[constr_9303] Existence of GlobalTimeMaster.communicationConnector

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the reference to CommunicationConnector in the role communicationConnector shall exist.⌋

[constr_9304] Existence of GlobalTimeMaster.isSystemWideGlobalTimeMaster

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the attribute isSystemWideGlobalTimeMaster shall exist.⌋

[constr_9305] Existence of GlobalTimeMaster.syncPeriod

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeMaster, the attribute syncPeriod shall exist.⌋

Class	***GlobalTimeSlave*** (abstract)			
Note	This represents the generic concept of a global time slave.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	GlobalTimeCanSlave, GlobalTimeEthSlave, GlobalTimeFrSlave, UserDefinedGlobalTimeSlave			
Aggregated by	GlobalTimeDomain.slave			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	Communication Connector	0..1	ref	The GlobalTimeSlave is bound to the Communication Connector. **Tags:** atp.Splitkey=communicationConnector

▽

△

Class	***GlobalTimeSlave*** (abstract)			
followUp TimeoutValue	TimeValue	0..1	attr	Rx timeout for the follow-up message.
icvVerification	GlobalTimeIcv VerificationEnum	0..1	attr	Defines how an Integrity Check Value (ICV) shall be handled at the receiver. This Attribute is only used by the AUTOSAR Classic Platform.
timeLeapFuture Threshold	TimeValue	0..1	attr	Defines the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value.
timeLeap HealingCounter	PositiveInteger	0..1	attr	Defines the required number of updates to the Time Base where the time difference to the previous received value has to remain within the bounds of timeLeapFuture Threshold and timeLeapPastThreshold until that Time Base is considered healed.
timeLeapPast Threshold	TimeValue	0..1	attr	Defines the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value.

Table 9.6: GlobalTimeSlave

[constr_9306] Existence of `GlobalTimeSlave.communicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `GlobalTimeSlave`, the reference to `CommunicationConnector` in the role `communicationConnector` shall exist.⌋

Class	**GlobalTimeGateway**			
Note	This represents the ability to define a time gateway for establishing a global time domain over several communication clusters.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.gateway			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
host	EcuInstance	0..1	ref	The GlobalTimeGateway is hosted by the referenced Ecu Instance. **Tags:** atp.Splitkey=host This Attribute is only used by the AUTOSAR Classic Platform.
master	GlobalTimeMaster	0..1	ref	This represents the master of the global time gateway. **Tags:** atp.Splitkey=master
slave	GlobalTimeSlave	0..1	ref	This represents the slave of the GlobalTimeGateway. **Tags:** atp.Splitkey=slave

Table 9.7: GlobalTimeGateway

[constr_9307] Existence of `GlobalTimeGateway.master`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `GlobalTimeGateway`, the reference to `GlobalTimeMaster` in the role `master` shall exist.⌋

[constr_9308] Existence of `GlobalTimeGateway.slave`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `GlobalTimeGateway`, the reference to `GlobalTimeSlave` in the role `slave` shall exist.⌋

[TPS_SYST_02115] Applicability of `GlobalTimeDomain.globalTimeDomainProperty` ⌈The defined properties at `GlobalTimeDomain.globalTimeDomainProperty` may be defined individually per `GlobalTimeDomain`. This allows to define different value sets for each `GlobalTimeDomain` and any of the sub-domains.⌋

[TPS_SYST_02163] Applicability of `syncLossTimeout` ⌈`GlobalTimeDomain.syncLossTimeout` shall be specified for `GlobalTimeDomain`s that have an aggregated slave and for all other cases this attribute is not applicable.⌋

Class	**GlobalTimeCorrectionProps**			
Note	This meta-class defines the attributes for rate and offset correction.			
Base	*ARObject*			
Aggregated by	GlobalTimeDomain.globalTimeCorrectionProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
allowProvider RateCorrection	Boolean	0..1	attr	Defines whether the rate correction value of a Time Base can be set by means of the method setRateCorrection(). false: rate correction cannot be set by method setRate Correction(). true: rate correction can be set by method setRateCorrection().
offsetCorrection AdaptionInterval	TimeValue	0..1	attr	Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation.
offsetCorrection JumpThreshold	TimeValue	0..1	attr	Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump.
providerRate DeviationMax	PositiveInteger	0..1	attr	This attribute describes the maximum allowed absolute value of the rate deviation value [unit: ppm].
rateCorrection Measurement Duration	TimeValue	0..1	attr	Definition of the time span which is used to calculate the rate deviation.
rateCorrections Per Measurement Duration	PositiveInteger	0..1	attr	Defines the number of simultaneous rate measurements to determine the current rate deviation.
rateCorrection Threshold	PositiveInteger	0..1	attr	This attribute defines the threshold for rate correction calculation to determine whether the absolute value of the calculated rate deviation exceeds the accepted range. The value is given in ppm. A value of 0 deactivates the check.
rateDeviation Measurement Duration	TimeValue	0..1	attr	Time span used to calculate the rate deviation. Unit: seconds.

Table 9.8: GlobalTimeCorrectionProps

[TPS_SYST_02436] Attributes of `GlobalTimeCorrectionProps` applicable for `GlobalTimeMaster` ⌈The following attributes of `GlobalTimeCorrectionProps` are applicable to the `GlobalTimeMaster`:

- allowProviderRateCorrection
- providerRateDeviationMax

⌋

[TPS_SYST_02437] Attributes of GlobalTimeCorrectionProps applicable for GlobalTimeSlave ⌈The following attributes of GlobalTimeCorrectionProps are applicable to the GlobalTimeSlave:

- offsetCorrectionAdaptionInterval
- offsetCorrectionJumpThreshold
- rateCorrectionMeasurementDuration
- rateCorrectionsPerMeasurementDuration
- rateCorrectionThreshold
- rateDeviationMeasurementDuration

⌋

9.2.1 Cloning of Global Time Information

The cloning of a synchronized global time information from one GlobalTimeDomain to another GlobalTimeDomain allows to implement a replication of the global time information. The significant difference to a GlobalTimeGateway (see also [TPS_SYST_05008]) is that the cloning of global time information creates a new GlobalTimeDomain that is independent from the source GlobalTimeDomain.

One use-case for time base cloning is feeding back the actually synchronized time information to the network it originates from. This cloned time base may then be used by ECUs to measure the quality of the synchronization, e.g. by comparing the time information received from the original GlobalTimeMaster with the cloned time information.

Another use-case for time base cloning is the reverse synchronization of global time information, e.g. for the purpose of measuring the quality of the synchronization using external development tooling (see also section 9.2.2).

[TPS_SYST_03151] Semantics of GlobalTimeDomain.clonedFromTimeDomain ⌈The reference from a GlobalTimeDomain to another GlobalTimeDomain in the role clonedFromTimeDomain defines a cloning relationship between the two GlobalTimeDomains. The time information of the referenced GlobalTimeDomain is cloned to the referencing GlobalTimeDomain.⌋

In figure 9.7 the system level model of cloning the *TimeDomain1* in the *Cloning ECU* to the *ClonedTimeDomain7* is illustrated. *ECU 1* is the source of the global time information and provides the *TimeDomain1* via the *TM1* to the Ethernet. The *Cloning ECU* takes this time information at *TS12* and clones it to the *ClonedTimeDomain7*. The *ClonedTimeDomain7* is then provided via *TM7* to the same Ethernet network as the original *TimeDomain1*. *ECU 1* is now able to consume the *ClonedTimeDomain7* at *TS7* and compare it with the original time information at *TM1* for the sake of measuring the quality of the synchronization.

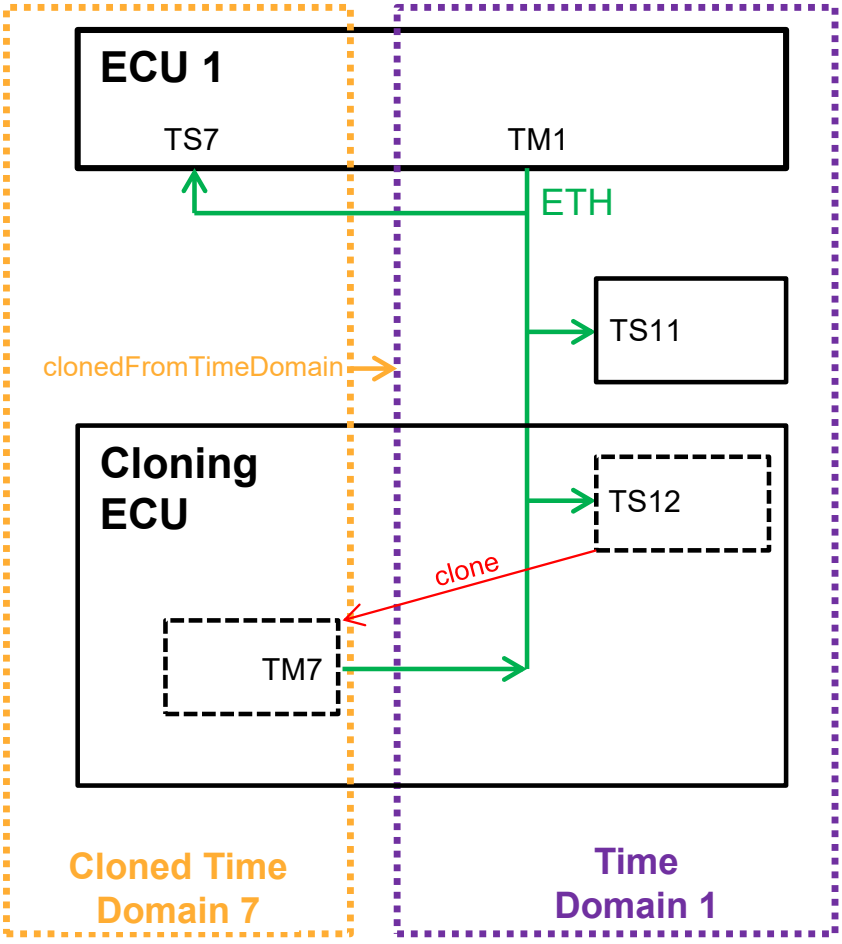

Figure 9.7: Example Global Time cloning topology

In figure 9.8 the modeling of the source `GlobalTimeDomain` and the cloned `GlobalTimeDomain` is shown. The reference from the cloned `GlobalTimeDomain` to the source `GlobalTimeDomain` in the role `clonedFromTimeDomain` defines this cloning relationship.

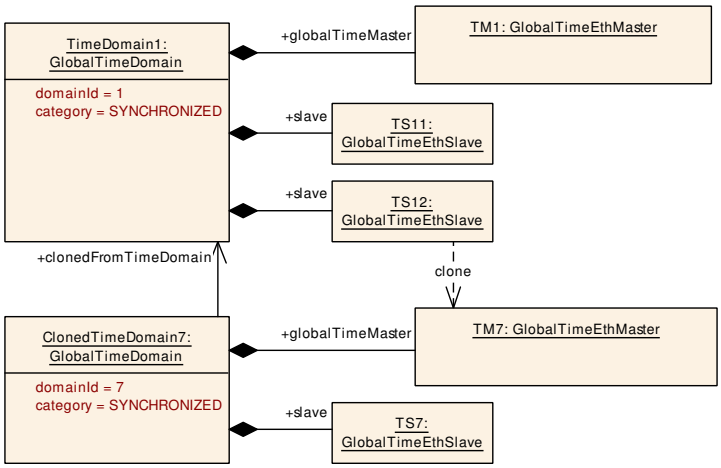

Figure 9.8: System Description of Global Time Cloning example

In figure 9.9 only the model elements are shown which belong to the *Cloning ECU*. From this view it is obvious that the cloning operation shall happen from *TS12* to *TM7*.

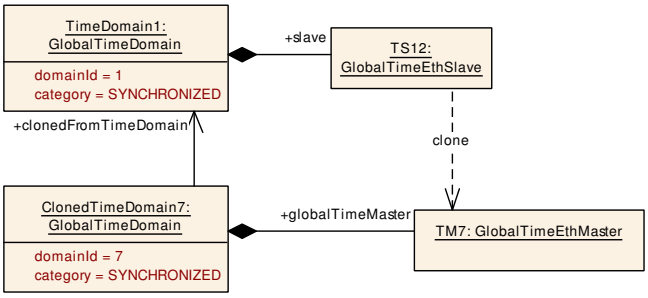

Figure 9.9: System Extract of Global Time Cloning example

9.2.2 Reverse Sync Method

The AUTOSAR_FO_EXP_TimeSensitiveNetworkFeatures [53] mentions several methods how the quality (accuracy) of the global time at a `GlobalTimeSlave` can be measured. These are development features which can be enabled to perform measurements for a specific `EcuInstance`.

The Reverse Sync Method has the special approach that a `GlobalTimeSlave` takes the synchronized global time he synchronizes to and clones this time back to the network using a different `GlobalTimeDomain`, i.e. a different `GlobalTimeDomain.domainId`.

An example model for time base cloning is discussed in section 9.2.1.

This Reverse Sync may be activated for the sole purpose of measurement and debugging. In a production ECU this Reverse Sync will typically be turned off.

The Reverse Sync might use a PhysicalChannel where already one or more GlobalTimeDomains are communicated on and thus, the number of GlobalTimeDomain.domainId may run short. So typically there is one domainId used to handle the Reverse Sync of all connected EcuInstances. The assumption is, that only up to one of the connected EcuInstances will actually turn on the Reverse Sync and thus use the respective domainId.

From the PhysicalChannel there are several GlobalTimeMasters defined which all use the same GlobalTimeDomain.domainId value. As long as only one of these GlobalTimeMasters becomes active at the same time this setup operates properly.

[TPS_SYST_02403] Usage of an identical GlobalTimeDomain.domainId in the GlobalTimeMaster role on one PhysicalChannel ⌈It is a valid configuration to define several GlobalTimeMasters with an identical GlobalTimeDomain.domainId on the same PhysicalChannel. During runtime it shall be guaranteed that only up to one of the GlobalTimeMasters is active at any point in time.⌋

9.3 Detailed Description of Global Time Synchronization

This chapter describes how the concept of *global time synchronization* is applied to various communication bus systems.

Although the characteristics of the supported bus systems differ widely in terms of their communication behavior, the modeling is actually quite similar for all of the supported bus systems.

9.3.1 Time Synchronization over CAN

This chapter described the detailing of how the concept of *global time synchronization* is applied to the CAN bus in particular.

The implementation of *global time synchronization* on the CAN bus is modeled by means of GlobalTimeCanMaster, a concrete subclass of GlobalTimeMaster. A similar approach applies for the GlobalTimeCanSlave, which is derived from GlobalTimeSlave.

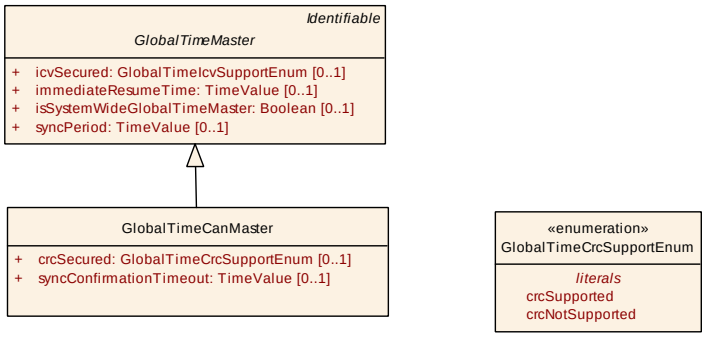

Figure 9.10: Modeling of the GlobalTimeCanMaster

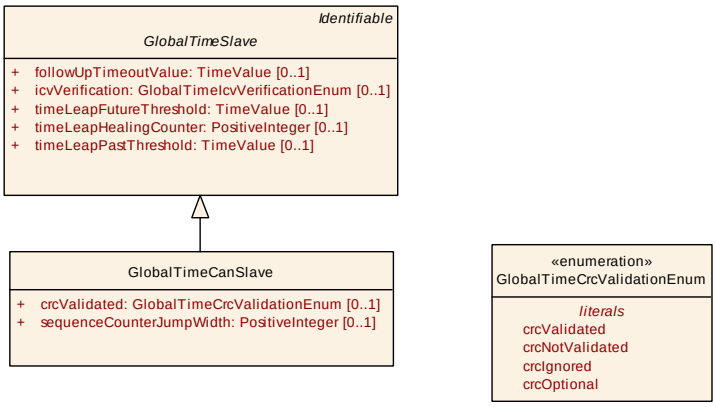

Figure 9.11: Modeling of the `GlobalTimeCanSlave`

In addition to the CAN specific Master and Slave properties CAN specific `CanGlobalTimeDomainProps` can be described.

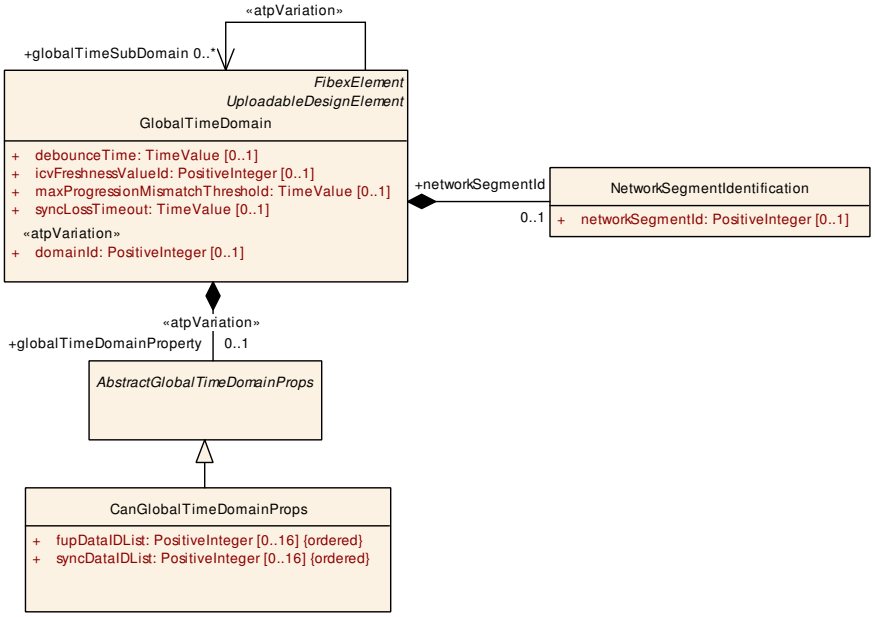

Figure 9.12: Modeling of the CAN specific `CanGlobalTimeDomainProps`

Class	**GlobalTimeCanMaster**			
Note	This represents the specialization of the GlobalTimeMaster for the CAN communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
crcSecured	GlobalTimeCrcSupport Enum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.
sync Confirmation Timeout	TimeValue	0..1	attr	This represents the value for the confirmation timeout. Unit: seconds. **Tags:** atp.Status=obsolete

Table 9.9: GlobalTimeCanMaster

Class	GlobalTimeCanSlave			
Note	This represents the specialization of the GlobalTimeSlave for the CAN communication.			
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.slave			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcValidated	GlobalTimeCrc ValidationEnum	0..1	attr	Definition of whether or not validation of the CRC is supported.
sequence CounterJump Width	PositiveInteger	0..1	attr	Specifies the maximum allowed gap of the sequence counter between two SYNC messages.

Table 9.10: GlobalTimeCanSlave

[constr_9310] Existence of `GlobalTimeCanSlave.crcValidated`

Imposition time: CP: `IT_SysDesc`

⌈For each `GlobalTimeCanSlave`, the attribute `crcValidated` shall exist.⌋

Class	CanGlobalTimeDomainProps			
Note	Enables the definition of Can Global Time specific properties.			
Base	*ARObject*, *AbstractGlobalTimeDomainProps*			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
fupDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for FUP messages to calculate CRC.
syncDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for SYNC messages to calculate CRC.

Table 9.11: CanGlobalTimeDomainProps

9.3.2 Time Synchronization over Ethernet

This chapter described the detailing of how the concept of *global time synchronization* is applied to the Ethernet bus in particular. For details concerning the functional behavior please refer to [52].

The implementation of *global time synchronization* on the Ethernet bus is modeled by means of `GlobalTimeEthMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `GlobalTimeEthSlave`, which is derived from `GlobalTimeSlave`.

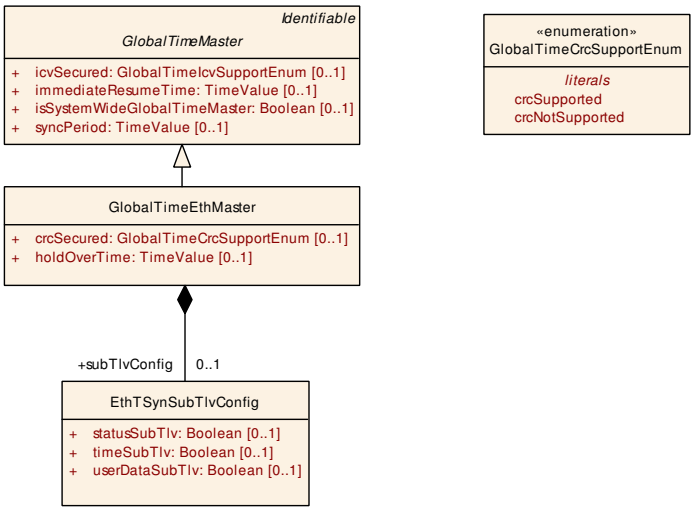

Figure 9.13: Modeling of the `GlobalTimeEthMaster`

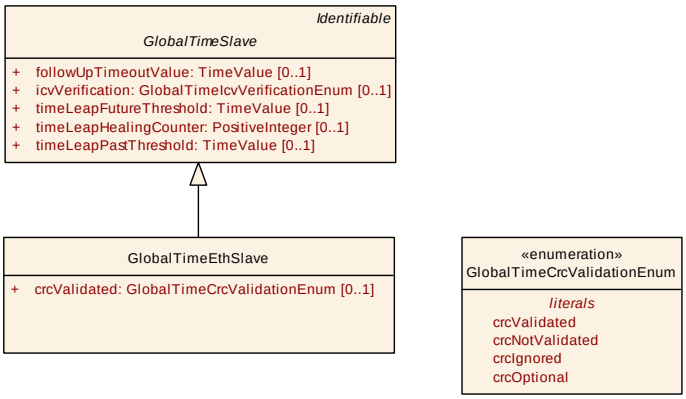

Figure 9.14: Modeling of the `GlobalTimeEthSlave`

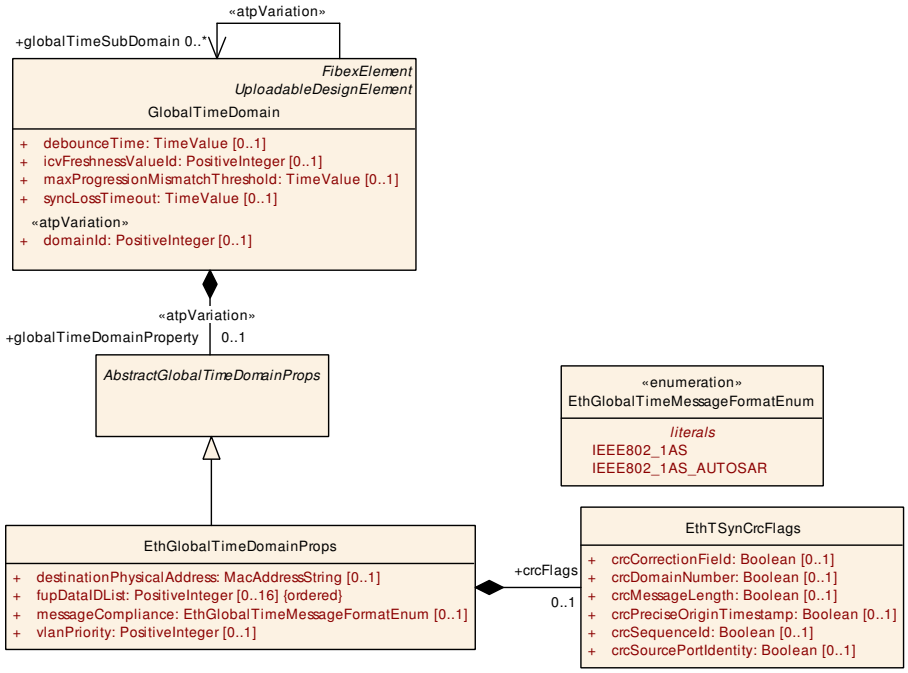

Figure 9.15: Modeling of the `EthGlobalTimeDomainProps`

Class	**GlobalTimeEthMaster**			
Note	This represents the specialization of the GlobalTimeMaster for Ethernet communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
crcSecured	GlobalTimeCrcSupportEnum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.
holdOverTime	TimeValue	0..1	attr	This attribute defines the timeout for transmission of Sync and Follow_Up messages on Master ports in absence of reception of Sync and Follow_Up messages on Slave port.
subTlvConfig	EthTSynSubTlvConfig	0..1	aggr	Defines the subTLV fields which shall be included in the time sync message. **Tags:** atp.Splitkey=subTlvConfig

Table 9.12: GlobalTimeEthMaster

Class	**EthTSynSubTlvConfig**			
Note	Defines the subTLV fields which shall be included in the time sync message.			
Base	*ARObject*			
Aggregated by	GlobalTimeEthMaster.subTlvConfig			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
statusSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used.
timeSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used.
userDataSubTlv	Boolean	0..1	attr	Defines whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used.

Table 9.13: EthTSynSubTlvConfig

Class	GlobalTimeEthSlave			
Note	This represents the specialization of the GlobalTimeSlave for Ethernet communication.			
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.slave			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcValidated	GlobalTimeCrc ValidationEnum	0..1	attr	Definition of whether or not validation of the CRC is supported.

Table 9.14: GlobalTimeEthSlave

Class	EthGlobalTimeDomainProps			
Note	Enables the definition of Ethernet Global Time specific properties.			
Base	*ARObject*, *AbstractGlobalTimeDomainProps*			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcFlags	EthTSynCrcFlags	0..1	aggr	Defines the fields of the message which shall be taken into account for CRC calculation and verification. **Tags:** atp.Splitkey=crcFlags
destination Physical Address	MacAddressString	0..1	attr	Defines the MAC multicast address the Ethernet time sync messages are communicated on.
fupDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for FUP messages to calculate CRC.
managed CouplingPort	EthGlobalTime ManagedCouplingPort	*	aggr	Collection of CouplingPorts which are managed in the scope of this Ethernet GlobalTimeDomain. **Tags:** atp.Splitkey=managedCouplingPort
message Compliance	EthGlobalTimeMessage FormatEnum	0..1	attr	Defines the compliance of the Ethernet time sync messages to specific standards.
vlanPriority	PositiveInteger	0..1	attr	Defines which VLAN priority shall be assigned to a time sync message in case the message is sent using a VLAN tag.

Table 9.15: EthGlobalTimeDomainProps

[constr_9311] Existence of `EthGlobalTimeDomainProps.messageCompliance`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈For each `EthGlobalTimeDomainProps`, the attribute `messageCompliance` shall exist.⌋

Class	EthTSynCrcFlags			
Note	Defines the fields of the message which shall be taken into account for CRC calculation and verification.			
Base	*ARObject*			
Aggregated by	EthGlobalTimeDomainProps.crcFlags			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcCorrection Field	Boolean	0..1	attr	CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.
crcDomain Number	Boolean	0..1	attr	DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.
crcMessage Length	Boolean	0..1	attr	MessageLength from the Follow_Up Message Header shall be included in CRC calculation.

▽

△

Class	**EthTSynCrcFlags**			
crcPrecise Origin Timestamp	Boolean	0..1	attr	PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.
crcSequenceId	Boolean	0..1	attr	SequenceId from the Follow_Up Message Header shall be included in CRC calculation.
crcSourcePort Identity	Boolean	0..1	attr	SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.

Table 9.16: EthTSynCrcFlags

Enumeration	**EthGlobalTimeMessageFormatEnum**
Note	Specifies which message formats are available to for the Ethernet time sync protocol.
Aggregated by	EthGlobalTimeDomainProps.messageCompliance
Literal	***Description***
IEEE802_1AS	Message format according to IEEE 802.1AS standard. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=IEEE802-1AS
IEEE802_1AS_ AUTOSAR	Message format according to IEEE 802.1AS standard with AUTOSAR extensions. **Tags:** atp.EnumerationLiteralIndex=1 xml.name=IEEE802-1AS-AUTOSAR

Table 9.17: EthGlobalTimeMessageFormatEnum

[constr_3312] Consistency of `vlanPriority` and `EthernetCommunicationConnector`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈A `GlobalTimeEthMaster` refers to an `EthernetCommunicationConnector` in the role `communicationConnector`. If that `EthernetCommunicationConnector` is referenced by an `EthernetPhysicalChannel` in the role `commConnector` and the `EthernetPhysicalChannel` has a vLan tag defined via the `VlanConfig` then the `GlobalTimeDomain` of the `GlobalTimeEthMaster` shall aggregate `EthGlobalTimeDomainProps` in the role `globalTimeDomainProperty` and the attribute `EthGlobalTimeDomainProps.vlanPriority` shall exist.⌋

In Ethernet networks the usage of Ethernet switches introduces another layer of delay in the transportation of data. This of course also applies to the global time synchronization messages and there are means to compensate these delays available in AUTOSAR.

In order to cope with delays on global time sync of Ethernet transport technology two use-cases are supported:

- an ECU is connected to an Ethernet switch but does not manage the switch (see section 9.3.2.2)
- an ECU is connected to an Ethernet switch and also manages this switch (see section 9.3.2.3)

The CouplingPort is used in either use-case to describe the connection of the ECU to the Ethernet network / switch. Thus there are some attributes related to the CouplingPort which apply to both use-cases.

9.3.2.1 Time Synchronization and Ethernet propagation delay

The propagation delay measurement is applicable to the CouplingPort in scope of the global time synchronization.

The default propagation delay time (which is used if propagation delay measurement is disabled or is not yet measured) is defined at the GlobalTimeCouplingPortProps with the attribute propagationDelay. The GlobalTimeCouplingPortProps are aggregated at the CouplingPortDetails in the role globalTimeProps.

Whether an ECU shall initiate a propagation delay measurement at a certain CouplingPort and on a specific GlobalTimeDomain is defined by the attribute pdelayRequestPeriod at the EthGlobalTimeManagedCouplingPort.

[TPS_SYST_03016] Applicability of EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod ⌈When EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod is not defined or has the value 0 then initiation of propagation delay measurement is disabled for the CouplingPort referenced by couplingPort and the GlobalTimeDomain the EthGlobalTimeManagedCouplingPort belongs to.⌋

Whether an ECU shall respond to propagation delay measurement at a certain CouplingPort and on a specific GlobalTimeDomain is defined by the attribute pdelayResponseEnabled at the EthGlobalTimeManagedCouplingPort.

9.3.2.2 Time Synchronization and Ethernet connection

In case the ECU is directly connected to the Ethernet network and does not manage an Ethernet switch (of course there may be Ethernet switches used in the topology, but for this use-case these switches are not visible to the description of this ECU) the Ethernet time synchronization only needs to cope with the connection of this ECU to the Ethernet network. Considering the example in figure 9.17 this applies to the ECUs TS1, TS2, TS3, and TM which are just connected to the Ethernet switches but do not manage any Ethernet switch.

[TPS_SYST_03017] Reference to CouplingPort in the context of a GlobalTimeDomain ⌈In case a GlobalTimeDomain is communicated via a CouplingPort and the respective ECU does not manage an Ethernet switch then the reference EthGlobalTimeManagedCouplingPort.couplingPort shall reference a CouplingPort which is aggregated by the EthernetCommunicationController in the role couplingPort. The EthernetCommunicationController itself shall be referenced by a GlobalTimeMaster or GlobalTimeEthSlave (via the

CommunicationConnector) and that GlobalTimeMaster or GlobalTimeEth-Slave shall be aggregated by the GlobalTimeDomain initially mentioned.⌋

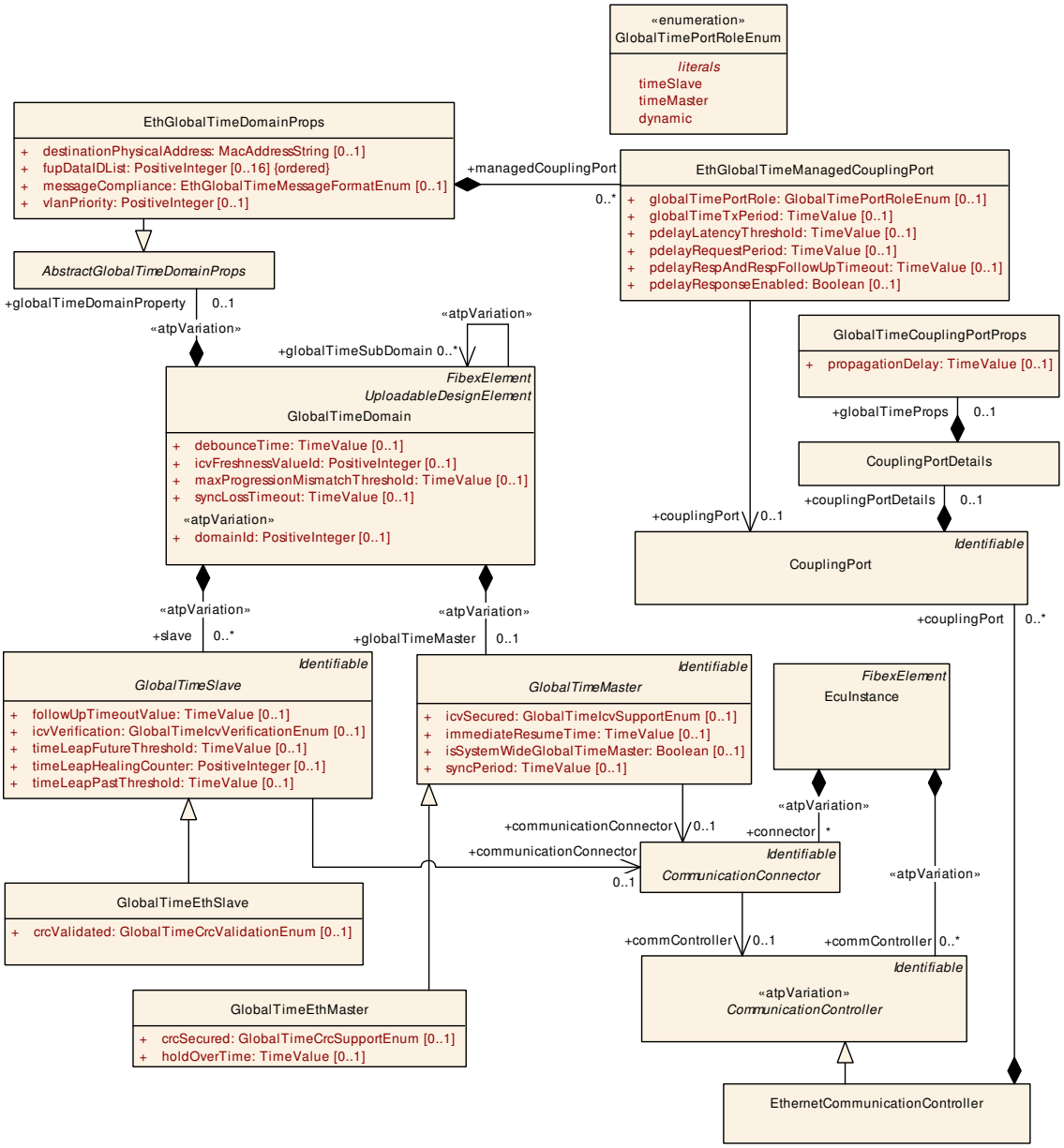

Figure 9.16: Overview of the Ethernet time sync in relation with a CouplingPort of an ECU

9.3.2.3 Time Synchronization and managed Ethernet switch

In case an ECU manages an Ethernet switch then that management ECU can basically be the GlobalTimeEthMaster or the GlobalTimeEthSlave located (see also figure 9.17). For the description of the time delay compensation on System Template level this does not matter.

It is essential to configure all possible time synchronization communication paths between the involved entities.

In case of ECU A in figure 9.17 the GlobalTimeEthMaster shall

- refer to the CouplingPort 0 since this is where
  - all time sync messages will be sent out by the GlobalTimeEthMaster
  - all adjusted follow up messages will be sent out by the GlobalTimeEthMaster
- refer to the CouplingPort 1 because this is where the Ethernet switch is connected to the management ECU A
- refer to the CouplingPorts 2, 3, and 4 since this is where the time sync messages will be forwarded by the switch
- not refer to the CouplingPort 5 because this one is not involved in that GlobalTimeDomains communication

In case of ECU B in figure 9.17 the GlobalTimeEthSlave shall

- refer to the CouplingPort 0 since this is where
  - the time sync messages will be received by the GlobalTimeEthSlave
  - all adjusted follow up messages will be sent out by the GlobalTimeEthSlave
- refer to the CouplingPort 1 because this is where the Ethernet switch is connected to the management ECU B
- refer to the CouplingPort 3 because this is where the time sync messages will be received from the GlobalTimeEthMaster on ECU TM
- refer to the CouplingPorts 2 and 4 since this is where the time sync messages will be forwarded by the switch
- not refer to the CouplingPort 5 because this one is not involved in that GlobalTimeDomains communication

Please note that the non-involvement of the CouplingPort 5 is used for illustration purposes. It would also be possible to involve CouplingPort 5 in that GlobalTimeDomain definition although currently there is no ECU as an GlobalTimeEthSlave defined. In that case the CouplingPort 5 is prepared to be connected to an ECU with an GlobalTimeEthSlave later.

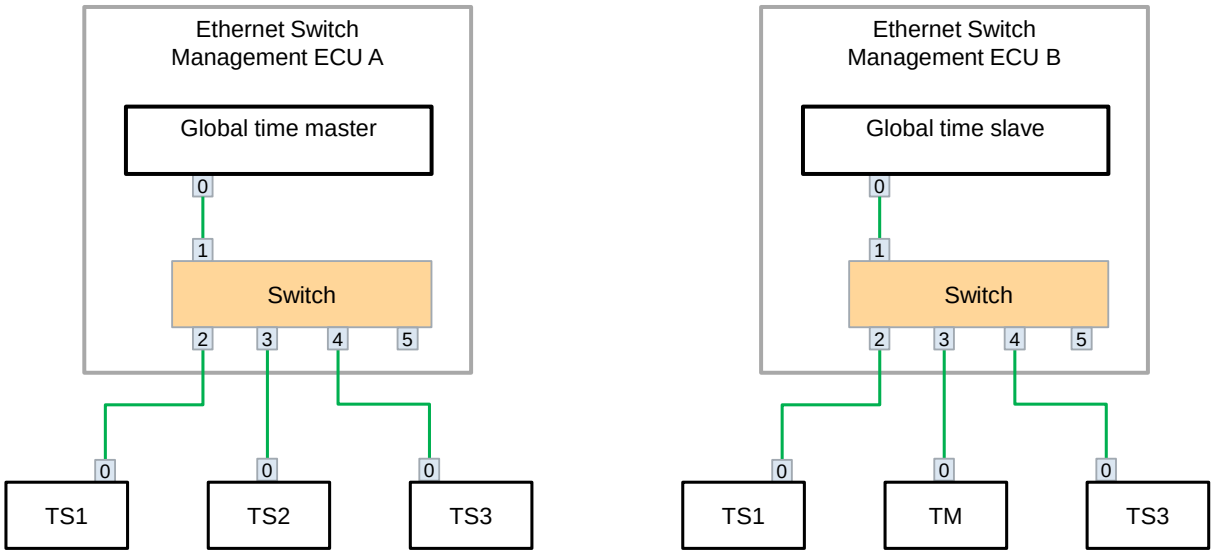

Figure 9.17: Example of a managed Ethernet Switch

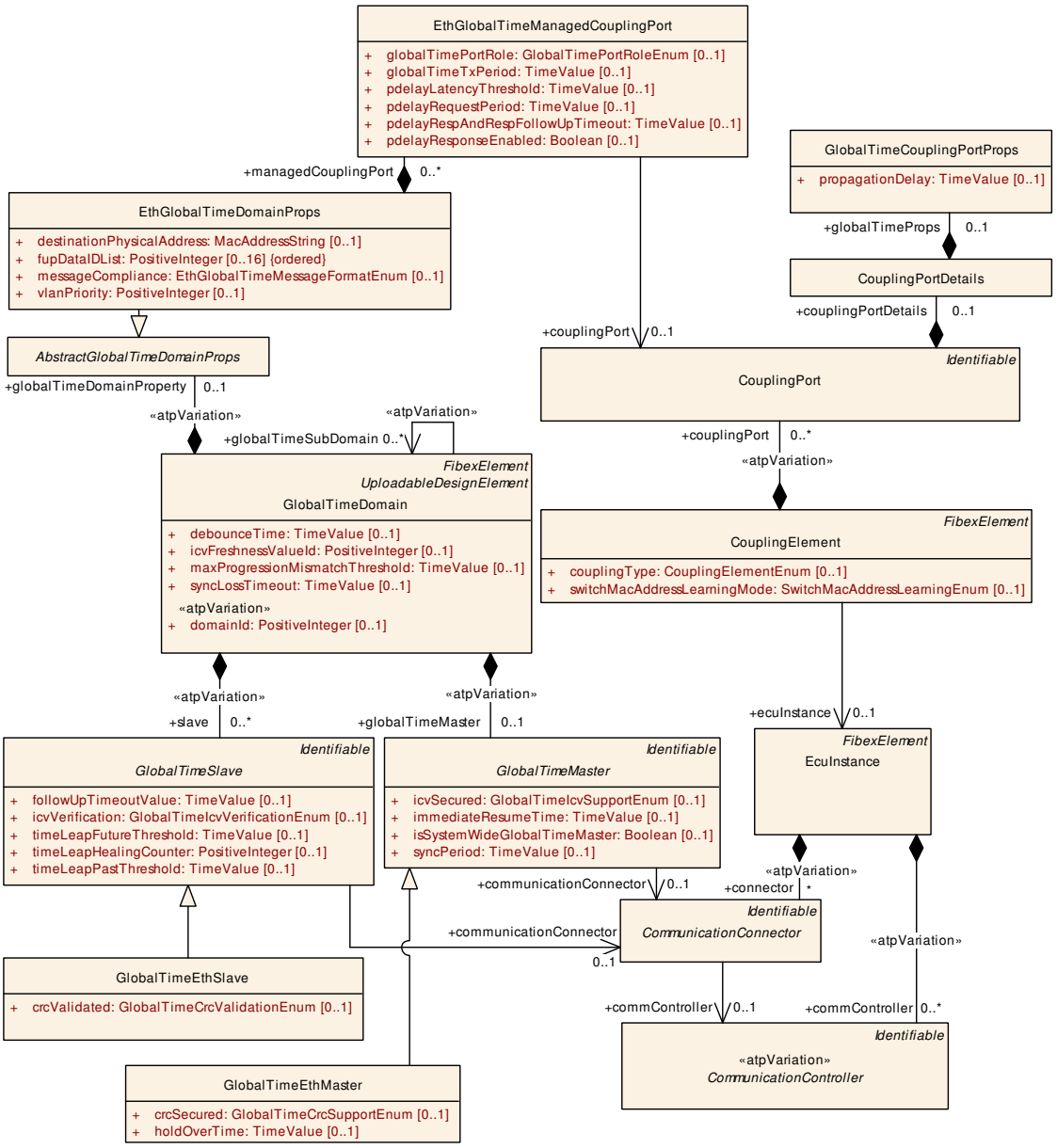

Figure 9.18: Overview of the Ethernet time sync in relation with a `CouplingPort` of an Ethernet switch

Class	**EthGlobalTimeManagedCouplingPort**			
Note	Specifies a CouplingPort which is managed by an Ethernet Global Time Domain.			
Base	*ARObject*			
Aggregated by	EthGlobalTimeDomainProps.managedCouplingPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
couplingPort	CouplingPort	0..1	ref	Defines which CouplingPort is managed by this EthGlobal TimeManagedCouplingPort. **Tags:** atp.Splitkey=couplingPort

△

Class	**EthGlobalTimeManagedCouplingPort**			
globalTimePort Role	GlobalTimePortRole Enum	0..1	attr	This attribute defines the port behavior.
globalTimeTx Period	TimeValue	0..1	attr	This attribute defines the TX period in seconds
pdelayLatency Threshold	TimeValue	0..1	attr	Threshold for calculated Pdelay. If a measured Pdelay exceeds pdelayLatencyThreshold, the measured Pdelay value is discarded.
pdelayRequest Period	TimeValue	0..1	attr	Defines the period for the pdelay request messages.
pdelayRespAnd RespFollowUp Timeout	TimeValue	0..1	attr	Timeout value for Pdelay_Resp and Pdelay_Resp_ Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 or not defining this attribute deactivates this timeout observation.
pdelay Response Enabled	Boolean	0..1	attr	Defines whether PDELAY RESPONSE and PDELAY RESPONSE FOLLOW UP shall be sent on this Coupling Port.

Table 9.18: EthGlobalTimeManagedCouplingPort

Class	**GlobalTimeCouplingPortProps**			
Note	Defines properties for the usage of the CouplingPort in the scope of Global Time Sync.			
Base	*ARObject*			
Aggregated by	CouplingPortDetails.globalTimeProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
propagation Delay	TimeValue	0..1	attr	If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter defines a fixed value for the propagation delay.

Table 9.19: GlobalTimeCouplingPortProps

[constr_9312] Existence of EthGlobalTimeManagedCouplingPort.pdelayResponseEnabled

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each EthGlobalTimeManagedCouplingPort, the attribute pdelayResponseEnabled shall exist.⌋

[constr_9313] Existence of GlobalTimeCouplingPortProps.propagationDelay

Imposition time: CP: IT_SysDesc, AP: IT_SysDes

⌈For each GlobalTimeCouplingPortProps, the attribute propagationDelay shall exist.⌋

Enumeration	**GlobalTimePortRoleEnum**
Note	Selection of port behavior to Time Slave, Time Master or Dynamic (Time Slave or Time Master at runtime).
Aggregated by	EthGlobalTimeManagedCouplingPort.globalTimePortRole
Literal	***Description***
dynamic	Time Slave or Time Master port behavior at runtime. **Tags:** atp.EnumerationLiteralIndex=2
timeMaster	timeMaster port behavior **Tags:** atp.EnumerationLiteralIndex=1
timeSlave	TimeSlave port behavior **Tags:** atp.EnumerationLiteralIndex=0

Table 9.20: GlobalTimePortRoleEnum

[constr_5389] Dependency between `globalTimeTxPeriod` and `globalTimePortRole`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `EthGlobalTimeManagedCouplingPort.globalTimeTxPeriod` shall only be set to a value if the attribute `EthGlobalTimeManagedCouplingPort.globalTimePortRole` is set to `timeMaster` or `dynamic`.⌋

The reason for [constr_5389] is that `EthGlobalTimeManagedCouplingPort.globalTimeTxPeriod` determines the transmission interval of Sync messages.

[constr_5390] The `globalTimePortRole` shall not be configured to `timeSlave` several times in the same `GlobalTimeDomain`

Imposition time: CP: `IT_SysDesc`, AP: `IT_SysDes`

⌈The attribute `globalTimePortRole` shall not be set to `timeSlave` for two or more `EthGlobalTimeManagedCouplingPort`s that are aggregated by the same `GlobalTimeDomain` (via `globalTimeDomainProperty`).⌋

9.3.2.4 Fine-Grained `GlobalTimeDomain` Ethernet switch configuration

The attributes defined at the `GlobalTimeDomain` (as depicted in figure 9.18) are in many cases applicable to the whole Ethernet switch. The `GlobalTimeMaster` and all the `GlobalTimeSlave`s representing the Ethernet switch ports often share identical values for the attributes.

In figure 9.19 the example of a global time configuration for a switch involving 3 ports is shown. The *GlobalTimeDomain_0* is received via *Port 1* and is forwarded to *Port 2* and *Port 4*.

The `EthGlobalTimeDomainProps` holds the port specific settings in the 3 `EthGlobalTimeManagedCouplingPort`s.

The GlobalTimeDomain defines the attribute icvFreshnessValueId and the reference to the SecOcSecureComProps where the value for authentication-BuildAttempts is defined.

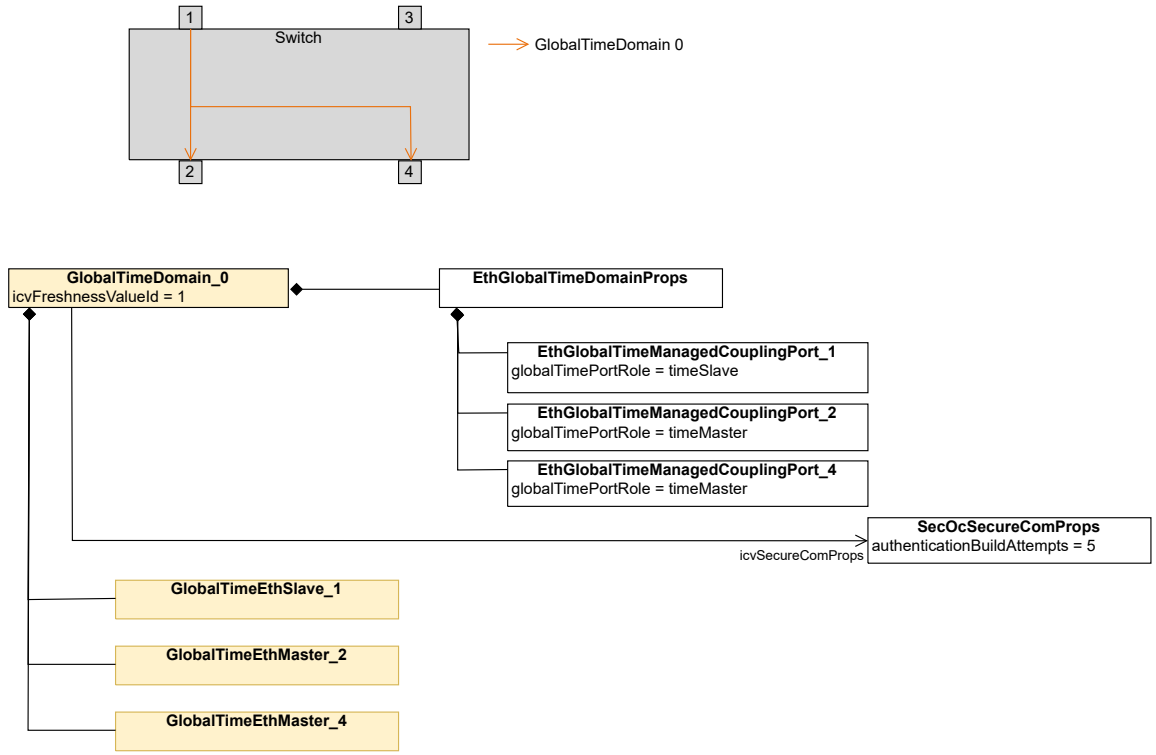

Figure 9.19: Example of a managed Ethernet Switch ports

But if, for example, the values for icvFreshnessValueId and authentication-BuildAttempts need to be configured differently for the individual Ethernet switch ports, then an extended modeling of the GlobalTimeDomain is required.

In order to represent the 3 Ethernet switch ports using the GlobalTimeDomain the *GlobalTimeDomain_0* object is replicated to define 3 globalTimeSubDomains.

GlobalTimeDomain_sub01 represents the GlobalTimeEthSlave where the global time information is received. The EthGlobalTimeDomainProps aggregated by *GlobalTimeDomain_sub01* define *EthGlobalTimeManagedCouplingPort_1* where globalTimePortRole is set to timeSlave. For *GlobalTimeDomain_sub02* and *GlobalTimeDomain_sub04* corresponding settings of globalTimePortRole to timeMaster are defined.

With the duplicated GlobalTimeDomains it is now also possible to define individual values for icvFreshnessValueId and authenticationBuildAttempts, as shown in figure 9.20

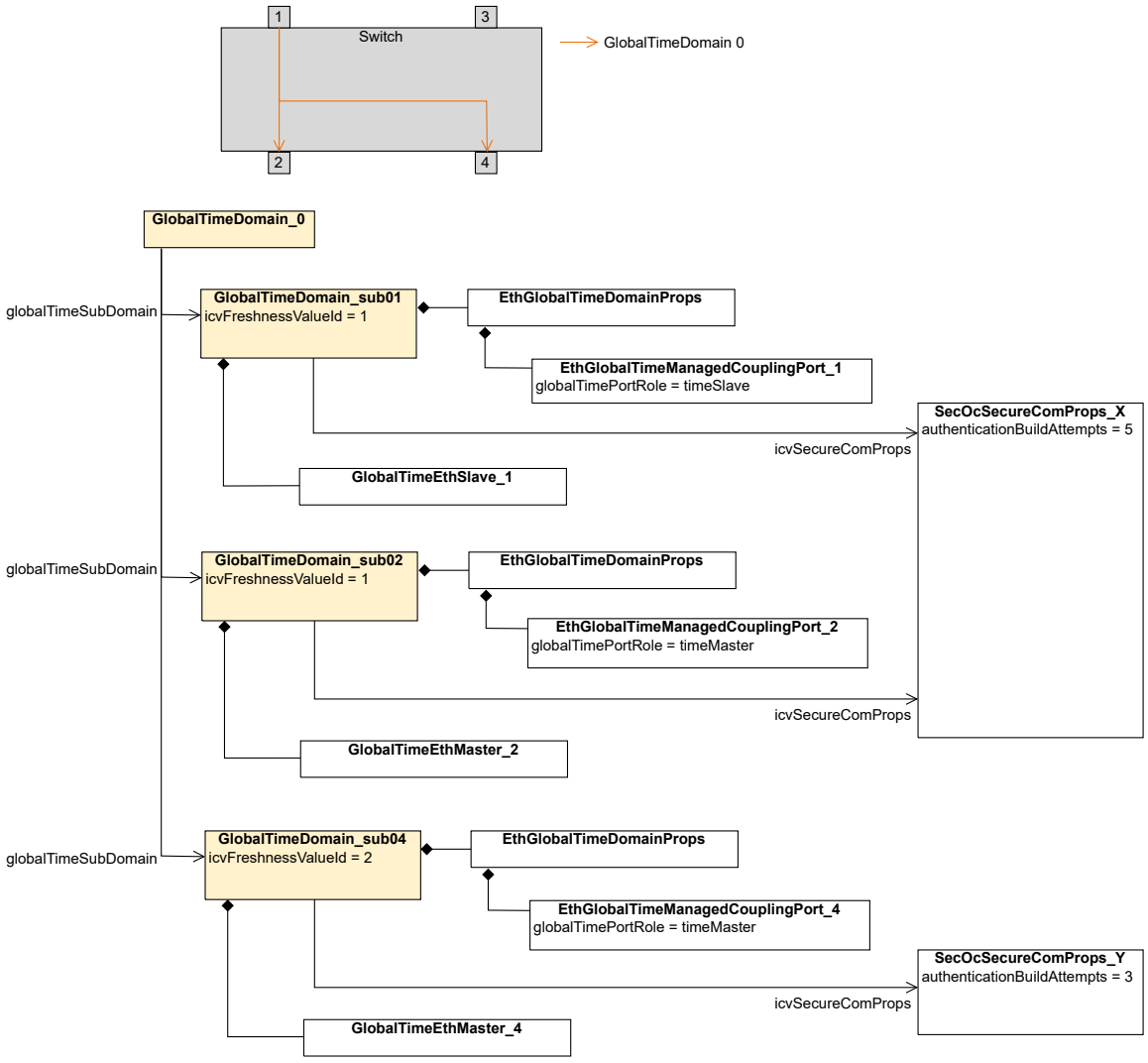

Figure 9.20: Example of a managed Ethernet Switch with individual configuration values

9.3.3 Time Synchronization over FlexRay

This chapter described the detailing of how the concept of *global time synchronization* is applied to the Flexray bus in particular.

The implementation of *global time synchronization* on the Flexray bus is modeled by means of `GlobalTimeFrMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `GlobalTimeFrSlave`, which is derived from `GlobalTimeSlave`.

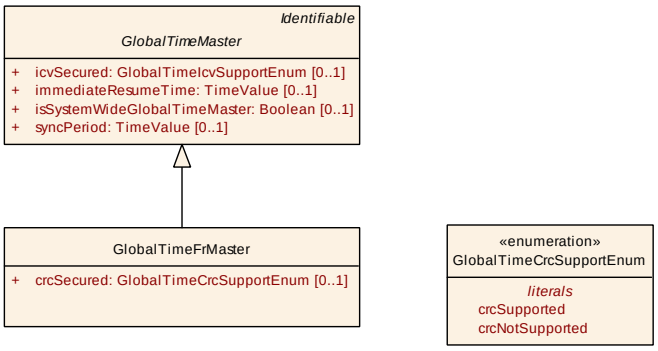

Figure 9.21: Modeling of the `GlobalTimeFrMaster`

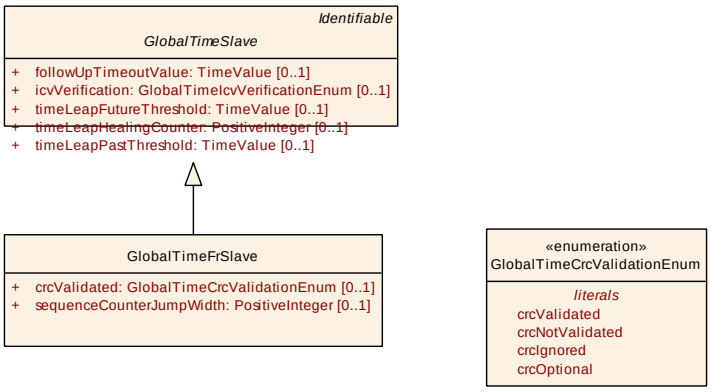

Figure 9.22: Modeling of the `GlobalTimeFrSlave`

In addition to the FlexRay specific Master and Slave properties FlexRay specific `FrGlobalTimeDomainProps` can be described.

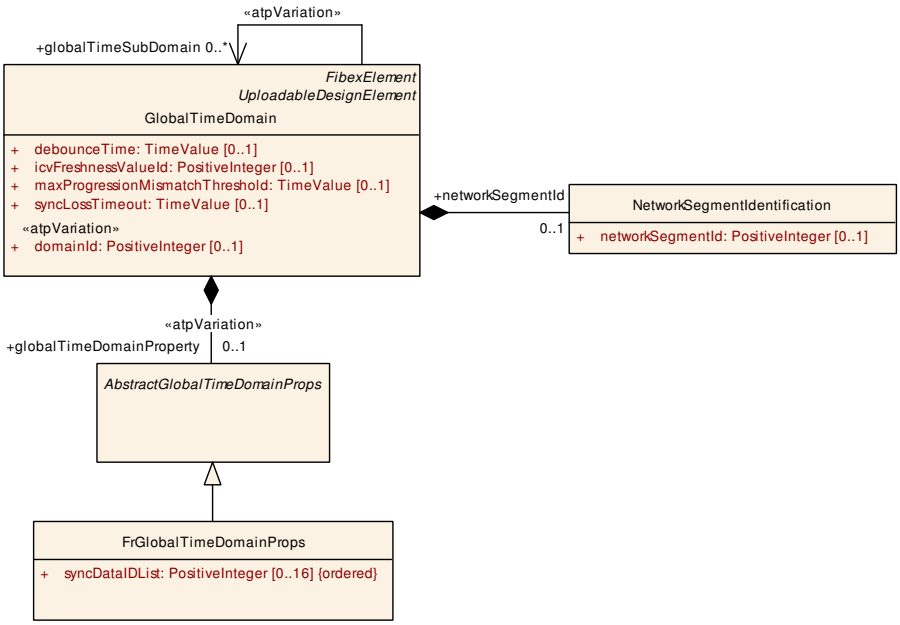

Figure 9.23: Modeling of the FlexRay specific `FrGlobalTimeDomainProps`

Class	GlobalTimeFrMaster			
Note	This represents the specialization of the GlobalTimeMaster for Flexray communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcSecured	GlobalTimeCrcSupport Enum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.

Table 9.21: GlobalTimeFrMaster

Class	GlobalTimeFrSlave			
Note	This represents the specialization of the GlobalTimeSlave for Flexray communication.			
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.slave			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
crcValidated	GlobalTimeCrc ValidationEnum	0..1	attr	Definition of whether or not validation of the CRC is supported.
sequence CounterJump Width	PositiveInteger	0..1	attr	Specifies the maximum allowed gap of the sequence counter between two SYNC messages.

Table 9.22: GlobalTimeFrSlave

[constr_9314] Existence of `GlobalTimeFrMaster.crcSecured`

Imposition time: CP: `IT_SysDesc`

⌈For each `GlobalTimeFrMaster`, the attribute `crcSecured` shall exist.⌋

[constr_9315] Existence of `GlobalTimeFrSlave.crcValidated`

Imposition time: CP: `IT_SysDesc`

⌈For each `GlobalTimeFrSlave`, the attribute `crcValidated` shall exist.⌋

[constr_9366] `GlobalTimeSlave.followUpTimeoutValue` not applicable to `GlobalTimeFrSlave`

Imposition time: CP: `IT_SysDesc`

⌈The attribute `GlobalTimeSlave.followUpTimeoutValue` shall not be defined for a `GlobalTimeFrSlave`.⌋

Class	FrGlobalTimeDomainProps			
Note	Enables the definition of Flexray GlobalTime specific properties.			
Base	*ARObject*, *AbstractGlobalTimeDomainProps*			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
syncDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for SYNC messages to calculate CRC.

Table 9.23: FrGlobalTimeDomainProps

9.3.4 Time Synchronization by user defined Timebase Provider

This chapter describes the details of how the concept of global time synchronization is applied to user defined Timebase Providers. The implementation of global time synchronization by user defined timebase providers is modeled by means of `UserDefinedGlobalTimeMaster`, a concrete subclass of `GlobalTimeMaster`. A similar approach applies for the `UserDefinedGlobalTimeSlave`, which is derived from `GlobalTimeSlave`.

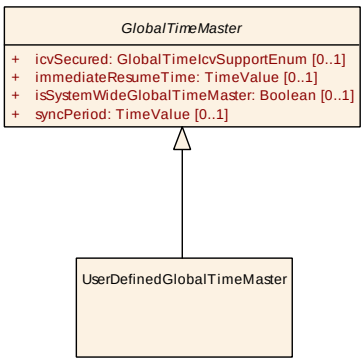

Figure 9.24: Modeling of the `UserDefinedGlobalTimeMaster`

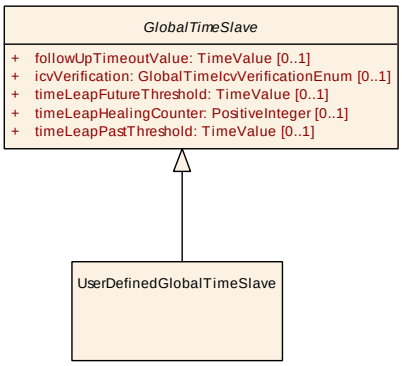

Figure 9.25: Modeling of the `UserDefinedGlobalTimeSlave`

Class	**UserDefinedGlobalTimeMaster**			
Note	This represents the specialization of the GlobalTimeMaster for user defined communication.			
Base	*ARObject*, *GlobalTimeMaster*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 9.24: UserDefinedGlobalTimeMaster

Class	**UserDefinedGlobalTimeSlave**
Note	This represents the specialization of the GlobalTimeSlave for user defined communication.
Base	*ARObject*, *GlobalTimeSlave*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	GlobalTimeDomain.slave

△

Class	**UserDefinedGlobalTimeSlave**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 9.25: UserDefinedGlobalTimeSlave

9.3.5 Time Synchronization Common Properties

The purpose of this chapter is basically to provide the class tables of meta-classes taken to implement configuration properties in the context of *global time synchronization*. The specifics about how these meta-classes are used is explained in the bus-specific chapters (i.e. chapters 9.3.1, 9.3.2, and 9.3.3).

Enumeration	**GlobalTimeCrcSupportEnum**
Note	This enumeration is used to define whether and how CRC on the TX side shall be utilized.
Aggregated by	GlobalTimeCanMaster.crcSecured, GlobalTimeEthMaster.crcSecured, GlobalTimeFrMaster.crcSecured
Literal	***Description***
crcNotSupported	This indicates that CRC is not supported **Tags:** atp.EnumerationLiteralIndex=0
crcSupported	This indicates that CRC is supported **Tags:** atp.EnumerationLiteralIndex=1

Table 9.26: GlobalTimeCrcSupportEnum

Enumeration	**GlobalTimeCrcValidationEnum**
Note	This enumeration provides values for the evaluation of the CRC
Aggregated by	GlobalTimeCanSlave.crcValidated, GlobalTimeEthSlave.crcValidated, GlobalTimeFrSlave.crcValidated
Literal	***Description***
crcIgnored	The CRC is supposed to be ignored **Tags:** atp.EnumerationLiteralIndex=0
crcNotValidated	The CRC is not supposed to be present. If CRC is present the message is ignored. **Tags:** atp.EnumerationLiteralIndex=1
crcOptional	Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done. **Tags:** atp.EnumerationLiteralIndex=3
crcValidated	This CRC is supposed to be validated. **Tags:** atp.EnumerationLiteralIndex=2

Table 9.27: GlobalTimeCrcValidationEnum

Enumeration	**GlobalTimeIcvSupportEnum**
Note	Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*GlobalTimeMaster*.icvSecured
Literal	***Description***
icvNotSupported	The ICV is not supported **Tags:** atp.EnumerationLiteralIndex=1
icvSupported	The ICV is supported **Tags:** atp.EnumerationLiteralIndex=0

Table 9.28: GlobalTimeIcvSupportEnum

Enumeration	**GlobalTimeIcvVerificationEnum**
Note	This enumeration is used to define how an Integrity Check Value (ICV) shall be handled at the receiver. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	*GlobalTimeSlave*.icvVerification
Literal	***Description***
icvIgnored	If the ICV is present, then it is ignored **Tags:** atp.EnumerationLiteralIndex=2
icvNotVerified	The ICV is not supposed to be present. If the ICV is present, then the message is ignored. **Tags:** atp.EnumerationLiteralIndex=1
icvOptional	If the ICV is present, then it will be verified. If the ICV is not present, then this is also a valid reception (no verification required). **Tags:** atp.EnumerationLiteralIndex=3
icvVerified	The ICV is required and will be verified. **Tags:** atp.EnumerationLiteralIndex=0

Table 9.29: GlobalTimeIcvVerificationEnum

10 Description of Service Discovery Services in Classic Platform

10.1 Representation of Service Interfaces on VFB level

ServiceInterfaces in the Adaptive Platform and in SOME/IP consist of Events, Fields and Methods. AUTOSAR classic platform does not support ServiceInterfaces but provides the possibility to communicate in a service oriented way over SOME/IP.

To mimic a ServiceInterface in the Classic Platform any combination of ClientServerInterfaces, SenderReceiverInterfaces or TriggerInterfaces may be used.

[TPS_SYST_02283] Collection of ServiceInterface elements ⌈The collection of PortInterfaces that represent one ServiceInterface shall be wrapped by a Collection element with category SET and collectionSemantics SO_SERVICE_INTERFACE.⌋

An example of ServiceInterface named *RadarService* is shown in the following listing.

```
<AR-PACKAGE>
  <SHORT-NAME>ServiceInterfaces</SHORT-NAME>
  <ELEMENTS>
    <!-- /ServiceInterfaces/RadarService -->
    <COLLECTION>
      <SHORT-NAME>RadarService</SHORT-NAME>
      <CATEGORY>SET</CATEGORY>
      <COLLECTION-SEMANTICS>SO_SERVICE_INTERFACE</COLLECTION-SEMANTICS>
      <ELEMENT-REFS>
        <ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/PortInterfaces/
            RadarService_BrakeEvent/BrakeEvent</ELEMENT-REF>
        <ELEMENT-REF DEST="CLIENT-SERVER-OPERATION">/PortInterfaces/
            RadarService_Methods/Calibrate</ELEMENT-REF>
        <ELEMENT-REF DEST="CLIENT-SERVER-OPERATION">/PortInterfaces/
            RadarService_Methods/Adjust</ELEMENT-REF>
        <ELEMENT-REF DEST="COLLECTION">/Fields/RadarService_UpdateRate</
            ELEMENT-REF>
        <ELEMENT-REF DEST="COLLECTION">/FireAndForgetMethods/
            RadarService_FireAndForgetMethods</ELEMENT-REF>
      </ELEMENT-REFS>
    </COLLECTION>
  </ELEMENTS>
</AR-PACKAGE>
```

Listing 10.1: Example for a ServiceInterface Collection

The *RadarService* in this example consist of:

- an Event named *BrakeEvent*,
- a Method named *Calibrate*,

- a Method named *Adjust*,
- a Field named *RadarService_UpdateRate*,
- a Collection of Fire_And_Forget Methods.

10.1.1 Representation of Events

[TPS_SYST_02284] Event in a ServiceInterface ⌈An event in a ServiceInterface shall be described as a VariableDataPrototype in a SenderReceiverInterface.⌋

Please note that in SOME/IP and other description formats like ASAM FIBEX an Event consist of one or several Parameters. In AUTOSAR an Event with several parameters needs to be described as an VariableDataPrototype that is typed by an AutosarDataType of category STRUCTURE. Each parameter of the Event shall be represented as a member of the STRUCTURE.

10.1.2 Representation of Methods

[TPS_SYST_02285] Method in a ServiceInterface ⌈A method of a ServiceInterface shall be described as a ClientServerOperation in a ClientServerInterface.⌋

Each method parameter shall be described as an argument of the ClientServerOperation. Please note that the order of method parameters is expressed by the order of arguments.

10.1.3 Representation of Fire and Forget Methods

A so-called “fire & forget” method in SOME/IP represents a special form of a method dedicated to the sole purpose of conveying information from the service consumer to the service provider. The semantics of a “fire & forget” method is comparable to the semantics of an event, only reverse. In case that the ServiceInterface contains a “fire & forget” method the service consumer is able to call this method without the expectation of any response from the service provider.

[TPS_SYST_02286] “fire & forget” method with data in a ServiceInterface ⌈A “fire & forget” method of a ServiceInterface that contains data shall be described as a VariableDataPrototype in a SenderReceiverInterface. The service provider will consume the VariableDataPrototype. The service consumer will provide the VariableDataPrototype.⌋

If such a “fire & forget” method contains several parameters the VariableDataPrototype shall be typed by an AutosarDataType of category STRUCTURE. Each parameter of the “fire & forget” method shall be represented as a member of the STRUCTURE.

[TPS_SYST_02287] “Fire & forget” method without data in a ServiceInterface
⌈A “fire & forget” method of a ServiceInterface that does not contain any data shall be described as a Trigger in a TriggerInterface. The service provider will consume the Trigger. The service consumer will provide the Trigger.⌋

To distinguish “fire & forget” methods with parameters from Events a sub-collection is introduced in the ServiceInterface Collection. The sub-collection refers to all “fire & forget” methods represented as VariableDataPrototypes and Triggers that are defined by the ServiceInterface.

[TPS_SYST_02288] “Fire & forget” method in a ServiceInterface
⌈The “fire & forget” methods of ServiceInterface shall be wrapped by a Collection element with category SET and collectionSemantics SO_SERVICE_FIRE_AND_FORGET_METHODS.⌋

```
<AR-PACKAGE>
  <SHORT-NAME>FireAndForgetMethods</SHORT-NAME>
  <ELEMENTS>
    <!-- /FireAndForgetMethods/RadarService_FireAndForgetMethods-->
    <COLLECTION>
      <SHORT-NAME>RadarService_FireAndForgetMethods</SHORT-NAME>
      <CATEGORY>SET</CATEGORY>
      <COLLECTION-SEMANTICS>SO_SERVICE_FIRE_AND_FORGET_METHODS</COLLECTION-
          SEMANTICS>
      <ELEMENT-REFS>
        <ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/PortInterfaces/
            RadarService_Reset/Reset</ELEMENT-REF>
        <ELEMENT-REF DEST="TRIGGER">/PortInterfaces/
            RadarService_LogCurrentState/RadarService_LogCurrentState</
            ELEMENT-REF>
      </ELEMENT-REFS>
    </COLLECTION>
  </ELEMENTS>
</AR-PACKAGE>
```

Listing 10.2: Example for a fire & forget Method Collection

10.1.4 Representation of Fields

A Field represents a piece of data hosted by a server that exposes to one or more client(s) a get accessor and/or a set mutator. Clients can optionally receive notifications of changes of the Field’s value. In addition a Field has a concrete value at any time. As soon as the ServiceInterface is offered a Field of this ServiceInterface can be accessed by a client.

[TPS_SYST_02289] `Field` in a `ServiceInterface` ⌈The elements of a `Field` shall be wrapped by a `Collection` element with category SET and `collection-Semantics` SO_SERVICE_FIELD.⌋

[TPS_SYST_02290] `Field` elements ⌈A `Collection` element with category SO_SERVICE_FIELD is allowed to contain one, two or all three of the following items:

- a reference to a `VariableDataPrototype` representing a `Field Notifier`,
- a reference to a `ClientServerOperation` representing a `Field Getter`,
- a reference to a `ClientServerOperation` representing a `Field Setter`.

⌋

[TPS_SYST_02291] `Field Notifier` ⌈A `Field Notifier` is represented by a `VariableDataPrototype` in a `SenderReceiverInterface`.⌋

[TPS_SYST_02292] `Field Getter` ⌈A `Field Getter` is represented by a `Client-ServerOperation` with `ArgumentDataPrototype`s with `direction out`.⌋

After the getter call the requester will receive the current value of the `Field` in the response.

[TPS_SYST_02293] `Field Setter` ⌈A `Field Setter` is represented by a `Client-ServerOperation` with `ArgumentDataPrototype`s with `direction out` and `in`.⌋

Please note that it is the decision of the Service Provider to accept the setter request or to deny it. The current value of the Field will always be sent back to the requester as response.

```
<AR-PACKAGE>
  <SHORT-NAME>Fields</SHORT-NAME>
  <ELEMENTS>
    <!-- /Fields/RadarService_UpdateRate -->
    <COLLECTION>
      <SHORT-NAME>RadarService_UpdateRate</SHORT-NAME>
      <CATEGORY>SET</CATEGORY>
      <COLLECTION-SEMANTICS>SO_SERVICE_FIELD</COLLECTION-SEMANTICS>
      <ELEMENT-REFS>
        <ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/PortInterfaces/
            RadarService_UpdateRate_Notifier/UpdateRate</ELEMENT-REF>
        <ELEMENT-REF DEST="CLIENT-SERVER-OPERATION">/PortInterfaces/
            RadarService_UpdateRate_GetterSetter/Getter</ELEMENT-REF>
        <ELEMENT-REF DEST="CLIENT-SERVER-OPERATION">/PortInterfaces/
            RadarService_UpdateRate_GetterSetter/Setter</ELEMENT-REF>
      </ELEMENT-REFS>
    </COLLECTION>
  </ELEMENTS>
</AR-PACKAGE>
```

Listing 10.3: Example for a `Field` `Collection`

[constr_5330] `ServiceInterface` elements shall belong to exactly one `ServiceInterface`

Imposition time: CP: `IT_SysDesc`

⌈If an element like

- a `VariableDataPrototype` that represents a `ServiceInterface event`
- a `ClientServerOperation` that represents a `ServiceInterface method`
- a `Collection` with `collectionSemantics` SO_SERVICE_FIELD that represents a `ServiceInterface field`
- a `Collection` with `collectionSemantics` SO_SERVICE_FIRE_AND_FORGET_METHOD that represents a "fire & forget" `method`

is referenced in the role `element` by a `Collection` that has the `collectionSemantics` set to SO_SERVICE_INTERFACE then this element shall not be referenced by any other `Collection` element that has the `collectionSemantics` SO_SERVICE_INTERFACE in the scope of the `System`.⌋

[constr_5330] is introduced to mimic the `ServiceInterface` modeling in the Adaptive Platform where the `Event`, `Method` and `Field` are aggregated by the `ServiceInterface`.

10.2 Representation of Service Interfaces on network level

Elements of `ServiceInterface`s are mapped by the `DataMapping` to `SystemSignal`s and later to `ISignal`s and `ISignalIPdu`s for transport over the network. For the serialization as an UINT8 Array two different Transformers are supported, as defined in [TPS_SYST_02294] and [TPS_SYST_02295].

[TPS_SYST_02294] Serialization of `ServiceInterfaces` using `ComBasedTransformer` ⌈If the `ComBasedTransformer` is used to serialize the members of a `ServiceInterface`, then

- the `SenderReceiverToSignalGroupMapping` is used to map the `VariableDataPrototype` that represents an `Event` or `Field Notifier` to a `SystemSignalGroup`. Please note that the datatype of an `Event` is a Structure as described above. Each primitive member of this Structure is mapped by the `SenderRecRecordElementMapping` to a `SystemSignal` in the `SystemSignalGroup`. The `SystemSignalGroup` is mapped as `ISignalGroup` into an `ISignalIPdu` that is transported over the network.
- the serialization of `Method`s, `Field Setter`s, `Field Getter`s by the ComBasedTransformer is not supported.

⌋

[TPS_SYST_02295] Serialization of ServiceInterfaces using SomeipTransformer ⌈If the SomeipTransformer is used to serialize the members of a ServiceInterface, then

- the SenderReceiverToSignalMapping is used to map the VariableDataPrototype that represents the Event or Field Notifier to a single SystemSignal. The SystemSignal is mapped as an ISignal into an ISignalIPdu that is transported over the network.
- the ClientServerToSignalMapping is used to map the ClientServerOperation that represents the Method or Field Setter or Field Getter to a Call-SystemSignal and to a Return-SystemSignal. Both SystemSignals are mapped as ISignals into dedicated ISignalIPdus that are transported over the network.
- the SenderReceiverToSignalMapping is used to map the VariableDataPrototype that represents the "fire & forget" method with data to a SystemSignal. The SystemSignal is mapped as ISignal into an ISignalIPdu that is transported over the network. The messageType in the SOMEIPTransformationISignalProps that is aggregated by the ISignal shall be set to requestNoReturn.
- the TriggerToSignalMapping is used to map the Trigger that represents the "fire & forget" method without data to a SystemSignal. The SystemSignal is mapped as ISignal into an ISignalIPdu that is transported over the network. The messageType in the SOMEIPTransformationISignalProps that is aggregated by the ISignal shall be set to requestNoReturn.

⌋

The ServiceInterfaces itself are represented as ServiceInstances in the System Description. The ProvidedServiceInstance is used to describe that a specific instance of a ServiceInterface is provided on an ApplicationEndpoint. The ConsumedServiceInstance is used to describe that a search for a specific instance of a ServiceInterface is executed on an ApplicationEndpoint.

The Pdus that represent the elements of the ServiceInterface are attached to the ServiceInstances via PduActivationRoutingGroups.

Please note that the Pdus that represent the Methods, "fire & forget" methods, Field Setter and Field Getter are assigned to the ServiceInstance in a methodActivationRoutingGroup since they are activated by the SD module as soon as the Service is offered.

[TPS_SYST_03131] "Fire & Forget" method assignment to AbstractServiceInstances ⌈A Pdu with an ISignal that is used to transmit a SystemSignal to which a VariableDataPrototype is mapped that represents a "fire & forget" method (see [TPS_SYST_02286] and [TPS_SYST_02287]) is assigned to an AbstractServiceInstance via a methodActivationRoutingGroup that references the PduTriggering of the Pdu via a SoConIPduIdentifier.⌋

The Pdus that represent the Events and Field Notifiers are assigned to EventHandlers on the provided side as pduActivationRoutingGroups or to ConsumedEventGroups on the receiver side as pduActivationRoutingGroups.

[TPS_SYST_02296] eventGroupControlType of a unicast Event ⌈If the Pdu represents an Event that is transmitted or received over unicast then the eventGroupControlType attribute of the PduActivationRoutingGroup that refers the Pdu shall be set to activationUnicast.⌋

[TPS_SYST_02297] eventGroupControlType of a multicast Event ⌈If the Pdu represents an Event that is transmitted or received over multicast then the eventGroupControlType attribute of the PduActivationRoutingGroup that refers the Pdu shall be set to activationMulticast.⌋

[TPS_SYST_02444] eventGroupControlType of a unicast and multicast Event ⌈If the Pdu represents an Event that is transmitted or received over unicast and multicast, then the eventGroupControlType attribute of the PduActivationRoutingGroup that refers to the Pdu shall be set to activationUnicastAndMulticast.⌋

[TPS_SYST_02298] eventGroupControlType of a unicast Field ⌈If the Pdu represents a Field Notifier that is transmitted or received over unicast then either:

- two PduActivationRoutingGroups shall be used where:
  - the eventGroupControlType attribute of the PduActivationRoutingGroup that refers the Pdu is set to activationUnicast.
  - the eventGroupControlType attribute of the PduActivationRoutingGroup that refers the Pdu is set to triggerUnicast.
- or one PduActivationRoutingGroup shall be used where the eventGroupControlType attribute of the PduActivationRoutingGroup that refers the Pdu shall be set to activationAndTriggerUnicast.

⌋

With this approach it is ensured that the current value of the Field is sent back immediately to the subscriber in an event-like notification pattern as soon as the subscription to the field becomes effective (TriggerUnicast). Additional update notifications will be sent to subscribers over IP Unicast whenever the value of the field gets updated (ActivationUnicast). Please note that the immediate transmission of the current value to the subscriber of the Field value is supported over IP Unicast only.

Apart of the immediate transmission (TriggerUnicast) the server may choose to transmit further notifications over IP Multicast in case that the multicastThreshold is set to a value >= 1. In this case a PduActivationRoutingGroup shall be used that has the eventGroupControlType attribute set to activationMulticast or activationUnicastAndMulticast.

The following listing shows an example for a ProvidedServiceInstance of the *RadarService*.

```
<PROVIDED-SERVICE-INSTANCE>
  <SHORT-NAME>RadarService_1</SHORT-NAME>
  <MAJOR-VERSION>1</MAJOR-VERSION>
  <METHOD-ACTIVATION-ROUTING-GROUPS>
    <!-- /ServiceInstancaCollectionSets/SomeIpServiceInstances/
       RadarService_1/RadarService_1_methodActivationGroup -->
    <PDU-ACTIVATION-ROUTING-GROUP>
      <SHORT-NAME>RadarService_1_methodActivationGroup</SHORT-NAME>
      <I-PDU-IDENTIFIER-UDP-REFS>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_Calibrate_Call</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_Calibrate_Return</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_Adjust_Call</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_Adjust_Return</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_UpdateRate_Getter_Call</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_UpdateRate_Getter_Return</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_UpdateRate_Setter_Call</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_UpdateRate_Setter_Return</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/
           RadarService_LogCurrentState</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
           SoConIpduIdentifierSet/RadarServicePduSet/RadarService_Reset</I-
           PDU-IDENTIFIER-UDP-REF>
      </I-PDU-IDENTIFIER-UDP-REFS>
    </PDU-ACTIVATION-ROUTING-GROUP>
  </METHOD-ACTIVATION-ROUTING-GROUPS>
  <EVENT-HANDLERS>
    <!-- /ServiceInstancaCollectionSets/SomeIpServiceInstances/
       RadarService_1/EventGroup_All -->
    <EVENT-HANDLER>
      <SHORT-NAME>EventGroup_All</SHORT-NAME>
      <EVENT-GROUP-IDENTIFIER>1</EVENT-GROUP-IDENTIFIER>
      <EVENT-MULTICAST-ADDRESSS>
        <APPLICATION-ENDPOINT-REF-CONDITIONAL>
```

```
      <APPLICATION-ENDPOINT-REF DEST="APPLICATION-ENDPOINT">/
          CommunicationClusters/EthernetCluster/Vlan73/
          MyEcuMulticastSocketAddress/MyEcuMulticastAep</APPLICATION-
          ENDPOINT-REF>
    </APPLICATION-ENDPOINT-REF-CONDITIONAL>
  </EVENT-MULTICAST-ADDRESSS>
  <MULTICAST-THRESHOLD>2</MULTICAST-THRESHOLD>
  <PDU-ACTIVATION-ROUTING-GROUPS>
    <!-- /ServiceInstancaCollectionSets/SomeIpServiceInstances/
       RadarService_1/EventGroup_All/
       RadarService_1_eventAndNotifierActivationUnicastGroup -->
    <PDU-ACTIVATION-ROUTING-GROUP>
      <SHORT-NAME>RadarService_1_eventAndNotifierActivationUnicastGroup
          </SHORT-NAME>
      <EVENT-GROUP-CONTROL-TYPE>ACTIVATION-UNICAST</EVENT-GROUP-CONTROL
          -TYPE>
      <I-PDU-IDENTIFIER-UDP-REFS>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
            SoConIpduIdentifierSet/RadarServicePduSet/
            RadarService_BrakeEvent</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
            SoConIpduIdentifierSet/RadarServicePduSet/
            RadarService_UpdateRate_Notifier</I-PDU-IDENTIFIER-UDP-REF>
      </I-PDU-IDENTIFIER-UDP-REFS>
    </PDU-ACTIVATION-ROUTING-GROUP>
    <!-- /ServiceInstancaCollectionSets/SomeIpServiceInstances/
       RadarService_1/EventGroup_All/RadarService_1_TriggerUnicastGroup
        -->
    <PDU-ACTIVATION-ROUTING-GROUP>
      <SHORT-NAME>RadarService_1_TriggerUnicastGroup</SHORT-NAME>
      <EVENT-GROUP-CONTROL-TYPE>TRIGGER-UNICAST</EVENT-GROUP-CONTROL-
          TYPE>
      <I-PDU-IDENTIFIER-UDP-REFS>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
            SoConIpduIdentifierSet/RadarServicePduSet/
            RadarService_UpdateRate_Notifier</I-PDU-IDENTIFIER-UDP-REF>
      </I-PDU-IDENTIFIER-UDP-REFS>
    </PDU-ACTIVATION-ROUTING-GROUP>
    <!-- /ServiceInstancaCollectionSets/SomeIpServiceInstances/
       RadarService_1/EventGroup_All/
       RadarService_1_eventAndNotifierActivationMulticastGroup -->
    <PDU-ACTIVATION-ROUTING-GROUP>
      <SHORT-NAME>
          RadarService_1_eventAndNotifierActivationMulticastGroup</SHORT
          -NAME>
      <EVENT-GROUP-CONTROL-TYPE>ACTIVATION-MULTICAST</EVENT-GROUP-
          CONTROL-TYPE>
      <I-PDU-IDENTIFIER-UDP-REFS>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
            SoConIpduIdentifierSet/RadarServicePduSet/
            RadarService_BrakeEvent</I-PDU-IDENTIFIER-UDP-REF>
        <I-PDU-IDENTIFIER-UDP-REF DEST="SO-CON-I-PDU-IDENTIFIER">/
            SoConIpduIdentifierSet/RadarServicePduSet/
            RadarService_UpdateRate_Notifier</I-PDU-IDENTIFIER-UDP-REF>
      </I-PDU-IDENTIFIER-UDP-REFS>
    </PDU-ACTIVATION-ROUTING-GROUP>
```

```
      </PDU-ACTIVATION-ROUTING-GROUPS>
    </EVENT-HANDLER>
  </EVENT-HANDLERS>
  <INSTANCE-IDENTIFIER>1</INSTANCE-IDENTIFIER>
  <LOCAL-UNICAST-ADDRESSS>
    <APPLICATION-ENDPOINT-REF-CONDITIONAL>
      <APPLICATION-ENDPOINT-REF DEST="APPLICATION-ENDPOINT">/
          CommunicationClusters/EthernetCluster/Vlan73/
          MyEcuUnicastSocketAddress/MyEcuUnicastAep</APPLICATION-ENDPOINT-
          REF>
    </APPLICATION-ENDPOINT-REF-CONDITIONAL>
  </LOCAL-UNICAST-ADDRESSS>
  <MINOR-VERSION>0</MINOR-VERSION>
  <SERVICE-IDENTIFIER>27</SERVICE-IDENTIFIER>
</PROVIDED-SERVICE-INSTANCE>
```

Listing 10.4: Example for a `ProvidedServiceInstance`

11 Software Cluster

11.1 Big Picture

Among the different architecture approaches implemented in automotive software solutions, the so-called *domain architecture* is characterized by the existence of so-called *domain controllers*[1] that act as a “manager” for a specific domain (e.g. powertrain, body, chassis) in automotive software.

Domain controllers are connected to each other via a backbone communication bus and also typically utilize various communication buses to communicate with domain-specific controller ECUs. The concept of *domain controllers* is sketched in Figure 11.1.

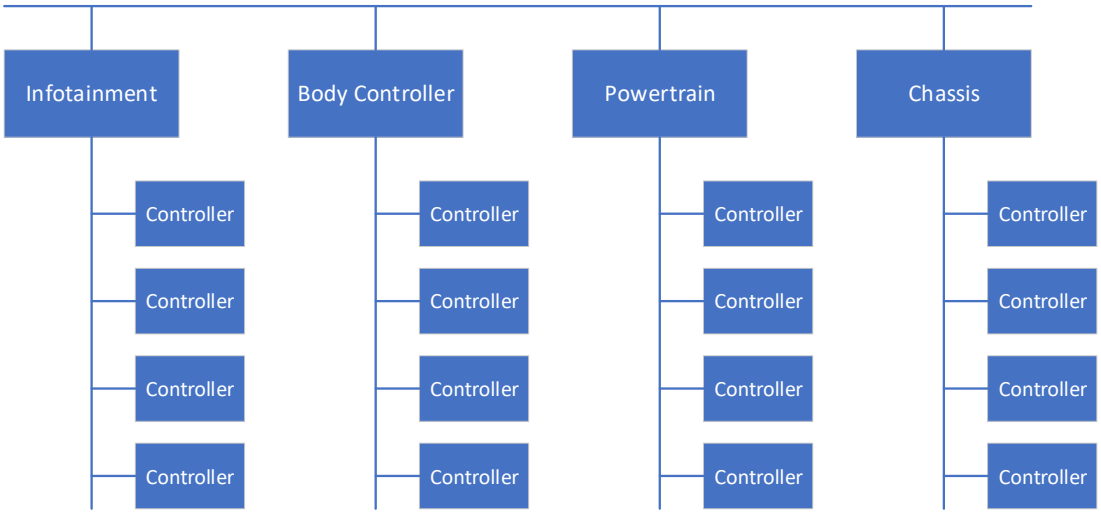

Figure 11.1: Sketch of an automotive *domain controller* architecture

A consequence of the *domain controller* concept is that the individual *domain controller* ECUs become very complex and typically host several thousand software-components. As far as the methodology on the *AUTOSAR classic platform* is concerned[2], the ECU needs to be fully integrated **before** the generation of the RTE can be started.

The integration involves not only the consistent configuration of all basic software modules but also requires the creation of connections between the application software and the service software-components that represent the service layer of the AUTOSAR basic software.

This leads to the following list of issues:

- Generation of the RTE takes a long time.

[1] This term is inspired by the existence of functional domains inside a typical vehicle, not to be confused with the definition of the term (“central authentication server”) in general computer science.

[2] The workflow on the *AUTOSAR classic platform* is centered around the idea to build a complete executable image for each ECU in one step

- Single failures can block the entire integration process.
- Build times become very high.

Please note that similar problems occur in any kind of software architecture which centralizes software functionality in specific ECUs, e.g. zone Controllers.

In response to the increase in complexity, integration time, possible mistakes, and build time, the software on such complex ECUs can be structurally decomposed into so-called `Software Cluster`s of arbitrary complexity.

A `Software Cluster` can be developed, integrated against, and built independently of the surrounding host system on the enclosing ECU. After completion of the development workflow, a `Software Cluster` can be deployed to its host ECU independently, i.e. the deployment of such a `Software Cluster` does not require the entire ECU to be re-programmed.

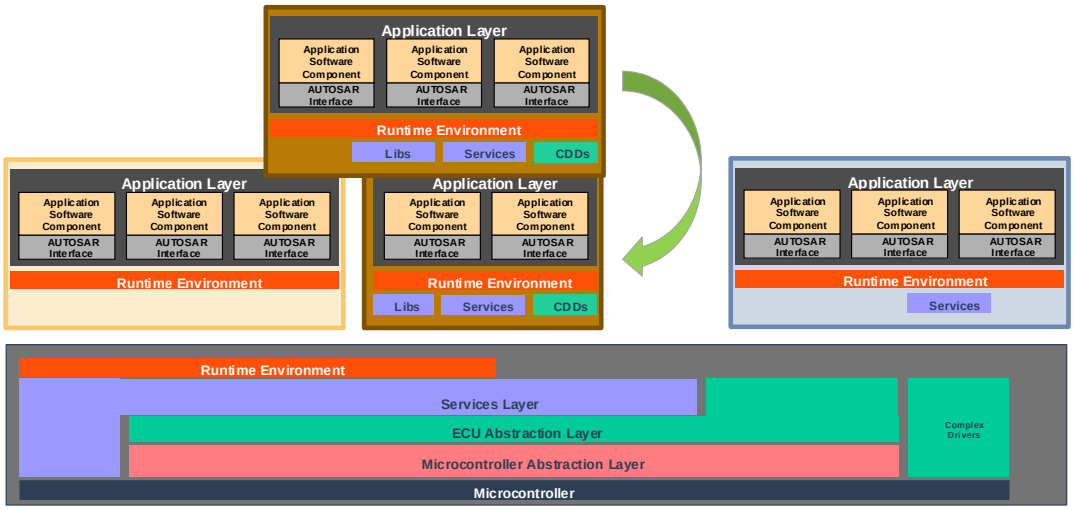

Figure 11.2: Replace an existing `Software Cluster` without reprogramming the entire ECU

In other words, the rest of the software on the respective ECU can be left unchanged and is kept binary identical if a given `Software Cluster` is flashed onto the respective ECU.

[TPS_SYST_02315] Definition of a software cluster on the *AUTOSAR classic platform* ⌈On the *AUTOSAR classic platform*, a `Software Cluster` is represented by meta-class `CpSoftwareCluster`. A `CpSoftwareCluster` is defined by references to a collection of either

- `CompositionSwComponentType` in the role `swComposition`, in this case the `CpSoftwareCluster` is described as a re-usable asset out of the context of a concrete `System`.

- SwComponentPrototype in the role swComponent, in the case the CpSoftwareCluster is defined in the context of a System.

⌋

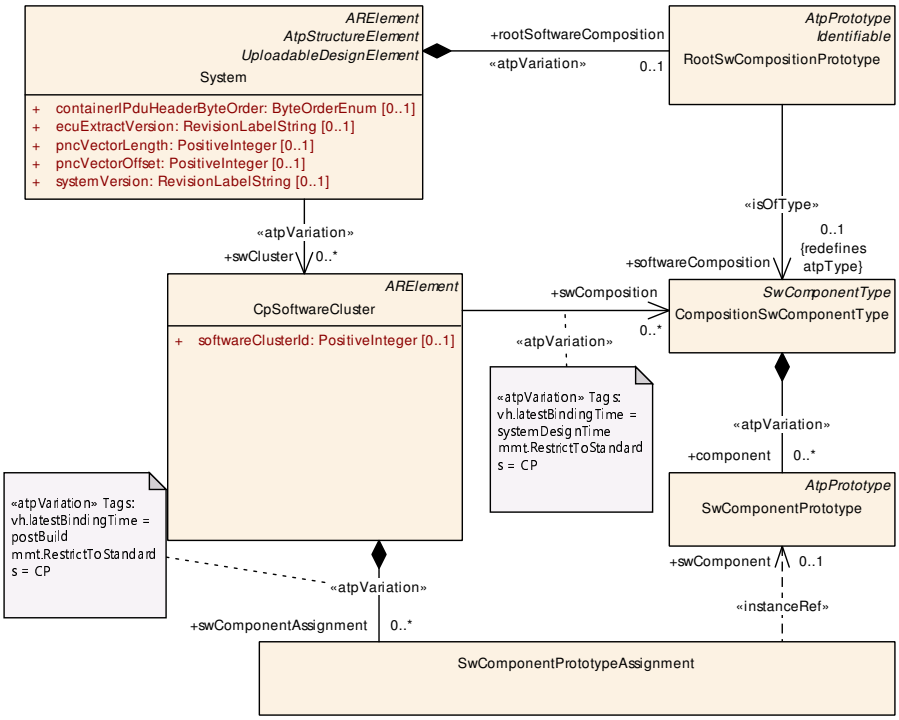

Figure 11.3: Modeling of the CpSoftwareCluster

[TPS_SYST_02316] Semantics of meta-class SwComponentPrototype-Assignment ⌈Meta-class SwComponentPrototypeAssignment as well as its aggregation at CpSoftwareCluster in the role swComponentAssignment supports the definition of a variation point in the relation between CpSoftwareCluster and SwComponentPrototype.⌋

Class	**CpSoftwareCluster**			
Note	This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings. **Tags:** atp.recommendedPackage=CpSoftwareClusters This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster.

▽

△

Class	**CpSoftwareCluster**			
swComponent Assignment	SwComponent PrototypeAssignment	*	aggr	This is the collection of SwComponentPrototype Assignments **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentAssignment, swComponent Assignment.variationPoint.shortLabel vh.latestBindingTime=postBuild
swComposition	CompositionSw ComponentType	*	ref	Software Components in the context of a CompositionSw ComponentType belonging to this CP Software Cluster. This reference can be used to describe the belonging SWCs when the CP Software Cluster is described out of the context of a System, e.g. reusable CP Software Cluster. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComposition, swComposition.composition SwComponentType, swComposition.variationPoint.short Label vh.latestBindingTime=systemDesignTime

Table 11.1: CpSoftwareCluster

Class	**SwComponentPrototypeAssignment**			
Note	This meta-class is only required to allow for the variant modeling of an instanceRef. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	CpSoftwareCluster.swComponentAssignment			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
swComponent	SwComponent Prototype	0..1	iref	hierarchical tree(s) of Software Components belonging to this CP Software Cluster. This reference is used to describe the belonging SWCs if the CP Software Cluster is described in the context of a System, **Tags:** atp.Splitkey=swComponent, swComponent.context Composition, swComponent.contextComponent, sw Component.targetComponent **InstanceRef implemented by:** ComponentInSystem InstanceRef

Table 11.2: SwComponentPrototypeAssignment

[TPS_SYST_02317] References from `CpSoftwareCluster` to `Composition-SwComponentType` and `SwComponentPrototype` ⌈The usage of the reference `CpSoftwareCluster.swComposition` shall eventually be replaced by the `swComponentAssignment` aggregation when the `SwComponentPrototype`s referenced by the `CpSoftwareCluster` (via the role `swComponentAssignment`) are integrated into a concrete Ecu Extract.⌋

[constr_5327] Existence of attribute `CpSoftwareCluster.category`

Imposition time: CP: `IT_SwCluSysDesc`

⌈For each `CpSoftwareCluster`, attribute `category` shall exist.⌋

[constr_5335] `CpSoftwareCluster.softwareClusterId` shall be unique in the scope of an `EcuInstance`

Imposition time: CP: `IT_SwCluSysDesc`

⌈The `softwareClusterId` shall be unique for each `CpSoftwareCluster` that is mapped to the same `EcuInstance` with the `CpSoftwareClusterToEcuInstanceMapping`.⌋

[constr_5336] Existence of `CpSoftwareCluster.softwareClusterId`

Imposition time: CP: `IT_SwCluSysDesc`

⌈For each `CpSoftwareCluster`, attribute `softwareClusterId` shall exist.⌋

Class	**CompositionSwComponentType**			
Note	A `CompositionSwComponentType` aggregates `SwComponentPrototype`s (that in turn are typed by `SwComponentType`)s as well as `SwConnector`s for primarily connecting `SwComponentPrototype`s among each others and towards the surface of the `CompositionSwComponentType`. By this means, a hierarchical structures of software-components can be created. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
component	SwComponent Prototype	*	aggr	The instantiated components that are part of this composition. The aggregation of `SwComponentPrototype` is subject to variability with the purpose to support the conditional existence of a `SwComponentPrototype`. Please be aware: if the conditional existence of `SwComponentPrototype`s is resolved post-build, the deselected `SwComponentPrototype`s are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE. The use case for having 0 components owned by the `CompositionSwComponentType` could be to deliver an empty `CompositionSwComponentType` to e.g. a supplier for filling the internal structure. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=component, component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	SwConnector	*	aggr	`SwConnector`s have the principal ability to establish a connection among `PortPrototype`s. They can have many roles in the context of a `CompositionSwComponentType`. Details are refined by subclasses. The aggregation of `SwConnector`s is subject to variability with the purpose to support variant data flow. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	**CompositionSwComponentType**			
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the `ConstantSpecificationMapping` to be applied for initValues of `PPortComSpec`s and `RPortComSpec`. **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the `DataTypeMappingSet` to be applied for the used `ApplicationDataType`s in `PortInterface`s. Background: when developing subsystems it may happen that `ApplicationDataType`s are used on the surface of `CompositionSwComponentType`s. In this case it would be reasonable to be able to also provide the intended mapping to the `ImplementationDataType`s. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the `CompositionSwComponentType`s. Rationale: if the mapping of `ApplicationDataType`s on the delegated and inner `PortPrototype` matches then the mapping to `ImplementationDataType`s is not impacting compatibility. **Tags:** atp.Splitkey=dataTypeMapping
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationRTEEventProps, instantiation RTEEventProps.shortLabel, instantiationRTEEvent Props.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime This Attribute is only used by the AUTOSAR Classic Platform.
physical Dimension Mapping	PhysicalDimension MappingSet	0..1	ref	This reference identifies the `PhysicalDimensionMappingSet` that is applicable in the context of the enclosing `CompositionSwComponentType`. The `PhysicalDimensionMapping`s contained in the `PhysicalDimensionMappingSet` shall be taken into account for the assessment of the compatibility of `PhysicalDimension`s in the context of creation of a `PortInterfaceMapping` in the scope of the `CompositionSwComponentType`. **Tags:** atp.Splitkey=physicalDimensionMapping

Table 11.3: CompositionSwComponentType

Class	**SwComponentPrototype**			
Note	Role of a software component within a composition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.component			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
type	SwComponentType	0..1	tref	Type of the instance. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table 11.4: SwComponentPrototype

[TPS_SYST_02318] Membership in `System` ⌈The membership of a given `CpSoftwareCluster` can be formalized by means of the reference `System.swCluster`.⌋

[TPS_SYST_02319] Semantics of attribute `CpSoftwareCluster.category` ⌈The following values for attribute `CpSoftwareCluster.category` are standardized by AUTOSAR:

- **`HOST_SOFTWARE_CLUSTER`**: the `CpSoftwareCluster` that contains the major part of the basic-software stack, especially micro-controller-dependent modules including the operating system.
- **`APPLICATION_SOFTWARE_CLUSTER`**: the `CpSoftwareCluster` represents application-level functionality conceptually located on top of the `CpSoftwareCluster` of `category` `HOST_SOFTWARE_CLUSTER`.

⌋

[constr_5176] Existence of `CpSoftwareCluster` of `category` `HOST_SOFTWARE_CLUSTER` on one `EcuInstance`

Imposition time: CP: `IT_SwCluSysDesc`

⌈On each `EcuInstance`, exactly one `CpSoftwareCluster` of `category` `HOST_SOFTWARE_CLUSTER` shall exist.⌋

[constr_5177] Validity of reference `CpSoftwareClusterToEcuInstanceMapping.swCluster`

Imposition time: CP: `IT_SwCluSysDesc`

⌈A `CpSoftwareClusterToEcuInstanceMapping` that references a given `CpSoftwareCluster` in the role `CpSoftwareClusterToEcuInstanceMapping.swCluster` shall be aggregated by the same `System` (in the role `System.mapping.swMapping`) that also refers to the referenced `CpSoftwareCluster` in the role `System.swCluster`.⌋

The interaction of `Software Cluster`s with the service layer of the basic software stack requires the definition of `Software Cluster Resource`s (see Figure 11.4). `Software Cluster`s declare required and provided `Software Cluster Resource`s that can be satisfied from a central resource pool.

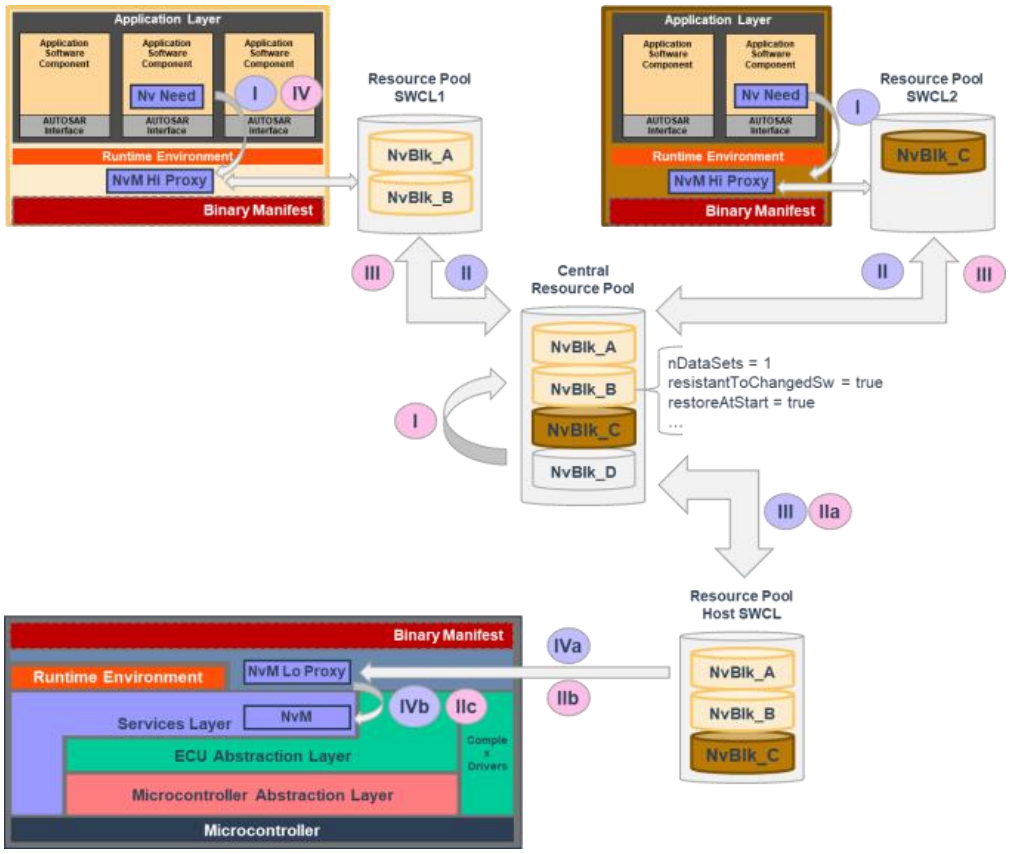

Figure 11.4: Example usage of `Software Cluster Resource`s

Details regarding the `Software Cluster Resource` are described in section 11.2.

There are in principle two different approaches to connect `Software Cluster`s with each other:

Off-board In this case the connection is created by a software tool on the basis of the binary image of the `Software Cluster` and modeled information that formally describes the so-called `Software Cluster Binary Manifest`. The content of the model of the `Software Cluster Binary Manifest` represents meta-data of the actual connection information. The main benefit of this approach is that the meta-data does not have to be stored on-board, i.e. on the target Ecu.

On-board In this case the connection is created by software running on the target Ecu during the reprogramming phase on the basis of meta-data of the connection endpoints (semantically identical to the content of the model of the respective `Software Cluster Binary Manifest`). The connection data is stored in modifiable memory. The meta-data content is stored on device, and the required memory markup for this aspect needs to be taken into account for Ecu design.

It is important to understand that the Software Cluster and the corresponding Software Cluster Binary Manifest are associated with different steps in the development workflow. A Software Cluster represents a design model element while the Software Cluster Binary Manifest is a derived information used to support a downstream integration phase for defining an off-board connection.

This relation leads to modeling decisions, such that elements of the Software Cluster Binary Manifest can only reference elements of the Software Cluster, but can't be aggregated by them.

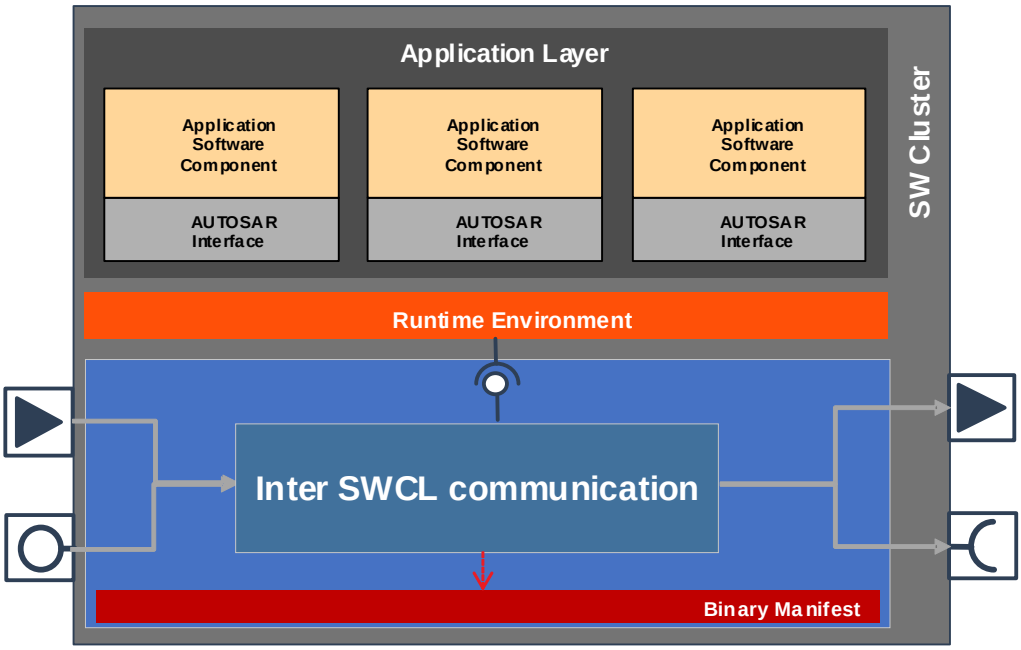

Figure 11.5: Software Cluster Binary Manifest of a single Software Cluster

The Software Cluster Binary Manifest is described in more detail in section 11.3.

The mapping of Software Clusters to EcuInstance and Application-Partition is described in section 5.6.

11.2 Software Cluster Resources

A CpSoftwareCluster is able to provide CpSoftwareClusterResources that will be accessed by other CpSoftwareClusters. At the same time a CpSoftware-Cluster may require CpSoftwareClusterResources from other CpSoftware-Clusters to operate the software that belongs to the CpSoftwareCluster.

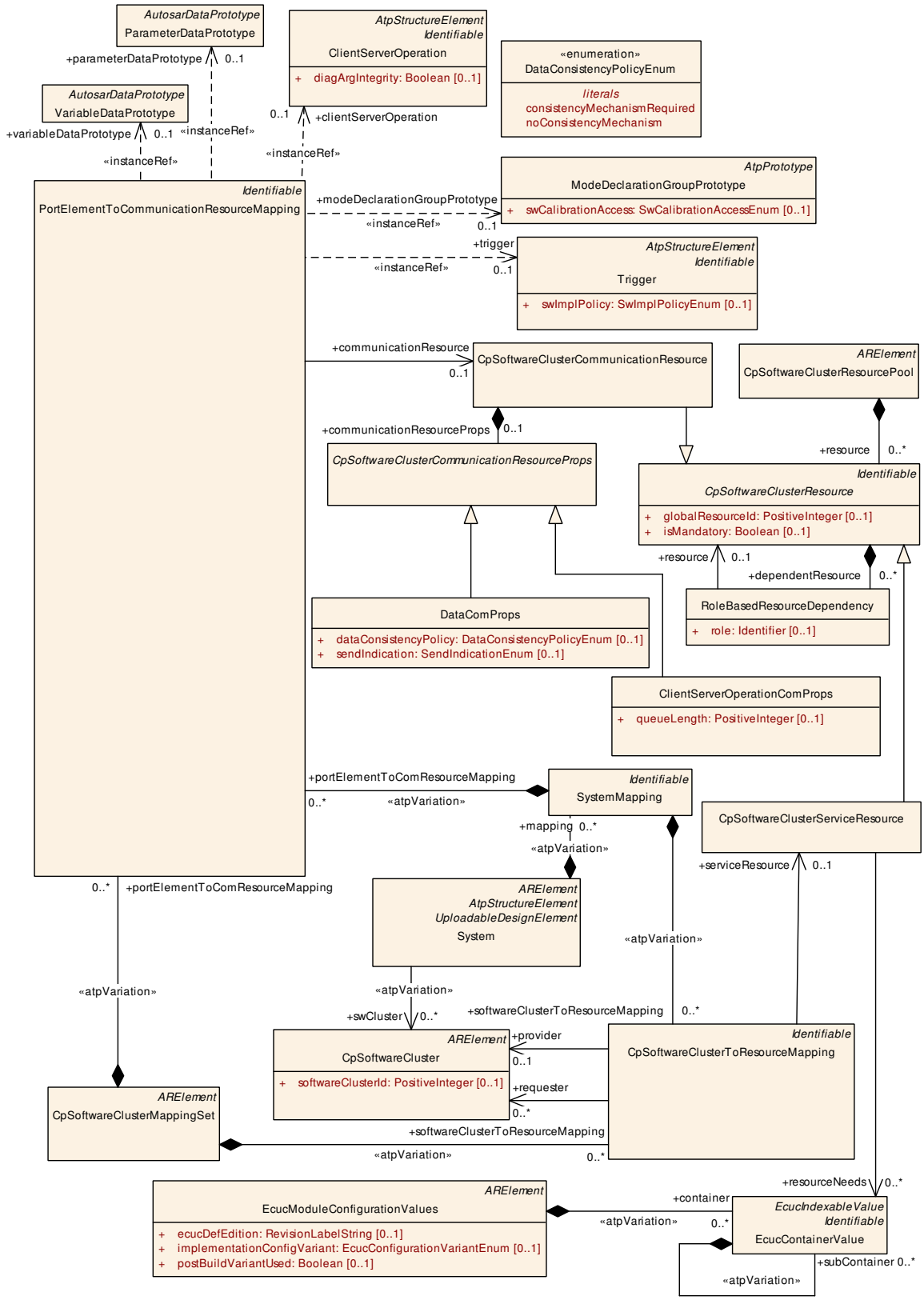

Figure 11.6: Software Cluster Resources

[TPS_SYST_02320] Kinds of CpSoftwareClusterResources ⌈There are two kinds of CpSoftwareClusterResources:

- CpSoftwareClusterCommunicationResource that relates to a port based communication on VFB level, for instance to a sender-receiver communication or client-server communication,
- CpSoftwareClusterServiceResource that relates to the Basic Software, for instance to a NvBlock.

⌋

Class	**CpSoftwareClusterResourcePool**			
Note	Represents the pool of resources which can be provided or required by CP Software Clusters. **Tags:** atp.recommendedPackage=CpSoftwareClusterResourcePools This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecuScope	EcuInstance	*	ref	This reference identifies the EcuInstance in which the resource pool is defined. **Tags:** atp.Splitkey=ecuScope
resource	CpSoftwareCluster Resource	*	aggr	This aggregation represents the collection of resources in the enclosing resource pool. **Tags:** atp.Splitkey=resource, resource.shortName

Table 11.5: CpSoftwareClusterResourcePool

Class	***CpSoftwareClusterResource*** (abstract)			
Note	Represents a single resource required or provided by a CP Software Cluster. **Tags:** atp.recommendedPackage=Resources This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CpSoftwareClusterCommunicationResource, CpSoftwareClusterServiceResource			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dependent Resource	RoleBasedResource Dependency	*	aggr	Link to a resource which depends on this resource to implement them. **Tags:** atp.Splitkey=dependentResource
globalResource Id	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines.
isMandatory	Boolean	0..1	attr	This attribute indicates, that the resource is mandatory to operate the Software Cluster. If the resource is not provided on the machine the connection process of any Software Cluster requiring this resource gets aborted.

Table 11.6: CpSoftwareClusterResource

Class	RoleBasedResourceDependency			
Note	This class specifies a dependency between CpSoftwareClusterResources. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	*CpSoftwareClusterResource*.dependentResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
resource	CpSoftwareCluster Resource	0..1	ref	Reference to resource for which the dependency is depicted. **Tags:** atp.Splitkey=resource
role	Identifier	0..1	attr	This is attributes characterizes the kind of dependency

Table 11.7: RoleBasedResourceDependency

[constr_5178] Existence of attribute CpSoftwareClusterResource.globalResourceId

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterResource, attribute globalResourceId shall exist.⌋

[constr_5179] Existence of attribute CpSoftwareClusterResource.isMandatory

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterResource, attribute isMandatory shall exist.⌋

[constr_5180] Allowed values for CpSoftwareClusterResource.globalResourceId

Imposition time: CP: IT_ResPool

⌈Attribute CpSoftwareClusterResource.globalResourceId shall not be set to 0.⌋

For explanation of [constr_5180], the value 0 is reserved to mark an invalid ID value.

Class	CpSoftwareClusterCommunicationResource			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication ResourceProps	CpSoftwareCluster Communication ResourceProps	0..1	aggr	This aggregation supports the further qualification of the enclosing CpSoftwareClusterCommunicationRecource by means of additional attributes depending on the nature of the CpSoftwareClusterCommunicationRecource. **Tags:** atp.Splitkey=communicationResourceProps

Table 11.8: CpSoftwareClusterCommunicationResource

Class	***CpSoftwareClusterCommunicationResourceProps*** (abstract)			
Note	Communication properties for cross cluster communication. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	ClientServerOperationComProps, DataComProps			
Aggregated by	CpSoftwareClusterCommunicationResource.communicationResourceProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 11.9: CpSoftwareClusterCommunicationResourceProps

Class	**DataComProps**			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterCommunicationResourceProps*			
Aggregated by	CpSoftwareClusterCommunicationResource.communicationResourceProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
data Consistency Policy	DataConsistencyPolicy Enum	0..1	attr	This attribute defines requirements on the data consistency mechanism in the cross cluster communication. If the attribute is not set, the default value consistencyMechanismRequired applies.
sendIndication	SendIndicationEnum	0..1	attr	Send indication behavior for last-is-the best data communication.

Table 11.10: DataComProps

Enumeration	**DataConsistencyPolicyEnum**
Note	Defines how data consistency is ensured in the cross cluster communication.
Aggregated by	DataComProps.dataConsistencyPolicy
Literal	***Description***
consistency Mechanism Required	In this case the data consistency is ensured by the implementation of the SwClucC module. **Tags:** atp.EnumerationLiteralIndex=0
noConsistency Mechanism	In this case the data consistency is not ensured by the SwClucC module. In this case it has to be ensured by scheduling. **Tags:** atp.EnumerationLiteralIndex=1

Table 11.11: DataConsistencyPolicyEnum

Class	**ClientServerOperationComProps**			
Note	Defines additional attributes for the implementation of Client Server communication between software clusters This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterCommunicationResourceProps*			
Aggregated by	CpSoftwareClusterCommunicationResource.communicationResourceProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ClientServerOperationComProps**			
queueLength	PositiveInteger	0..1	attr	Length of call request queue on the server side. The queue is implemented by the SwCluC. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.

Table 11.12: ClientServerOperationComProps

Enumeration	**SendIndicationEnum**
Note	This meta-class provides a way to specify in which way redundancy shall be applied on collection level. This Enumeration is only used by the AUTOSAR Classic Platform.
Aggregated by	DataComProps.sendIndication
Literal	*Description*
anySendOperation	This value represents the requirement that any send operation of the Software Cluster is indicated. **Tags:** atp.EnumerationLiteralIndex=2
none	This value represents the requirement that send operations of the Software Cluster are not indicated. **Tags:** atp.EnumerationLiteralIndex=1

Table 11.13: SendIndicationEnum

Class	**CpSoftwareClusterServiceResource**			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the BSW. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
resourceNeeds	EcucContainerValue	*	ref	Refernce(s) to one or multiple EcucContainerValue(s) qualifying the characteristics of the resource. **Tags:** atp.Splitkey=resourceNeeds

Table 11.14: CpSoftwareClusterServiceResource

[constr_5181] Existence of attribute CpSoftwareClusterService-Resource.category

Imposition time: CP: IT_ResPool

⌈For each CpSoftwareClusterServiceResource, attribute category shall exist.⌋

The applicable values of CpSoftwareClusterServiceResource.category are defined in the document "Software Cluster Connection".

CpSoftwareClusterResources are collected in CpSoftwareCluster-ResourcePools and are assigned to a CpSoftwareCluster in different ways as defined by the following specification items:

[TPS_SYST_02321] Assignment of CpSoftwareClusterCommunicationResources to CpSoftwareClusters in the context of a SwComponentPrototype ⌈In case that a SwComponentPrototype is defined in the context of a System and this SwComponentPrototype is assigned to the CpSoftwareCluster with the CpSoftwareCluster.swComponentAssignment and the SwComponentPrototype defines an interface of the CpSoftwareCluster then the PortElementToCommunicationResourceMapping aggregated by a SystemMapping maps the CpSoftwareClusterCommunicationResource to an element of a PortInterface used in the context of a PortPrototype in the SwComponentPrototype.

If the PortPrototype of the SwComponentPrototype is a PPortPrototype then the CpSoftwareClusterCommunicationResource is provided by the CpSoftwareCluster.

If the PortPrototype of the SwComponentPrototype is a RPortPrototype then the CpSoftwareClusterCommunicationResource is required by the CpSoftwareCluster.⌋

[TPS_SYST_02345] Assignment of CpSoftwareClusterCommunicationResources to CpSoftwareClusters in the context of a SwComponentType ⌈In case that a CompositionSwComponentType is defined and is assigned to the CpSoftwareCluster with the CpSoftwareCluster.swComposition and this SwComponentType defines an interface of the CpSoftwareCluster then the PortElementToCommunicationResourceMapping that is aggregated by a CpSoftwareClusterMappingSet maps the CpSoftwareClusterCommunicationResource to an element of a PortInterface used in the context of a PortPrototype of a CompositionSwComponentType.

If the PortPrototype of the CompositionSwComponentType is a PPortPrototype then the CpSoftwareClusterCommunicationResource is provided by the CpSoftwareCluster.

If the PortPrototype of the CompositionSwComponentType is a RPortPrototype then the CpSoftwareClusterCommunicationResource is required by the CpSoftwareCluster.⌋

Class	**PortElementToCommunicationResourceMapping**			
Note	This meta class maps a communication resource to CP Software Clusters. In this case the kind of Port Prototype specified whether the Software Cluster has to provide or to require the resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.portElementToComResourceMapping, SystemMapping.portElementToComResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**PortElementToCommunicationResourceMapping**			
clientServer Operation	ClientServerOperation	0..1	iref	ClientServerOperation instance qualifying the communication resource **Tags:** atp.Splitkey=clientServerOperation, clientServer Operation.contextComposition, clientServer Operation.contextComponent, clientServer Operation.contextPort, clientServerOperation.target Operation **InstanceRef implemented by:** OperationInSystem InstanceRef
communication Resource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:** atp.Splitkey=communicationResource
mode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	iref	ModeDeclarationGroupPrototype instance qualifying the communication resource **Tags:** atp.Splitkey=modeDeclarationGroupPrototype, modeDeclarationGroupPrototype.contextComposition, modeDeclarationGroupPrototype.contextComponent, modeDeclarationGroupPrototype.contextPort, mode DeclarationGroupPrototype.targetModeDeclarationGroup Prototype **InstanceRef implemented by:** ModeDeclarationGroup PrototypeInSystemInstanceRef
parameterData Prototype	ParameterData Prototype	0..1	iref	ParameterDataPrototype instance qualifying the communication resource. **Tags:** atp.Splitkey=parameterDataPrototype, parameter DataPrototype.contextComposition, parameterData Prototype.contextComponent, parameterData Prototype.contextPort, parameterDataPrototype.target ParameterDataPrototype **InstanceRef implemented by:** ParameterDataPrototype InSystemInstanceRef
trigger	Trigger	0..1	iref	Trigger instance qualifying the communication resource. **Tags:** atp.Splitkey=trigger, trigger.contextComposition, trigger.contextComponent, trigger.contextPort, trigger.targetTrigger **InstanceRef implemented by:** TriggerInSystemInstance Ref
variableData Prototype	VariableDataPrototype	0..1	iref	VariableDataPrototype instance qualifying the communication resource **Tags:** atp.Splitkey=variableDataPrototype, variableData Prototype.contextComponent, variableData Prototype.contextComposition, variableData Prototype.contextPort, variableDataPrototype.targetData Prototype **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef

Table 11.15: PortElementToCommunicationResourceMapping

Please note that the assignment of `CpSoftwareClusterCommunicationResource`s to `CpSoftwareCluster`s in the context of a `SwComponentType`s as described by [TPS_SYST_02345] shall only be used in an early stage of the Methodology if the `System` with the `RootSwCompositionPrototype` and all included `SystemMapping`s is not available yet.

In this case not all context references in the «instanceRef» of `PortElementToCommunicationResourceMapping` can be used since the `SwComponentPrototype` is not available yet.

[constr_9407] Standardized values for the attribute category of meta-class PortElementToCommunicationResourceMapping

Imposition time: CP: IT_SwCluSysDesc

⌈The following values of the attribute category of meta-class PortElementToCommunicationResourceMapping are reserved by the AUTOSAR standard:

- CLIENT_SERVER_REQUEST: This represents the usage of the PortElementToCommunicationResourceMapping for the purpose to assign an communicationResource to a clientServerOperation used for the request communication path.
- CLIENT_SERVER_RESPONSE: This represents the usage of the PortElementToCommunicationResourceMapping for the purpose to assign a communicationResource to a clientServerOperation used for the response communication path.

⌋

[TPS_SYST_02322] PortElementToCommunicationResourceMapping aggregated by SystemMapping supersedes PortElementToCommunicationResourceMapping aggregated by CpSoftwareClusterMappingSet ⌈If a PortElementToCommunicationResourceMapping that is aggregated by the SystemMapping and a PortElementToCommunicationResourceMapping that is aggregated by the CpSoftwareClusterMappingSet exist at the same time and both reference the same element in the same PortPrototype then the PortElementToCommunicationResourceMapping that is aggregated by the SystemMapping supersedes the PortElementToCommunicationResourceMapping that is aggregated by the CpSoftwareClusterMappingSet.⌋

[constr_5329] SynchronousServerCallPoints for cross cluster communication are not supported

Imposition time: CP: IT_SwCluSysDesc

⌈A ClientServerOperation in the context of PortPrototype which is referenced by a PortElementToCommunicationResourceMapping in the role clientServerOperation is not allowed

- to be referenced by a SynchronousServerCallPoint.operation or
- to be connected to another ClientServerOperation in the context of a PortPrototype that in turn is referenced by SynchronousServerCallPoint.operation

⌋

[constr_5182] PRPortPrototypes are excluded as CpSoftwareCluster interfaces

Imposition time: CP: IT_SwCluSysDesc

⌈A CpSoftwareClusterCommunicationResource is not allowed to be mapped by a PortElementToCommunicationResourceMapping to an element of a PortInterface in the context of a PRPortPrototype.⌋

Please note that it is allowed that a PPortPrototype that is referenced by a PortElementToCommunicationResourceMapping is allowed to be connected via a DelegationSwConnector to a PRPortPrototype.

[constr_5183] PortElementToCommunicationResourceMapping shall reference exactly one element of a PortInterface

Imposition time: CP: IT_SwCluSysDesc

⌈For any given PortElementToCommunicationResourceMapping, either the reference

- parameterDataPrototype or
- modeDeclarationGroupPrototype or
- trigger or
- clientServerOperation or
- variableDataPrototype

shall exist.⌋

[TPS_SYST_02323] Assignment of CpSoftwareClusterServiceResources to CpSoftwareClusters ⌈A CpSoftwareClusterServiceResource is mapped to CpSoftwareCluster with a CpSoftwareClusterToResourceMapping.⌋

Class	**CpSoftwareClusterToResourceMapping**			
Note	This meta class maps a service resource to CP Software Clusters. By this mapping it's specified whether the Software Cluster has to provide or to require the resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.softwareClusterToResourceMapping, SystemMapping.softwareClusterToResourceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provider	CpSoftwareCluster	0..1	ref	CP Software Cluster providing the resource **Tags:** atp.Splitkey=provider
requester	CpSoftwareCluster	*	ref	CP Software Cluster requesting the resource **Tags:** atp.Splitkey=requester
service Resource	CpSoftwareCluster ServiceResource	0..1	ref	Service resource for which the mapping applies. **Tags:** atp.Splitkey=serviceResource

Table 11.16: CpSoftwareClusterToResourceMapping

[TPS_SYST_02324] **CpSoftwareClusterServiceResource provided by the CpSoftwareCluster** ⌈A CpSoftwareClusterResource is provided by a CpSoftwareCluster if a CpSoftwareClusterToResourceMapping exists that

- references the CpSoftwareClusterResource in the role serviceResource and
- references the CpSoftwareCluster in the role provider

⌋

[TPS_SYST_02325] **CpSoftwareClusterServiceResource required by the CpSoftwareCluster** ⌈A CpSoftwareClusterResource is required by a CpSoftwareCluster if a CpSoftwareClusterToResourceMapping exists that

- references the CpSoftwareClusterResource in the role serviceResource and
- references the CpSoftwareCluster in the role requester.

⌋

[constr_5184] **CpSoftwareClusterServiceResource can be provided only once on an EcuInstance**

Imposition time: CP: IT_SwCluSysDesc

⌈A CpSoftwareClusterServiceResource shall not be mapped by several CpSoftwareClusterToResourceMappings to CpSoftwareClusters in the provider role if the CpSoftwareClusters are mapped to the same EcuInstance by CpSoftwareClusterToEcuInstanceMappings.⌋

[TPS_SYST_02326] **Aggregation possibilities of CpSoftwareClusterToResourceMapping** ⌈The CpSoftwareClusterToResourceMapping can be aggregated by CpSoftwareClusterMappingSet and by SystemMapping.

- CpSoftwareClusterMappingSet.softwareClusterToResourceMapping can be used in an early stage of the Methodology if the System with the RootSwCompositionPrototype and all included SystemMappings is not available yet.
- SystemMapping.softwareClusterToResourceMapping can be used in a later stage of Methodology if the System with the RootSwCompositionPrototype and all included SystemMappings is available.

⌋

[TPS_SYST_02346] `CpSoftwareClusterToResourceMapping` aggregated by `SystemMapping` supersedes `CpSoftwareClusterToResourceMapping` aggregated by `CpSoftwareClusterMappingSet` ⌈If a `CpSoftwareClusterToResourceMapping` that is aggregated by the `SystemMapping` and a `CpSoftwareClusterToResourceMapping` that is aggregated by the `CpSoftwareClusterMappingSet` exist at the same time and both are mapping the same `CpSoftwareClusterServiceResource` to the same `CpSoftwareCluster` then the `CpSoftwareClusterToResourceMapping` that is aggregated by the `SystemMapping` supersedes the `CpSoftwareClusterToResourceMapping` that is aggregated by the `CpSoftwareClusterMappingSet`.⌋

[constr_5360] Cross cluster communication involving `NvBlockSwComponentType` is not supported

Imposition time: CP: `IT_SwCluSysDesc`

⌈A `PortElementToCommunicationResourceMapping` that is referencing a `CpSoftwareClusterCommunicationResource` in the role `communicationResource` is not allowed to reference:

- a `VariableDataPrototype` in the role `variableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `NvDataInterface` or
- a `VariableDataPrototype` in the role `variableDataPrototype` which is connected to another `VariableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` and typed by a `NvDataInterface` or
- a `ClientServerOperation` in the role `clientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface` or
- a `ClientServerOperation` in the role `clientServerOperation` which is connected to another `ClientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface`

⌋

11.3 Software Cluster Binary Manifest

[TPS_SYST_02327] Role of the `Software Cluster Binary Manifest` ⌈The modeling of the `Software Cluster Binary Manifest` allows for the formal definition of how a given `Software Cluster` interacts with the environment. This formal model is **only relevant for the creation of an off-board connection** between `Software Cluster`s.

The core characteristics of the model of a Software Cluster Binary Manifest are:

- The model of a Software Cluster Binary Manifest is created during the Software Cluster's build because it has to store information (e.g. data and function addresses, ID values for the interaction with basic software services) that only become available during build time.
- The model of a Software Cluster Binary Manifest uses guarding information (e.g. hash values) to ensure that only compatible resources are connected to each other.

The Software Cluster Binary Manifest is defined on integration level[3].⌋

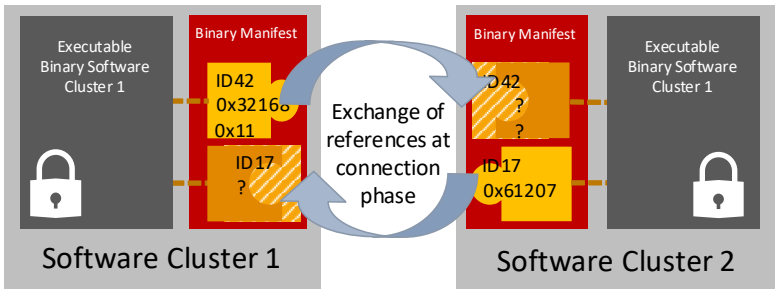

Figure 11.7: Two Software Cluster exchange information in their Software Cluster Binary Manifests.

This means that, at run-time, the Software Cluster Binary Manifest consists of data structures that are partly non-modifiable (to define e.g. meta-data or describe the access to a resource) and partly modifiable (to store the connection information, see Figure 11.7).

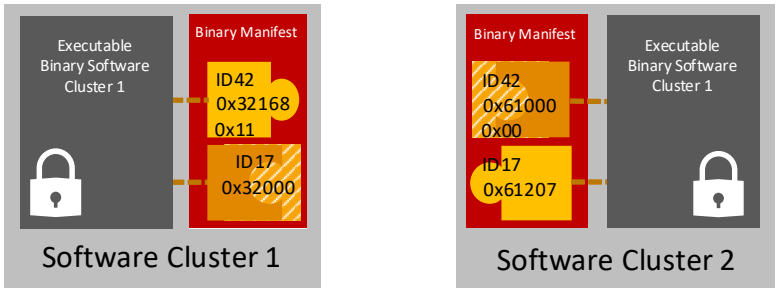

Figure 11.8: Two Software Cluster and their fully-configured Software Cluster Binary Manifests

The missing pieces in the Software Cluster Binary Manifest can be filled by exchanging information at either configuration-time (off-board) or run-time (on-board),

[3]As opposed to design level, on which the actual CpSoftwareCluster exists

see Figure 11.8. The algorithm used to create an off-board connection is identical to the algorithm used to create an on-board connection.

The difference ist mainly that the creation of the on-board connection needs the information stored in the model of a `Software Cluster Binary Manifest` on-board, i.e. a significant amount of on-board storage is required for holding the connection meta-data.

The logical structure of the `Software Cluster Binary Manifest` is depicted in Figure 11.9

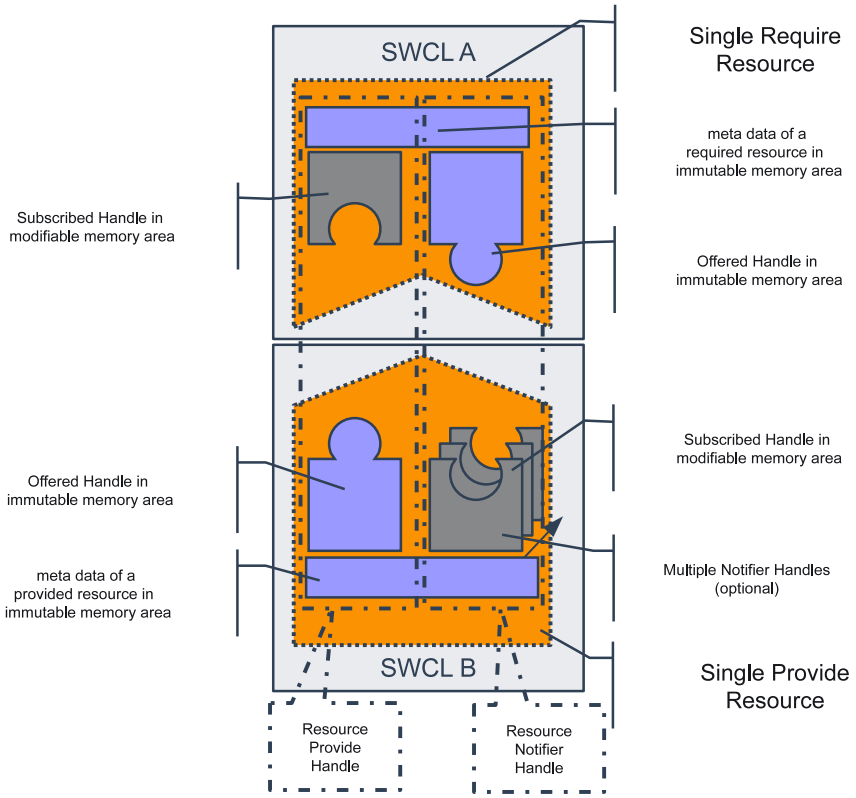

Figure 11.9: Logical structure of the `Software Cluster Binary Manifest`

[TPS_SYST_02328] Semantics of meta-class `CpSoftwareClusterBinaryManifestDescriptor` ⌈The existence of the `CpSoftwareClusterBinaryManifestDescriptor` represents the definition of the model of a `Software Cluster Binary Manifest` for a given `Software Cluster`.

Because of this relation, it is necessary that `CpSoftwareClusterBinaryManifestDescriptor` references the corresponding `CpSoftwareCluster` instead of being aggregated by it.⌋

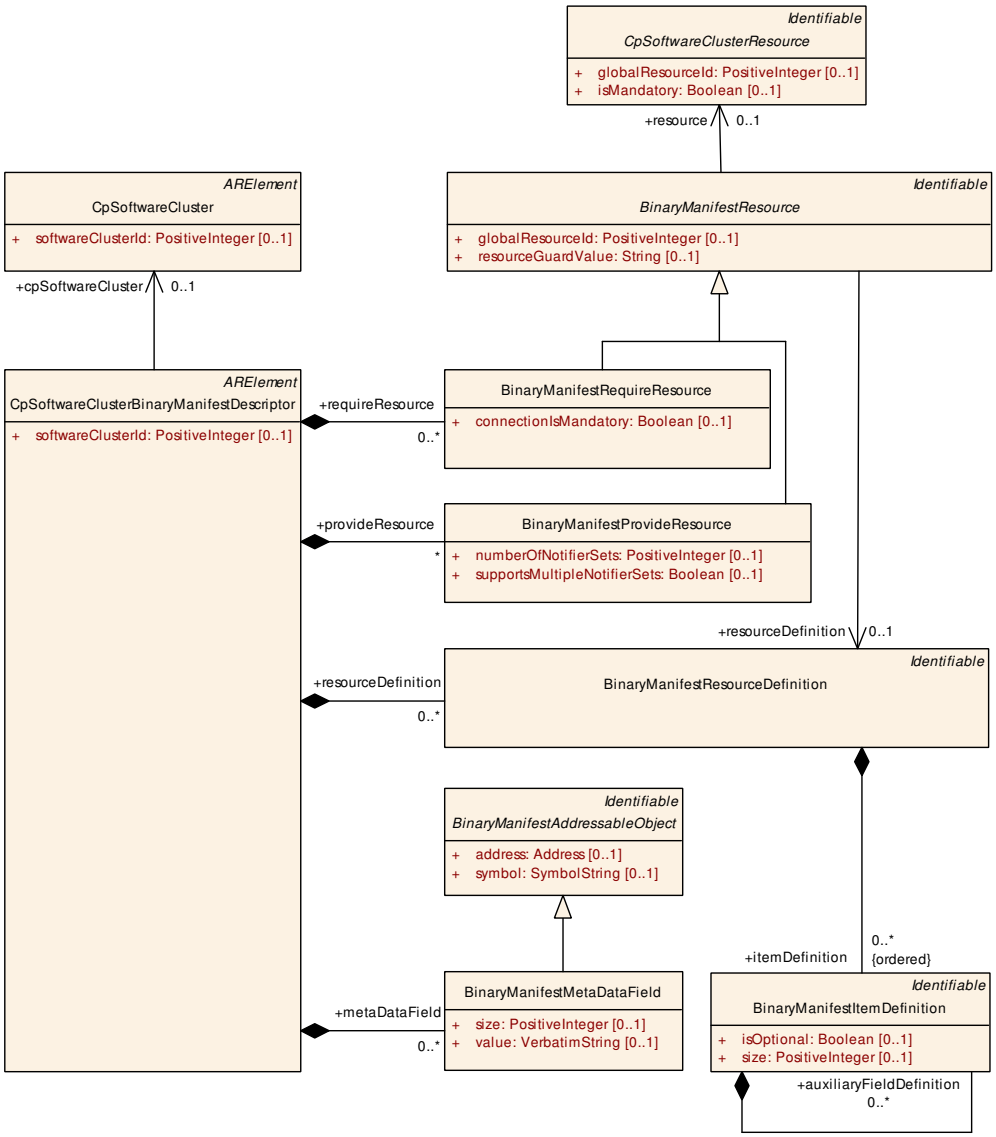

Figure 11.10: Modeling of the `Software Cluster Binary Manifest`

[TPS_SYST_02329] Provison of a `Software Cluster`'s ID ⌈The ID of a given `Software Cluster` formalized as `CpSoftwareCluster` cannot be defined as an attribute of `CpSoftwareCluster` because the ID represents an integration-level information. Therefore, the ID is provided by means of attribute `CpSoftwareCluster-BinaryManifestDescriptor.softwareClusterId`.⌋

[TPS_SYST_02330] Possible values of attribute `CpSoftwareClusterBinary-ManifestDescriptor.category` ⌈The following values for attribute `CpSoftware-Cluster.category` are standardized by AUTOSAR:

- **`HOST_SOFTWARE_CLUSTER`**: the `CpSoftwareClusterBinaryManifest-Descriptor` that contains the major part of the basic-software stack, especially micro-controller-dependent modules including the operating system.

- **APPLICATION_SOFTWARE_CLUSTER**: the CpSoftwareClusterBinaryManifestDescriptor represents application-level functionality conceptually located on top of the CpSoftwareCluster of category HOST_SOFTWARE_CLUSTER.
- **SUBSTITUTION_SOFTWARE_CLUSTER**: in this case the CpSoftwareClusterBinaryManifestDescriptor is used for debugging or prototyping purposes.

⌋

Please note that the standardized values of attribute category for CpSoftwareCluster and CpSoftwareClusterBinaryManifestDescriptor differ by the definition of value SUBSTITTION_SOFTWARE_CLUSTER.

This value is not applicable on the “System”-level, but may make sense in the local scope of the deployment configuration of a given CpSoftwareCluster by means of CpSoftwareClusterBinaryManifestDescriptor.

Class	**CpSoftwareClusterBinaryManifestDescriptor**			
Note	This meta-class has the ability to act as a hub for all information related to the binary manifest of a given CP software cluster. The manifest is subject to integrator work and therefore not a part of the definition of the CP software cluster itself. **Tags:** atp.recommendedPackage=CpSoftwareClusterBinaryManifestDescriptors This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
cpSoftware Cluster	CpSoftwareCluster	0..1	ref	This reference identifies the CpSoftwareCluster to which the enclosing CpSoftwareClusterBinaryManifest Descriptor belongs, The CpSoftwareClusterBinaryManifestDescriptor is defined in an integration phase while the referenced Cp SoftwareCluster represents a design element. Therefore, it makes sense to use a reference rather than an aggregation in the relation of the two meta-classes. **Tags:** atp.Splitkey=cpSoftwareCluster
metaDataField	BinaryManifestMeta DataField	*	aggr	This aggregation identifies the collection of meta-data contained in the enclosing binary manifest. **Tags:** atp.Splitkey=metaDataField, metaDataField.short Name
provide Resource	BinaryManifestProvide Resource	*	aggr	This aggregation represents the collection of provided resources in the enclosing binary manifest. **Tags:** atp.Splitkey=provideResource, provide Resource.shortName
require Resource	BinaryManifestRequire Resource	*	aggr	This aggregation represents the collection of required resources in the enclosing binary manifest. **Tags:** atp.Splitkey=requireResource, require Resource.shortName

▽

△

Class	**CpSoftwareClusterBinaryManifestDescriptor**			
resource Definition	BinaryManifest ResourceDefinition	*	aggr	This aggregation represents the collection of binary manifest resource definitions that belong to the enclosing CpSoftwareClusterBinaryManifestDescriptor. **Tags:** atp.Splitkey=resourceDefinition, resource Definition.shortName
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster. This id is assigned by an integrator, but may also be copied from CpSoftware Cluster.softwareClusterId if available.

Table 11.17: CpSoftwareClusterBinaryManifestDescriptor

[constr_5359] `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId`

Imposition time: CP: `IT_BinObjMetaData`

⌈The `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId` in case that the `softwareClusterId` is set in the `CpSoftwareCluster` that is referenced via `CpSoftwareClusterBinaryManifestDescriptor.cpSoftwareCluster`.⌋

[TPS_SYST_02331] Definition of provided resource in the context of the `Software Cluster Binary Manifest` ⌈A provided resource of a `Software Cluster Binary Manifest` represents an "offer" of the `Software Cluster` to other `Software Cluster`s.

For the formal point of view, this "offer" is defined by means of a `BinaryManifestProvideResource`, aggregated at `CpSoftwareClusterBinaryManifestDescriptor` in the role `provideResource`.

Attribute `supportsMultipleNotifierSets` indicates whether the resource support the call of multiple notifier call-back functions.⌋

Class	**BinaryManifestProvideResource**			
Note	This meta-class represents a provided resource in the binary manifest. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.provideResource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
numberOf NotifierSets	PositiveInteger	0..1	attr	This attribute provides an upper limit for the number of notifiers for this resource.
supports MultipleNotifier Sets	Boolean	0..1	attr	This attribute indicates whether the enclosing Binary ManifestResource supports multiple notifiers sets.

Table 11.18: BinaryManifestProvideResource

[constr_5185] Existence of attribute BinaryManifestProvideResource.globalResourceId

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute globalResourceId shall exist.⌋

[constr_5186] Existence of attribute BinaryManifestProvideResource.resourceGuardValue

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute resourceGuardValue shall exist.⌋

[constr_5187] Existence of attribute BinaryManifestProvideResource.supportsMultipleNotifierSets

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute supportsMultipleNotifierSets shall exist.⌋

[constr_5188] Existence of attribute BinaryManifestProvideResource.numberOfNotifierSets

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, attribute numberOfNotifierSets shall exist.⌋

[constr_5189] Existence of reference BinaryManifestProvideResource.resourceDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, the reference in the role resourceDefinition shall exist.⌋

[constr_5190] Existence of aggregation BinaryManifestProvideResource.item

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestProvideResource, the aggregation in the role item shall exist at least once.⌋

[constr_5191] Consequence of attribute BinaryManifestProvideResource.item.category

Imposition time: CP: IT_BinObjMetaData

⌈The following values of attribute BinaryManifestProvideResource.item.category shall require the existence of aggregations:

- If category is set to PROVIDER_HANDLE and the attribute isUnused is not set to true then the aggregation BinaryManifestProvideResource.item.value shall exist
- If category is set to NOTIFIER_HANDLE and the attribute isUnused is not set to true then the aggregation BinaryManifestProvideResource.item.defaultValue shall exist
- If category is set to AUXILARY_ACTUAL_NUMBER_NOTIFIER_SETS then the aggregation BinaryManifestProvideResource.item.defaultValue shall exist

⌋

Class	***BinaryManifestResource*** (abstract)			
Note	This meta-class acts as an abstract base class for specializations. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BinaryManifestProvideResource, BinaryManifestRequireResource			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
globalResourceId	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines.
item (ordered)	BinaryManifestItem	*	aggr	This aggregation represents the collection of binary manifest handles owned by the enclosing binary manifest resource.
resource	CpSoftwareClusterResource	0..1	ref	This reference identifies the CpSoftwareClusterResource (on design level) that corresponds to the BinaryManifestResource (on integration level). **Tags:** atp.Splitkey=resource
resourceDefinition	BinaryManifestResourceDefinition	0..1	ref	this reference identifies the definition of the BinaryManifestResource. The definition provides configuration information that is shared among all BinaryManifestResources that refer to the BinaryManifestResourceDefinition. **Tags:** atp.Splitkey=resourceDefinition
resourceGuardValue	String	0..1	attr	This attribute specifies the guard value of the enclosing binary manifest resource.

Table 11.19: BinaryManifestResource

[TPS_SYST_02332] Definition of required resource in the context of the Software Cluster Binary Manifest ⌈A required resource of a Software Cluster Binary Manifest represents an “request” of the Software Cluster towards other Software Clusters.

For the formal point of view, this “request” is defined by means of a BinaryManifestRequireResource, aggregated at CpSoftwareClusterBinaryManifestDescriptor in the role requireResource.

Required resources could be left unconnected unless attribute connectionIs-Mandatory is set to True.⌋

Class	BinaryManifestRequireResource			
Note	This meta-class represents a required resource in the binary manifest. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.requireResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
connectionIs Mandatory	Boolean	0..1	attr	This attribute indicates whether the connection of the enclosing BinaryManifestResource is mandatory.

Table 11.20: BinaryManifestRequireResource

[constr_5192] Existence of attribute BinaryManifestRequire-Resource.globalResourceId

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, attribute globalResourceId shall exist.⌋

[constr_5193] Existence of attribute BinaryManifestRequire-Resource.resourceGuardValue

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, attribute resourceGuardValue shall exist.⌋

[constr_5194] Existence of reference BinaryManifestRequire-Resource.resourceDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, the reference in the role resourceDefinition shall exist.⌋

[constr_5195] Existence of aggregation BinaryManifestRequire-Resource.item

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestRequireResource, the aggregation in the role item shall exist at least once.⌋

[constr_5196] Consequence of attribute BinaryManifestRequire-Resource.item.category

Imposition time: CP: IT_BinObjMetaData

⌈The following values of attribute BinaryManifestRequireResource.item.cat-egory shall require the existence of aggregations:

- If category is set to PROVIDER_HANDLE then the aggregation Binary-ManifestRequireResource.item.defaultValue shall exist

- If category is set to NOTIFIER_HANDLE then the aggregation BinaryManifestRequireResource.item.value shall exist

⌋

Class	BinaryManifestResourceDefinition			
Note	This meta-class represents the ability to specify a resource definition that provides information that can be shared by all resources that refer to the respective resource definition. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.resourceDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
itemDefinition (ordered)	BinaryManifestItemDefinition	*	aggr	This aggregation specifies the collection of handle definitions in the context of the enclosing binary manifest resource definitions.

Table 11.21: BinaryManifestResourceDefinition

[constr_5197] Existence of aggregation BinaryManifestResourceDefinition.itemDefinition

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestResourceDefinition, the aggregation in the role itemDefinition shall exist at least once.⌋

[TPS_SYST_02333] Purpose of meta-class BinaryManifestResourceDefinition ⌈The purpose of meta-class BinaryManifestResourceDefinition is to provide attributes that apply (by definition) to all BinaryManifestResources that refer to a specific BinaryManifestResourceDefinition.

In other words, the configuration modeled in a BinaryManifestResourceDefinition is shared among all BinaryManifestResources that reference the BinaryManifestResourceDefinition.⌋

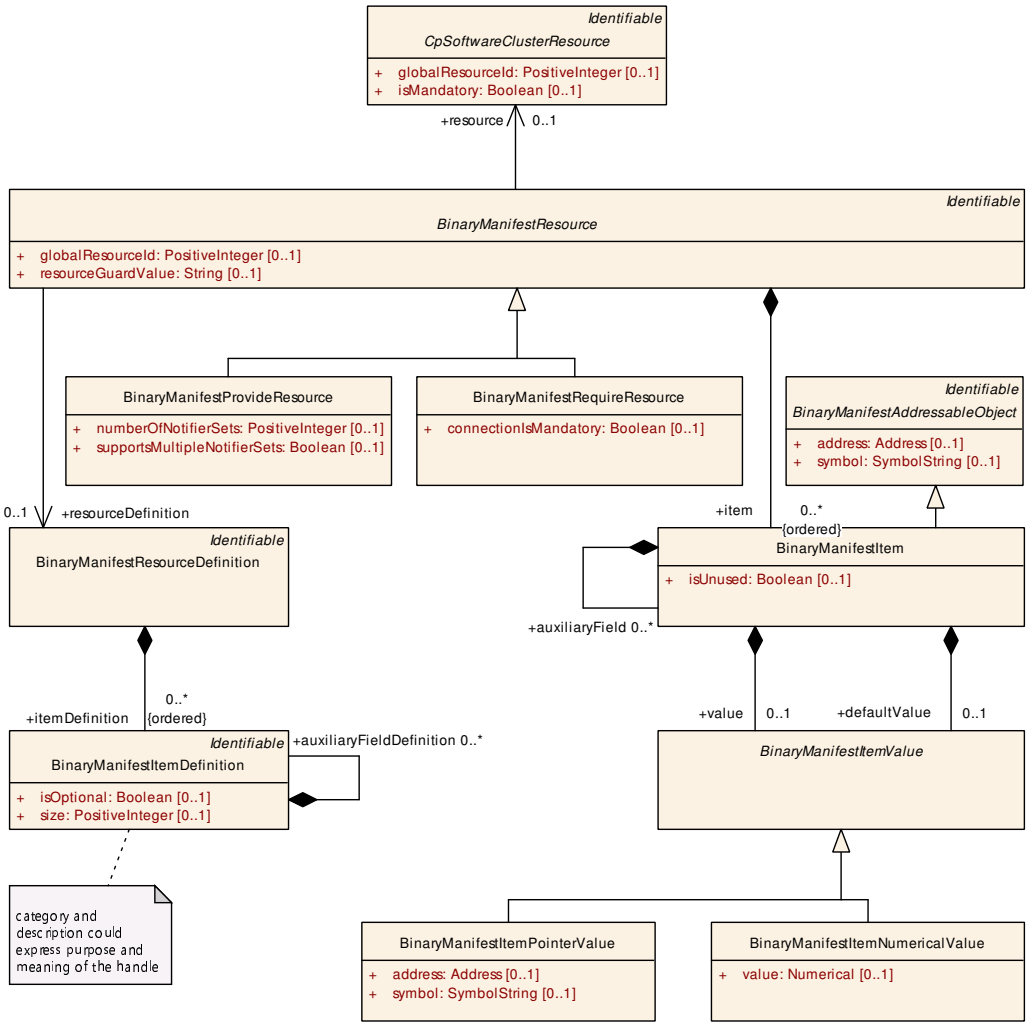

Figure 11.11: Modeling of the `BinaryManifestResource`

Please note that by referencing `BinaryManifestResourceDefinition` from `BinaryManifestResource` repetition can be avoided. It is expected that in the context of a given `CpSoftwareClusterBinaryManifestDescriptor` several `BinaryManifestResource`s exist that share the same `BinaryManifestResourceDefinition`.

[constr_5198] Allowed `BinaryManifestResource.resourceDefinition`

Imposition time: CP: `IT_BinObjMetaData`

⌈An `BinaryManifestResourceDefinition` shall only be referenced from a `BinaryManifestResource` that is aggregated in the same `CpSoftwareClusterBinaryManifestDescriptor` as the referenced `BinaryManifestResourceDefinition`.⌋

[TPS_SYST_02334] Semantics of meta-class `BinaryManifestItem` ⌈The purpose of meta-class `BinaryManifestItem` is to define elements that specify the detailed interface (on the level of symbols) of a `BinaryManifestResource`.

At run-time, the BinaryManifestItem is represented by its symbol located at the corresponding address.⌋

[constr_5271] Existence of attribute BinaryManifestItem.isUnused

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItem, the attribute isUnused shall exist.⌋

[constr_5272] Value of attribute BinaryManifestItem.isUnused

Imposition time: CP: IT_BinObjMetaData

⌈The attribute BinaryManifestItem.isUnused shall only permitted to be set to true if the related BinaryManifestItemDefinition has its attribute isOptional set to true.⌋

[TPS_SYST_02335] Semantics of aggregation BinaryManifestItem.auxiliaryField ⌈The aggregation BinaryManifestItem.auxiliaryField can be used to define structured BinaryManifestItem.⌋

Class	**BinaryManifestItem**			
Note	This meta-class represents the ability to describe a specific handle or auxiliary field in the context of binary manifest resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItem.auxiliaryField, *BinaryManifestResource*.item			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
auxiliaryField	BinaryManifestItem	*	aggr	This aggregation is used to define structured Binary ManifestItems. **Tags:** atp.Splitkey=auxiliaryField, auxiliaryField.shortName xml.sequenceOffset=20
defaultValue	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a default value for a binary manifest handle or an auxiliaryField. This value shall be taken if no connection for this resource is possible. **Tags:** atp.Splitkey=defaultValue xml.sequenceOffset=10
isUnused	Boolean	0..1	attr	If true, the handle or auxiliary field in the context of binary manifest resource relates to an optional BinaryManifest ItemDefinition and is not used.
value	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a value for a binary manifest handle or an auxiliaryField. This value shall be taken to establish a connection. **Tags:** atp.Splitkey=value

Table 11.22: BinaryManifestItem

[constr_5199] Consequence of attribute BinaryManifestItem.auxiliaryField.category

Imposition time: CP: IT_BinObjMetaData

⌈If attribute BinaryManifestItem.auxiliaryField.category is set to value AUXILARY_CONNECTED_SW_CLUSTER_ID then attribute BinaryManifestItem.auxiliaryField.defaultValue shall exist.⌋

[TPS_SYST_02336] Semantics of meta-class BinaryManifestItemDefinition ⌈The purpose of meta-class BinaryManifestItemDefinition is to provide attributes that are shared among all corresponding BinaryManifestItems.⌋

[TPS_SYST_02337] Semantics of aggregation BinaryManifestItemDefinition.auxiliaryFieldDefinition ⌈The aggregation BinaryManifestItemDefinition.auxiliaryFieldDefinition can be used to define structured BinaryManifestItemDefinition.⌋

[TPS_SYST_02338] Relation between BinaryManifestItemDefinition and BinaryManifestItem ⌈The relation between a particular BinaryManifestItemDefinition and a particular BinaryManifestItem is created by

- the **ordered** aggregation of meta-class BinaryManifestItemDefinition at BinaryManifestResource in the role itemDefinition and
- the **ordered** aggregation of meta-class BinaryManifestItem at BinaryManifestResource in the role item.

In other words, the mentioned correspondence is created such that the nth BinaryManifestResourceDefinition.itemDefinition applies to the nth BinaryManifestResource.item.⌋

Class	**BinaryManifestItemDefinition**			
Note	This meta-class provides the ability to define the handle definition or an auxiliary field of a binary manifest resource. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItemDefinition.auxiliaryFieldDefinition, BinaryManifestResourceDefinition.itemDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
auxiliaryField Definition	BinaryManifestItem Definition	*	aggr	This aggregation is used to define structured Binary ManifestItemDefinitions. **Tags:** atp.Splitkey=auxiliaryFieldDefinition, auxiliaryField Definition.shortName
isOptional	Boolean	0..1	attr	If true, the handle definition or auxiliary field of a binary manifest resource is optional and may not be used in all BinaryManifestResources referring to this BinaryManifest ResourceDefinition.
size	PositiveInteger	0..1	attr	This attribute provides the ability to specify the size of the enclosing BinaryManifestResourceDefinition.

Table 11.23: BinaryManifestItemDefinition

[constr_5200] Existence of attribute BinaryManifestItemDefinition.category

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemDefinition, attribute category shall exist.⌋

[constr_5201] Existence of attribute BinaryManifestItemDefinition.size

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemDefinition, attribute size shall exist.⌋

Class	***BinaryManifestAddressableObject*** (abstract)			
Note	This meta-class acts as an abstract base class for addressable objects in the context of the binary manifest of a CP software cluster. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BinaryManifestItem, BinaryManifestMetaDataField			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
address	Address	0..1	attr	This attribute specifies the address of the enclosing addressable object.
symbol	SymbolString	0..1	attr	This attribute specifies the symbol of the addressable object.

Table 11.24: BinaryManifestAddressableObject

[TPS_SYST_02339] Standardized values of attribute BinaryManifestAddressableObject.category ⌈The following list of values of attribute BinaryManifestAddressableObject.category is standardized by AUTOSAR:

- PROVIDER_HANDLE: the BinaryManifestAddressableObject is used to store a provider handle.
- NOTIFIER_HANDLE: the BinaryManifestAddressableObject is used to store a notifier handle.
- IMMUTABLE_TABLES_CHECKSUM: the Immutable Tables Checksum is built over all constants of the Software Cluster Binary Manifest which are not changed by the Software Cluster connection step.
- SUBSCRIBED_INTERFACE_VALIDITY_MARKER: the Subscribed Interface Validity Marker indicate that all tables storing subscribed handles are written after the Software Cluster connection step.
- AUXILARY_ACTUAL_NUMBER_NOTIFIER_SETS: the auxiliary field actual number of used notifier sets describes how many of the notifier sets are occupied by connected resources.
- AUXILARY_CONNECTED_SW_CLUSTER_ID: the auxiliary field connected Software Cluster Id holds the Software Cluster Id from which the handle values are taken in case an connection is established.

⌋

Please note that custom values of BinaryManifestAddressableObject.category are supported as long as the custom value is created in a way that it will not clash with standardized values that may be added in the future. Such a name clash can be prevented by using company-specific or project-specific prefixes, suffixes, or infixes in the value.

[TPS_SYST_02340] Semantics of abstract meta-class BinaryManifestItemValue ⌈Sub-classes of abstract meta-class BinaryManifestItemValue can be used to specify **default values** for a BinaryManifestItem.⌋

Class	***BinaryManifestItemValue*** (abstract)			
Note	This meta-class has the ability to act as an abstract base class for values of binary manifest item. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Subclasses	BinaryManifestItemNumericalValue, BinaryManifestItemPointerValue			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 11.25: BinaryManifestItemValue

Class	**BinaryManifestItemNumericalValue**			
Note	This meta-class has the ability to provide a numerical value for a binary manifest item. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	Numerical	0..1	attr	This attribute specifies the actual numerical value to be used in the binary manifest handle.

Table 11.26: BinaryManifestItemNumericalValue

[constr_5202] Existence of attribute BinaryManifestItemNumericalValue.value

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestItemNumericalValue, attribute value shall exist.⌋

[TPS_SYST_02341] Semantics of the aggregation of meta-class BinaryManifestItemPointerValue in the role defaultValue ⌈Meta-class BinaryManifestItemPointerValue is used to provide a default value for a pointer.

This means that the default value consists of the address and the symbol of the target object of the pointer.

In other words, the memory at the BinaryManifestItem.address is filled with the default target address taken from BinaryManifestItemPointerValue.address.⌋

Class	**BinaryManifestItemPointerValue**			
Note	This meta-class has the ability to provide a value for a pointer in the context of a binary manifest item. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
address	Address	0..1	attr	This attribute represents the address value of the enclosing pointer value.
symbol	SymbolString	0..1	attr	This attribute represents the symbol associated with the binary manifest handle.

Table 11.27: BinaryManifestItemPointerValue

[constr_5218] Existence of attribute `BinaryManifestItemPointerValue.address`

Imposition time: CP: `IT_BinObjMetaData`

⌈For each `BinaryManifestItemPointerValue`, attribute `address` shall exist.⌋

[constr_5203] Existence of attribute `BinaryManifestItemPointerValue.symbol`

Imposition time: CP: `IT_BinObjMetaData`

⌈For each `BinaryManifestItemPointerValue`, attribute `symbol` shall exist.⌋

[TPS_SYST_02342] Semantics of meta-class `BinaryManifestMetaDataField`

⌈Meta-class `BinaryManifestMetaDataField` is used to describe meta-data of a `Software Cluster Binary Manifest`. This can be achieved by means of the aggregation of `BinaryManifestMetaDataField` at `CpSoftwareClusterBinaryManifestDescriptor` in the role `metaDataField`.

As a part of the `Software Cluster Binary Manifest`, a `BinaryManifestMetaDataField` is represented at run-time by a `symbol` that is located at an `address`.

On the model level, attribute `category` can be used to further categorize a specific `BinaryManifestMetaDataField`.⌋

Class	**BinaryManifestMetaDataField**			
Note	This meta-class provides the ability to define a meta-data field for the binary manifest descriptor. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.metaDataField			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
size	PositiveInteger	0..1	attr	The value of this attribute represents the size of the meta-data field in bytes.
value	VerbatimString	0..1	attr	This attribute specifies the value of the meta-data field.

Table 11.28: BinaryManifestMetaDataField

[constr_5204] Existence of attribute BinaryManifestMetaDataField.category

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute category shall exist.⌋

[constr_5205] Existence of attribute BinaryManifestMetaDataField.size

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute size shall exist.⌋

[constr_5217] Existence of attribute BinaryManifestMetaDataField.value

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField of category IMMUTABLE_TABLES_CHECKSUM, attribute value shall exist.⌋

[constr_5206] Existence of attribute BinaryManifestMetaDataField.symbol

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute symbol shall exist .⌋

[constr_5207] Existence of attribute BinaryManifestMetaDataField.address

Imposition time: CP: IT_BinObjMetaData

⌈For each BinaryManifestMetaDataField, attribute address shall exist.⌋

Please note that the meta-data of a Software Cluster Binary Manifest represent information that is immutable during the phase of connecting Software Clusters with each other.

11.4 Software Cluster Extraction

A System with category SYSTEM_DESCRIPTION is typically used to describe the complete vehicle with all included ECUs. Such a SYSTEM_DESCRIPTION may also contain several CpSoftwareClusters that are assigned to different EcuInstances by CpSoftwareClusterToEcuInstanceMappings.

To describe the content of a single CpSoftwareCluster a new System category is introduced:

[TPS_SYST_02343] System with category SW_CLUSTER_SYSTEM_DESCRIPTION
⌈A System with category SW_CLUSTER_SYSTEM_DESCRIPTION describes the content of a single CpSoftwareCluster.⌋

[TPS_SYST_02344] SW_CLUSTER_SYSTEM_DESCRIPTION content ⌈A System with category SW_CLUSTER_SYSTEM_DESCRIPTION shall only contain the CpSoftwareCluster relevant content, e.g.

- the EcuInstance to which the CpSoftwareCluster is mapped by CpSoftwareClusterToEcuInstanceMapping,
- the ApplicationPartitions to which the CpSoftwareCluster is mapped by CpSoftwareClusterResourceToApplicationPartitionMapping,
- the CompositionSwComponentTypes referenced by the CpSoftwareCluster in the role swComposition with all included SwComponentPrototypes and all elements used by these SwComponentPrototypes directly or indirectly,
- the SwComponentPrototypes referenced by the CpSoftwareCluster via SwComponentPrototypeAssignment and all elements used by these SwComponentPrototypes directly or indirectly,
- the CpSoftwareClusterToResourceMappings that references the CpSoftwareCluster in the role provider or requester with all CpSoftwareClusterServiceResources that in turn are referenced by the CpSoftwareClusterToResourceMappings,
- the SwConnectors that describe the communication relation between SwComponentPrototypes of the CpSoftwareCluster in focus,
- the PortInterfaceMappings that are referenced by the included SwConnectors and do not lead to a conversion of the data/operation representation since the data scaling between CpSoftwareClusters is excluded,
- the DataMappings that describe the communication relation to different CpSoftwareClusters and to SwComponentPrototypes outside of the CpSoftwareCluster in focus,
- the communication matrix description related content that is related to the included SystemSignals, e.g. ISignals, ISignalTriggerings, ISignalIPdus and all the rest of them.

⌋

Please note that in some cases the relevant DataMappings for the CpSoftwareCluster may be described in the SYSTEM_DESCRIPTION on the opposite side, i.e. on different CpSoftwareClusters or on SwComponentPrototypes outside of the CpSoftwareCluster. The reason is that in the SYSTEM_DESCRIPTION the definition of DataMappings is sufficient on one side. But since these side will be removed during the creation of the SW_CLUSTER_SYSTEM_DESCRIPTION the relevant DataMappings shall be shifted to the CpSoftwareCluster in focus.

For the Ecu and RTE Configuration a System with category ECU_EXTRACT is derived from the SYSTEM_DESCRIPTION, ECU_SYSTEM_DESCRIPTION or SW_CLUSTER_SYSTEM_DESCRIPTION as described by [TPS_SYST_01139].

The derivation of an ECU_EXTRACT is a model transformation since a “flat view” of SwComponentPrototypes running on the EcuInstance is expected by the RTE.

The SwComponentPrototypes that are included in the CompositionSwComponentType that is referenced by a CpSoftwareCluster in the role swComposition need to be flattened according to the rules described in chapter 14 in the ECU_EXTRACT. So in case of a SW_CLUSTER_SYSTEM_DESCRIPTION the ECU_EXTRACT represents the "flat view" of a single CpSoftwareCluster.

Please note that during the creation of the ECU_EXTRACT the «instanceRef»s in PortElementToCommunicationResourceMapping shall be adapted because of the process of flattening the hierarchical Software Composition into the "flat view" representation.

[constr_5208] Existence of System.swCluster

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.swCluster shall exist at least once.⌋

[constr_5209] Existence of reference CpSoftwareCluster.swComponentAssignment.swComponent

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.swCluster.swComponentAssignment.swComponent shall exist.⌋

[constr_5210] Existence of reference SystemMapping.portElementToComResourceMapping

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.portElementToComResourceMapping shall exist at least once.⌋

[constr_5211] Existence of reference PortElementToCommunicationResourceMapping.communicationResource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.portElementToComResourceMapping.communicationResource shall exist at least once.⌋

[constr_5212] Existence of reference SystemMapping.resourceToApplicationPartitionMapping

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping shall exist.⌋

[constr_5213] Existence of reference CpSoftwareClusterResourceToApplicationPartitionMapping.applicationPartition

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping.applicationPartition shall exist.⌋

[constr_5214] Existence of reference CpSoftwareClusterResourceToApplicationPartitionMapping.resource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.resourceToApplicationPartitionMapping.resource shall exist.⌋

[constr_5215] Existence of reference CpSoftwareClusterToResourceMapping.serviceResource

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, the reference System.mapping.softwareClusterToResourceMapping.serviceResource shall exist.⌋

[constr_5216] Existence of reference CpSoftwareClusterToResourceMapping.requester and/or provider

Imposition time: CP: IT_SwCluSysDesc

⌈In a System with category SW_CLUSTER_SYSTEM_DESCRIPTION, at least one of the references System.mapping.softwareClusterToResourceMapping.requester or System.mapping.softwareClusterToResourceMapping.provider shall exist.⌋

11.4.1 Software Cluster Extraction and DataMappings

The creation of the SW_CLUSTER_SYSTEM_DESCRIPTION for the HOST_SOFTWARE_CLUSTER takes a special role. The DataMappings and CpSoftwareClusterToResourceMappings of all CpSoftwareClusters that are mapped to the same EcuInstance as the HOST_SOFTWARE_CLUSTER shall be made available in the SW_CLUSTER_SYSTEM_DESCRIPTION of the HOST_SOFTWARE_CLUSTER.

This is shown in the example in Figure 11.12, where the Mappings of the outerPort that is connected to “Composition1” are available in the SW_CLUSTER_SYSTEM_DESCRIPTION for “SWCL11” where the “Composition1” is located and the “HOST” CpSoftwareCluster.

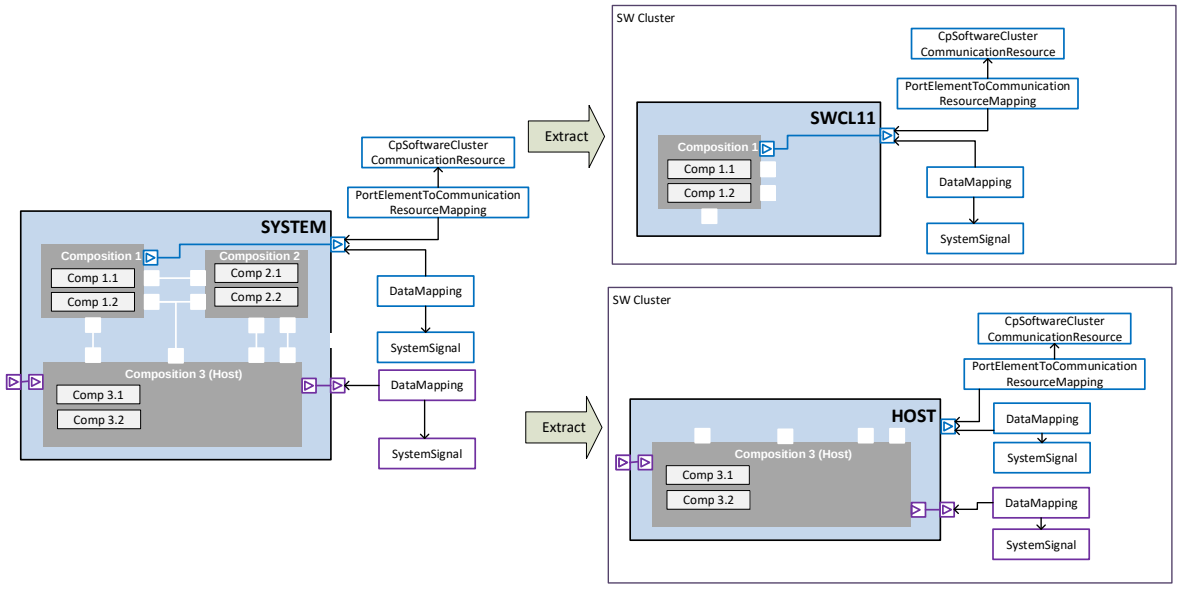

Figure 11.12: Handling of DataMapping during creation of SW_CLUSTER_SYSTEM_DESCRIPTION

The DataMappings on the HOST_SOFTWARE_CLUSTER are necessary for the generation of the AUTOSAR COM Stack. ISignals that are referencing the SystemSignals are mapped into ISignalIPdus and these ISignalIPdus are instantiated on the PhysicalChannels via PduTriggerings. This information is necessary to derive the COM Pdus. The ISignalIPdus are then mapped into Frames for communication over CAN or FlexRay or are assigned to ProvidedServiceInstances or ConsumedServiceInstances for communication over SOME/IP. All information that is necessary to configure the COM Stack shall be available in the HOST_SOFTWARE_CLUSTER.

The outerPorts of the APPLICATION_SOFTWARE_CLUSTERs and the available DataMappings and CpSoftwareClusterToResourceMappings in these CpSoftwareClusters are used to configure the RTE and the Transformers.

12 Usage of the System Template

As introduced in [TPS_SYST_01003] the System Template is used to describe a `System` with `category` `SYSTEM_CONSTRAINT_DESCRIPTION`, a `System` with `category` `ABSTRACT_SYSTEM_DESCRIPTION` and a `System` with `category` `SYSTEM_DESCRIPTION`. `System` with `category` `SYSTEM_EXTRACT` is described in more detail in chapter 13. `System` with `category` `ECU_EXTRACT` is described in more detail in chapter 14.

Certain elements of the System Template may have a different meaning at the different stages of the AUTOSAR Methodology. The following sections describe the differences.

12.1 System Constraint Description

Meta-classes, Chapters	Usage to describe the System Constraints	Usage to describe the System Configuration
CommunicationCluster, EcuInstance (Chapter 3)	The Topology is completely described in the System Constraint Description.	The Topology description will be unchanged copied to the System Configuration description. The Topology may only be changed during another iteration development step of the whole system.
FrameTriggering, PduTriggering, ISignalTriggering (Chapter 6)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION describes all FrameTriggerings that are predefined on all CommunicationClusters of a vehicle. The predefinition of the communication matrix forces the system generator to use the given FrameTriggerings. Constraints for the system generator arise here e.g. from the used bus bandwidth, used identifiers as well as from the timing and at which position in a Frame a Pdu is transmitted on a PhysicalChannel on a CommunicationCluster. Such a manual definition of the communication can be made for any reason where it is necessary to restrict the system generator. One example is the usage of legacy EcuInstances in an AUTOSAR System. The FrameTriggerings that are transmitted or received by these legacy EcuInstances are constraints for the system generator because they cannot be changed, if the compatibility is supposed to be achieved without any changes at the legacy EcuInstances.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION contains all FrameTriggerings, PduTriggerings, ISignalTriggerings that will be sent by any EcuInstance in the car. No matter if they were predefined (system constraint) or if they were generated by the system generator. The available information, in addition to the information, which is inserted by the AUTOSAR Ecu configuration generator step, will be used as input to configure the Basic SW for the communication.
Gateway (Chapter 8)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION describes all Gateways in the system including their IPduMappings and ISignalMappings that are predefined. The reasons for such predefinitions are quite the same as for the predefinitions of the FrameTriggerings.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION describes all Gateways with all their IPduMappings and ISignalMappings. No matter if they were predefined (System Constraint) or if they were generated by the System Generator.
SwcToEcuMapping (Chapter 5.1.1)	The mapping of Software Components to EcuInstances may be predefined. The predefinition will force the system generator to use the specified mapping. Thus, with the SwcToEcuMapping element it is possible to describe that one or more Software Components shall be mapped to a specific EcuInstance.	In a complete System with category SYSTEM_DESCRIPTION, all Software Components are mapped to EcuInstances.
MappingConstraint (Chapter 5.1.4) ComponentClustering (Chapter 5.1.4.1) ComponentSeparation (Chapter 5.1.4.2)	There may be system constraints that limit the system generators freedom to map Software Components to arbitrary EcuInstances. These system constraints can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator.	After the mapping has been completed, the System with category SYSTEM_DESCRIPTION will contain mapping descriptions for all elements, and the mapping constraints are obsolete. But that does not mean that mapping constraints have to be deleted after the system generation step. By deleting the mapping constraints you would lose the information why a mapping of a Software Component to an EcuInstance is chosen.

▽

△

Meta-classes, Chapters	Usage to describe the System Constraints	Usage to describe the System Configuration
DataMapping (Chapter 5.2) SenderReceiverToSignalMapping (Chapter 5.2.1.1) SenderReceiverToSignalGroupMapping (Chapter 5.2.1.2) ClientServerToSignalMapping (Chapter 5.2.1.3)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION may describe the predefined mapping of Software Components to certain EcuInstances (see chapter 5.1.1). Only if such a mapping exists, it is reasonable to define the DataMapping of the data exchanged between the Software Components.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION shall contain all DataMapping definitions. No matter if they were predefined (system constraint) or if they were generated by the System-Generator.
SignalPathConstraint (Chapter 5.2.2) CommonSignalPath (Chapter 5.2.2.1) ForbiddenSignalPath (Chapter 5.2.2.2) PermissibleSignalPath (Chapter 5.2.2.3) SeparateSignalPath (Chapter 5.2.2.4)	It can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator, which specific way a VariableDataPrototype or ClientServerOperation should take in the network without defining in which Pdu and Frame it is transmitted.	SignalPathConstraints are not an obligatory part of the System with category SYSTEM_DESCRIPTION. In the final System with category SYSTEM_DESCRIPTION every ISignal is assigned to a Pdu and every Pdu is assigned to a Frame. Thereby the paths of VariableDataPrototypes or ClientServerOperations on the network are implicitly described. But that does not mean that the SignalPathConstraints have to be deleted after the system generation step. By deleting the SignalPathConstraints you would lose the information why you have chosen e.g. a specific mapping of an ISignal into a Pdu. If you extend or change the system at a later stage the missing SignalPathConstraints could lead to not wanted signal mappings by the System Generator.

Table 12.1: Usage of the System Template

12.2 Abstract System Description

[TPS_SYST_01134] Abstract System Description ⌈Due to the fact that the functional view on vehicle system can differ from the actual technical definition of the software-architectures of individual EcuInstances the System Template optionally allows to define a System with category ABSTRACT_SYSTEM_DESCRIPTION.⌋

[TPS_SYST_01135] Refactoring of an Abstract System Description into a project specific technical view of the software architecture ⌈The System with category ABSTRACT_SYSTEM_DESCRIPTION concentrates on the functional aspects of the system design and provides an own abstract VFB. During the further activities this abstract view shall be refactored into a more project specific technical view of the software architecture.

It is important to note that during the refactoring of the System with category ABSTRACT_SYSTEM_DESCRIPTION into the System with category SYSTEM_DESCRIPTION no restrictions to the allowed actions apply (This is in contrast to the activity of deriving the System with category SYSTEM_EXTRACT from the System with category SYSTEM_DESCRIPTION, see [TPS_SYST_01123].⌋

[TPS_SYST_01136] `ViewMapSet` and `ViewMap` are used to trace the transformations between different models ⌈The `ViewMapSet` and `ViewMap` elements are used to trace the transformations between different models within the AUTOSAR environment.⌋

These classes are described in more detail in the Generic Structure Template [1].

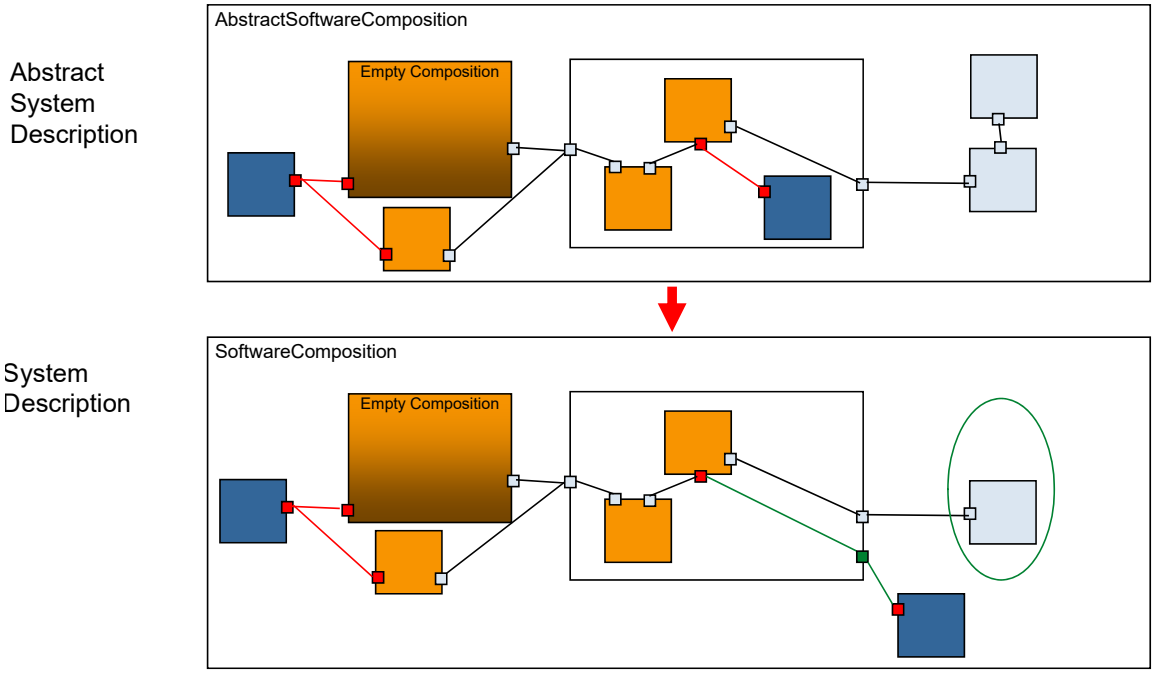

Figure 12.1: Abstract System Description refactoring to a System Description

13 System Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR work product System with category SYSTEM_EXTRACT, based on Meta Model elements contained in the System Template and Software Component Template.

The System with category SYSTEM_EXTRACT is introduced to allow a collaboration between an OEM and a Supplier.[1] The OEM/Supplier Collaboration scenario is described in more detail in chapter 13.1.

The OEM is often only interested in the required functionality and the integration of the functionality into the System. Thus the OEM provides a basis for designing a subsystem, which is developed by the supplier. One difference to the System with category ECU_EXTRACT is that the System with category SYSTEM_EXTRACT is not fully decomposed and still needs to be refined before it forms the basis for the ECU configuration. Another difference is that a System with category SYSTEM_EXTRACT is not fixed to an EcuInstance.

[TPS_SYST_01123] System Extract may cover one or many EcuInstances ⌈The System with category SYSTEM_EXTRACT may cover one or many EcuInstances.⌋

The System with category SYSTEM_EXTRACT is using the same meta model elements as the System with category SYSTEM_DESCRIPTION. The System with category SYSTEM_DESCRIPTION is a special case of a System with category SYSTEM_EXTRACT. From the technical point of view there is no difference. The distinction is only made for the sake of Methodology [3].

In the System with category SYSTEM_EXTRACT the OEM strips all information from the System with category SYSTEM_DESCRIPTION that is not needed for the definition of the subsystem. There is one exception to this simple ”remove” rule: the communication mapping may need to be extended, which will be described in more detail in chapter 13.2.

[TPS_SYST_03000] Co-existing System with category SYSTEM_DESCRIPTION and System with category SYSTEM_EXTRACT ⌈In order to be able to handle one System with category SYSTEM_DESCRIPTION and one or several Systems with category SYSTEM_EXTRACT within the same workspace it shall be possible to provide different full qualified names to the elements of System with category SYSTEM_EXTRACT.⌋

When different Systems with various categories co-exist it is possible to define ViewMap and ViewMapSet between their elements according to [TPS_SYST_01136].

[1] Collaboration scenarios between different departments of an OEM are also supported by the System with category SYSTEM_EXTRACT. For the sake of simplicity such scenarios are not addressed here.

In contrast to the System with category ECU_EXTRACT the System with category SYSTEM_EXTRACT may contain CompositionSwComponentTypes. Empty CompositionSwComponentTypes in the System with category SYSTEM_EXTRACT represent subsystems that need to be refined by a Supplier. Figure 13.1 shows an example where a System with category SYSTEM_DESCRIPTION is stripped down to a subsystem.

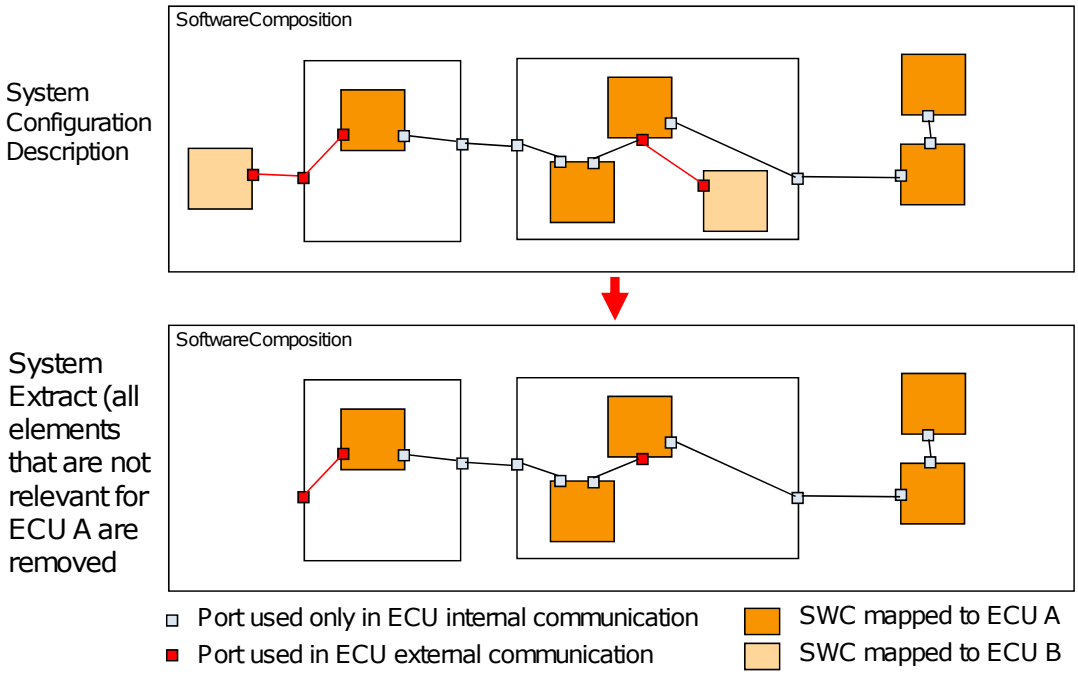

Figure 13.1: System Extract creation: irrelevant elements are removed from the System Description

13.1 OEM/Supplier Collaboration Scenario

In an important collaboration scenario, an OEM commissions a supplier to provide implementations of one or more functionalities to be integrated into an AUTOSAR system in the form of Application Components. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System VFB rather than the internal structure of such a component. On the other hand, the supplier, delivering both the component implementation in combination with the ECU it is destined to run on, may claim the internal structure of such a higher-level component contains substantial intellectual property, and hence may not want to disclose its internal works to the OEM.

Effectively, the use case can be described in the following manner:

- The OEM generates a System with category SYSTEM_EXTRACT from the System with category SYSTEM_DESCRIPTION. From the System with category SYSTEM_DESCRIPTION all elements are removed that are not relevant for the design of the subsystem, such as SW components or topology elements.

- The OEM can deliver a sub-structure of Software Compositions or even Atomic Software Components in the System with category SYSTEM_EXTRACT. But the System with category SYSTEM_EXTRACT can also contain empty Software Compositions. The OEM shall have the possibility to define only the outer shell of a Software Composition that is to implement a certain functionality. Such an empty CompositionSwComponentType does contain all the provided and required ports with the included ReceiverComSpecs and SenderComSpecs describing the requested component's outside communication needs. But it does not need to contain SwComponentPrototypes or SwConnectors at this stage.
- Such empty components are added to a System's VFB, the outside ports are connected with other components in the VFB. However, at this stage the inner structure of such CompositionSwComponentType can still be left empty.
- The System with category SYSTEM_EXTRACT contains the mapping of components to the target EcuInstances, including the empty compositions. Signal mappings affecting the empty compositions are targeting the CompositionSwComponentType's ports.
- The OEM delivers the System with category SYSTEM_EXTRACT to the Supplier.
- The Supplier adds the substructure to the empty CompositionSwComponentTypes by adding SwComponentPrototypes and SwConnectors. This once more leads to a hierarchical VFB, effectively the Supplier creates a local System Description for his subsystem.
- The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.
- The Supplier generates the System with category ECU_EXTRACT from his ECU-local system description. The resulting System with category ECU_EXTRACT does not include prototypes of type CompositionSwComponentType any longer.
- Based on this System with category ECU_EXTRACT the actual ECU configuration is done.

When the supplier receives the System with category SYSTEM_EXTRACT from the OEM he has basically two choices how to proceed:

1. The Supplier takes the System with category SYSTEM_EXTRACT of the OEM as the structural basis for the ECU development. In this case the following steps may follow:
   - The Supplier adds the substructure to the empty CompositionSwComponentTypes by adding SwComponentPrototypes and SwConnectors. This once more leads to a hierarchical VFB, effectively the Supplier creates a local System Description for his subsystem (System with category ECU_SYSTEM_DESCRIPTION).

- The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.

2. The Supplier creates an own structure to base the ECU development on System with category ECU_SYSTEM_DESCRIPTION and perform a view mapping between the OEM's System with category SYSTEM_EXTRACT and the System with category ECU_SYSTEM_DESCRIPTION. In this case the following steps may follow:
   - The Supplier develops an own structure how the ECU shall be designed but needs to respect the required outer boundary of the OEM's required communication behavior (ReceiverComSpecs and SenderComSpecs).
   - The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.

When the design of the System with category ECU_SYSTEM_DESCRIPTION is complete the following steps follow:

- The Supplier generates the System with category ECU_EXTRACT from his System with category ECU_SYSTEM_DESCRIPTION. The resulting System with category ECU_EXTRACT does not include prototypes of type CompositionSwComponentType any longer.
- Based on this System with category ECU_EXTRACT the actual ECU configuration is done.

13.2 Data Mapping in the System Extract

As mentioned before, there is a slight complication to the simple "remove" rule. This can be shown best with an example.

Example: Assume a simple topology with two EcuInstances A and B and three Pdus X (sent from A to B), Y (sent from B to A) and Z (sent from B to A) as shown in Figure 13.2.

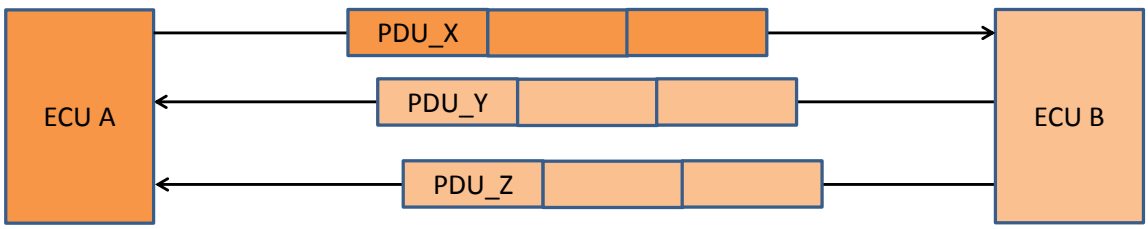

Figure 13.2: Example topology with two EcuInstances and three Pdus exchanged between them

Furthermore assume a composition of software-components realized by the meta-class CompositionSwComponentType as shown in Figure 13.3. It consists of six SwComponentPrototypes 'A1' to 'A3' (aggregated in composition 'SwCompA'),

'B1' / 'B2' (aggregated in composition 'SWCompB'), 'C1' (aggregated in composition 'SWCompAplusBplusC') and an empty composition 'SWCompC'.

The overall composition 'SWCompAplusBplusC' aggregates 'SwCompA', 'SWCompB', the empty 'SWCompC' and the SwComponentPrototype 'C1'.

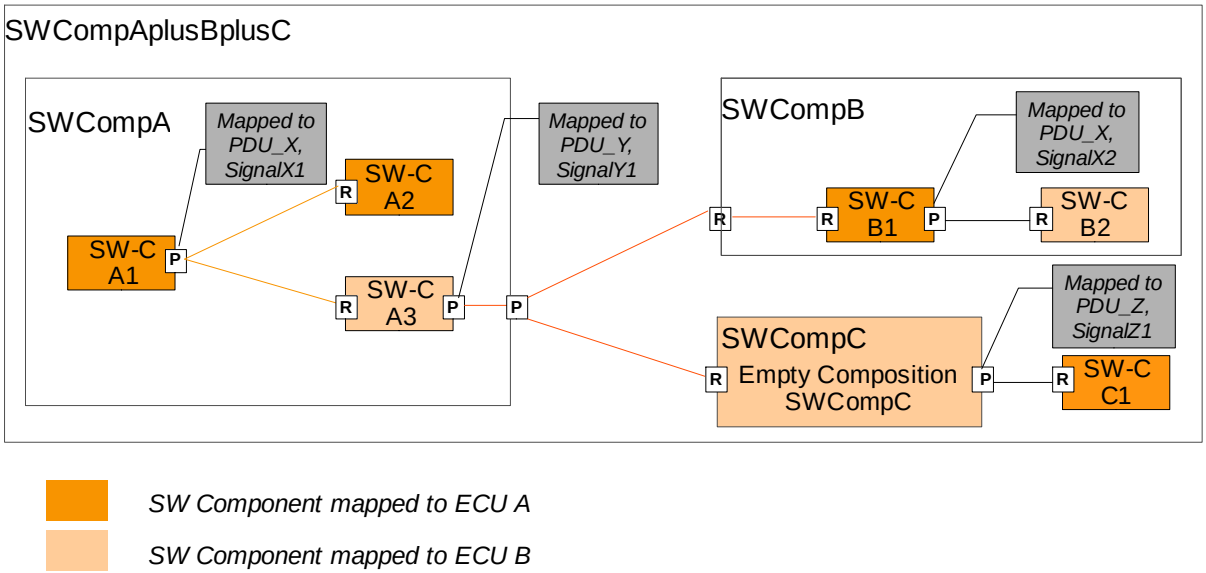

Figure 13.3: Example SW composition with mapping information

The atomic SwComponentPrototypes 'A1', 'A2', 'B1' and 'C1' are mapped to 'ECU A'. The atomic SwComponentPrototypes 'A3', 'B2' and the empty composition 'SWCompC' are mapped to 'ECU B'. The data sent from

- 'A1' to 'A3' is mapped to 'PDU_X', 'SignalX1',
- 'B1' to 'B2' is mapped to 'PDU_X', 'SignalX2' and
- 'A3' to 'B1' and 'A3' to 'SWCompC' is mapped to 'PDU_Y', 'SignalY1'
- 'SWCompC' to 'C1' is mapped to 'PDU_Z', 'SignalZ1'

As usual, the data mapping rules refer to the VariableDataPrototype in the PPortPrototype of the sending SW component. Note that DataMappings can be performed on compositions and on atomic SwComponentPrototypes as described in chapter 5.2.1. [2]

Figure 13.4 shows how the System extract for ECU A and for ECU B of this SW composition would look like: Only those elements are included that are relevant for the subsystem.

[2]Data mapping is allowed on empty compositions and on compositions that contain atomic SwComponentPrototypes.

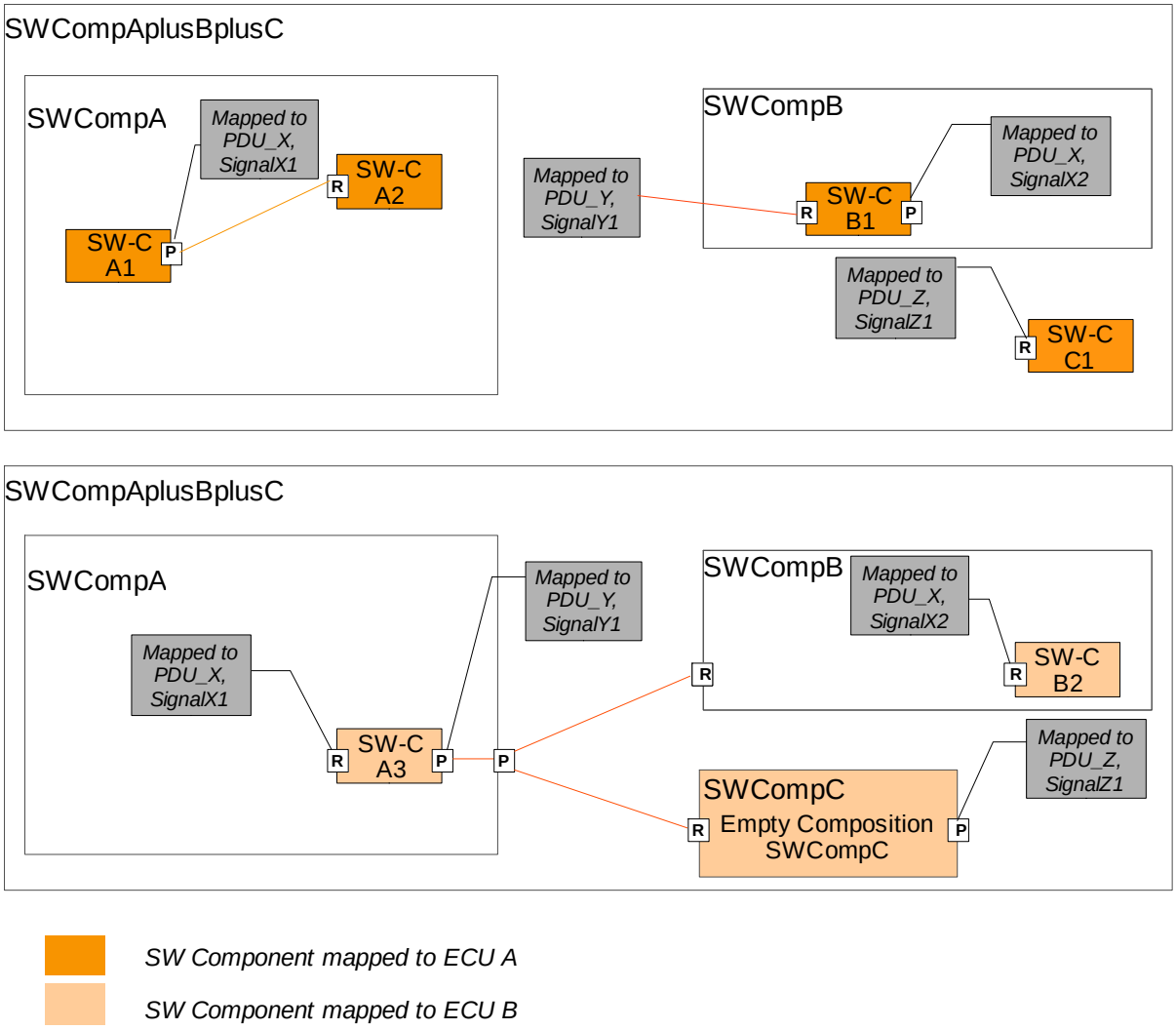

Figure 13.4: Example System extract for ECU A (upper figure) and ECU B (lower figure) of above introduced composition

In both figures all `SwComponentPrototype`s and compositions that are mapped onto the `EcuInstance` are included. The `SwConnector` between these `SwComponentPrototype`s are also included. Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

`SwConnector`s that were used to connect to SW components that are not included in the System Extract are not included. Instead, the mapping to an `ISignal` in a `Pdu` is used to identify the source/destination of that data.

The problem that new mapping rules have to be added arises for example in the System Extract for ‘ECU A’ with the mapping to ‘PDU Y’, ‘SignalY1’: Since SW component ‘A3’, which was referenced in the original mapping, is no longer included, the data mapping needs a new data element in a port to reference to. In the example, it is the required port of ‘B1’, so that the Supplier has the information that B1 receives the data via ‘PDU Y’.

13.3 SW component inclusion and top level data mapping

In section 13.2 the approach is to provide the DataMapping on the PortPrototypes of the SwComponentPrototypes which are mapped to one EcuInstance. Since the granularity of mapping SwComponentPrototypes to EcuInstances is possible for individual atomic SwComponentPrototypes this approach may result in many DataMappings from different software component PortPrototypes to the same SystemSignal (depending where in the hierarchical structure they are located).

An alternative approach is to provide the complete communication information of the whole System Extract on the RootSwCompositionPrototype and perform the DataMapping on the PortPrototypes of the RootSwCompositionPrototype only. This approach is illustrated in figure 13.5.

PortPrototypes are created on the RootSwCompositionPrototype representing the external communication of this EcuInstance. DelegationSwConnectors are created to establish the communication of the external software components with the software components inside the local EcuInstance.

In figure 13.5 the software components X, Y and Z are mapped to remote EcuInstances. Their communication needs are collected in PortPrototypes on the RootSwCompositionPrototype and the communication is delegated via SwConnectors inside the hierarchical software component structure.

In this example the approach for X and Y is trivial since there are only some DelegationSwConnectors required to connect the PortPrototypes of the RootSwCompositionPrototype with the PortPrototypes of the respective SwComponentPrototypes.

But for SwComponentPrototype Z the approach needs to be extended, because the communication on system level is designed to happen inside the composition V. In this case the communication needs to be delegated out of the composition (creation of DelegationSwConnectors inside the composition V) to be visible in the RootSwCompositionPrototype. Then again the approach of connection to the RootSwCompositionPrototype can be applied.

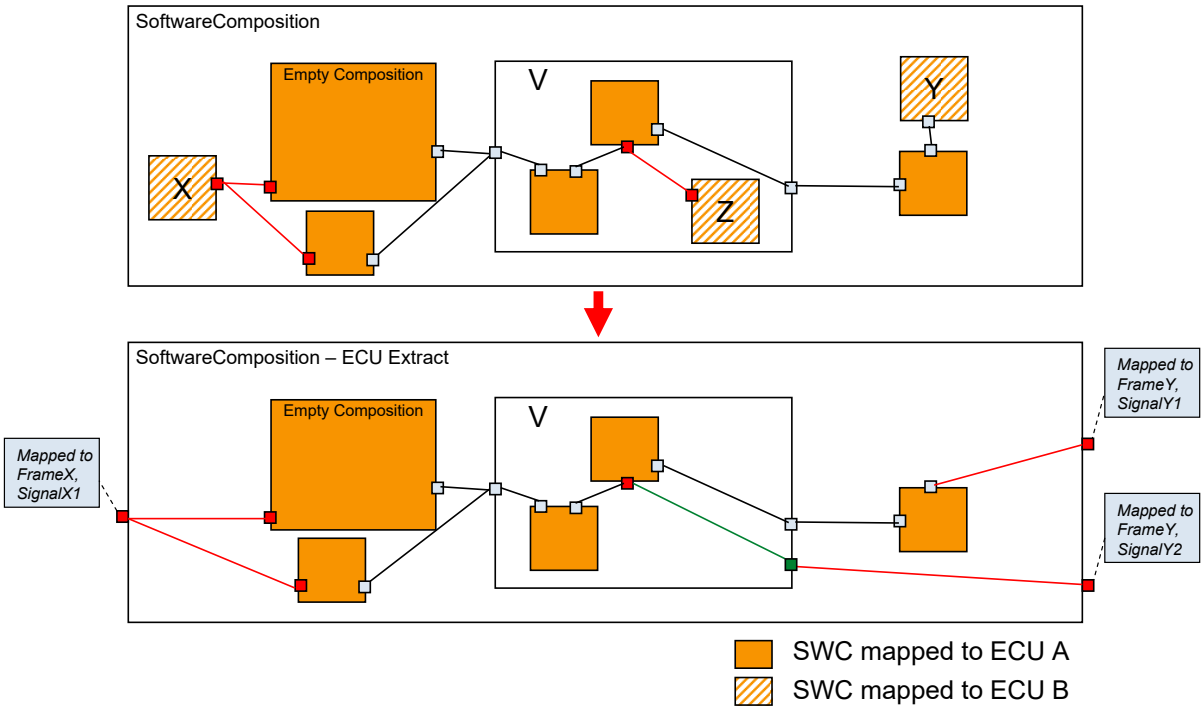

Figure 13.5: Example with software components mapped to two ECUs

13.4 Port-Interface Mapping in the System Extract

A similar problem as the one with DataMappings described in chapter 13.2 and chapter 13.3 exists for the PortInterfaceMappings as well. To illustrate this Figure 13.6 depicts an example with software components mapped to two different ECUs.

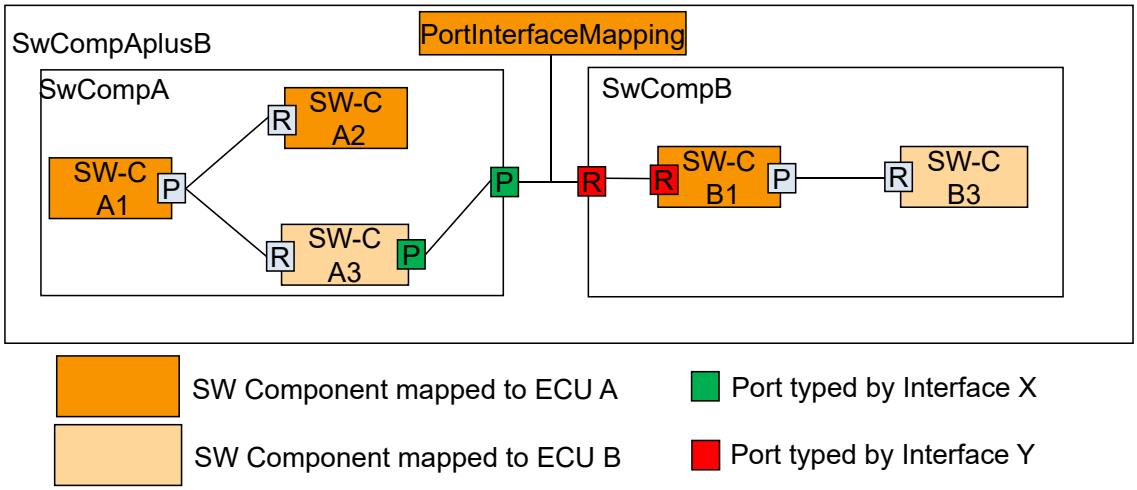

Figure 13.6: Example with software components mapped to two ECUs

Hereby the PPortPrototype typed with PortInterface *X* of *SWCompA* is connected with the RPortPrototype typed with PortInterface *Y* of *SWCompB* by means of an AssemblySwConnector. This AssemblySwConnector has an attached PortInterfaceMapping to perform a mapping between the elements (see chapter 4.3.1.5 of [4]) of the two otherwise incompatible PortInterfaces *X* and *Y*.

A System Extract for ECU *A* is now created by applying the approach described in chapter 13.3, i.e., by providing the complete communication information of the whole System Extract on the RootSwCompositionPrototype and performing the DataMapping on the PortPrototypes of the RootSwCompositionPrototype only.

When doing this however the following two additional things have to be considered:

- The PortInterfaceMapping shall be preserved during this process
- The information about the PortInterfaces referenced by the PortPrototypes connected by the AssemblySwConnector referencing the PortInterfaceMapping shall be preserved during this process

Just as in the approach described in chapter 13.3 PortPrototypes are created on the RootSwCompositionPrototype representing the external communication of this EcuInstance. The RPortPrototypes however are not typed by the PortInterface *X* of the RPortPrototypes of the SwComponentPrototypes inside *ECU A* (*SWCompB* in the example) but by the PortInterface *Y* of the PPortPrototype which was connected to the RPortPrototypes by means of the AssemblySwConnector. Afterwards the just like in the approach described in chapter 13.3 DelegationSwConnectors are created to connect the PortPrototypes

of the RootSwCompositionPrototype with the corresponding RPortPrototypes of the SwComponentPrototypes inside *ECU A*.

This however yields a DelegationSwConnector between RPortPrototype typed by PortInterface *Y* (which has been created on the RootSwCompositionPrototype) and the RPortPrototype typed by PortInterface *X* of *SWCompB*. In order to perform a mapping between the elements of these otherwise incompatible interfaces, the PortInterfaceMapping which has initially been referred to by the AssemblySwConnector needs to be referenced by the DelegationSwConnector.

The final result of this process in depicted in Figure 13.7

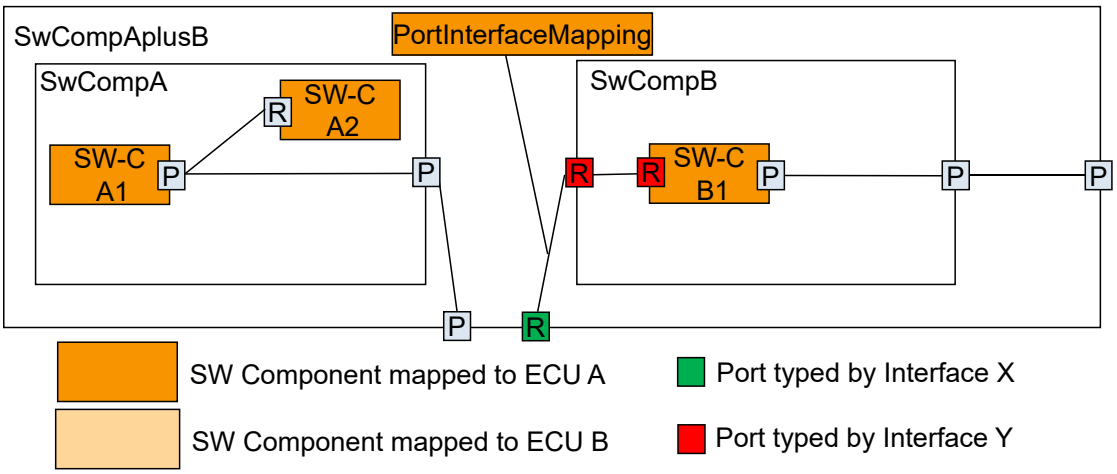

Figure 13.7: Example with software components mapped to two ECUs

14 ECU Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR System with category ECU_EXTRACT, based on Meta Model elements contained in the System Template and Software Component Template.

The System with category ECU_EXTRACT represents the view of one specific EcuInstance onto the overall System with category SYSTEM_DESCRIPTION. The System with category ECU_EXTRACT forms the basis for configuring that particular EcuInstance in focus.

For instance, RTE configuration fundamentally depends on the number and types of SwComponentPrototypes deployed onto the EcuInstance; Services are configured according to those Software Components' ServiceNeeds; the COM-stack BSW modules will be configured considering the EcuInstance's participation in the overall System Network Topology and Communication.

[TPS_SYST_01139] Ecu Extract derived from System Description or System Extract covers exactly one EcuInstance ⌈The System with category ECU_EXTRACT shall only contain the subset of information derived from the System with category SYSTEM_DESCRIPTION or System with category SYSTEM_EXTRACT relevant for configuring the targeted EcuInstance.⌋

In order to keep ECU configuration focused and manageable despite the complexity of a full System Configuration, all other information shall be stripped from the System with category SYSTEM_DESCRIPTION or from the System with category SYSTEM_EXTRACT when creating the System with category ECU_EXTRACT.

AUTOSAR VFB Descriptions naturally form hierarchies of CompositionSwComponentTypes. Consequently, in the System Configuration the SWC-related information for different EcuInstances is not separated but in general is intermingled. In contrast, for the task of ECU configuration (RTE configuration, Service Configuration, Measurement and Calibration) a hierarchically "flat view" on the SwComponentPrototypes running on the EcuInstances is preferable over a hierarchical view, which is more favored by application-software development. Thus, deriving an System with category ECU_EXTRACT actually is a model transformation, following a set of rules described in the following sections.

[constr_9333] FibexElements in ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈Each FibexElement that is used in the ECU_EXTRACT shall be referenced by the System element in the role fibexElement.⌋

[TPS_SYST_02313] Ecu Extract derived from ECU_SYSTEM_DESCRIPTION covers an EcuInstance ⌈The System with category ECU_SYSTEM_DESCRIPTION defines the content of a single EcuInstance and the same is true for the derived

ECU_EXTRACT. The derived ECU_EXTRACT is flattened and does not contain any hierarchies of CompositionSwComponentTypes.⌋

[TPS_SYST_02314] Ecu Extract derived from SW_CLUSTER_SYSTEM_DESCRIPTION covers a subset of an EcuInstance ⌈The System with category SW_CLUSTER_SYSTEM_DESCRIPTION defines the content of a single CpSoftwareCluster and the same is true for the derived ECU_EXTRACT. The derived ECU_EXTRACT is flattened and does not contain any hierarchies of CompositionSwComponentTypes.⌋

As System- and ECU development typically happens in iterations, the use case of repeatedly extracting the information from an incrementally changing System Configuration needs to be considered. In particular, it shall be possible to detect changes between consecutively generated ECU extracts in order to selectively update the existing ECU configuration (14.6).

AUTOSAR supports the definition and consequently the handling of Variability in the System Configuration. According to the specified binding time associated with a particular VariationPoint, typically some of these variants will already be resolved at the time of a System with category ECU_EXTRACT. If however the binding time occurs in a later stage of the AUTOSAR methodology, i.e. during ECU Configuration or later, the variability needs to be carried over to the System with category ECU_EXTRACT. This also holds true for Variation points that ultimately are resolved at system configuration time but affect post-build configuration parameters. (14.7)

The System with category ECU_EXTRACT logically forms one entity. Therefore, for ease of readability the rest of the chapter assumes just one file, “the XML file”. However, it explicitly is allowed to split the System with category ECU_EXTRACT over several files.

14.1 Topology

Only those Topology elements relevant for the EcuInstance in scope are taken over from the System with category SYSTEM_DESCRIPTION into the System with category ECU_EXTRACT.

- The System with category ECU_EXTRACT is always associated with exactly one EcuInstance. Therefore exactly one EcuInstance is included along with all classes included in EcuInstance by composition: CommunicationControllers and CommunicationConnectors with all their CommConnectorPorts.
- A CommunicationCluster is included along with all its PhysicalChannels if at least one PhysicalChannel is used by the EcuInstance. In other words, if at least one of the included CommunicationConnectors is referenced by any of a CommunicationCluster’s PhysicalChannels, the whole CommunicationCluster and all its PhysicalChannels are included.

- From the used PhysicalChannels, only those FrameTriggerings, PduTriggerings, ISignalTriggerings shall be included that are used by the EcuInstance, e.g. they are associated with a FramePort, IPduPort, ISignalPort belonging to one of the EcuInstance's CommunicationConnectors. *Note:* Including just a subset of a PhysicalChannel's FrameTriggerings, PduTriggerings, ISignalTriggerings is possible without changing the PhysicalChannel itself because of the ≪splitable≫ stereotype applied on the PhysicalChannel / FrameTriggering, PduTriggering, ISignalTriggering composition.

As the Topology elements are not modified when taken over into the System with category ECU_EXTRACT, their package structure and short names are not touched (see section 14.5.1).

14.2 Top-level Software Composition

In the System with category SYSTEM_DESCRIPTION the application software composition is hierarchic by nature as described in chapter 4. When mapping SwComponentPrototypes onto concrete EcuInstances using the SwcToEcuMapping class (section 5.1.1), either SwComponentPrototypes of type AtomicSwComponentType, or SwComponentPrototypes of type CompositionSwComponentType are deployed onto one specified EcuInstance.

In order to obtain this ECU-centric view, the hierarchical structure of the System with category SYSTEM_DESCRIPTION needs to be transformed into a 1-layer representation, where one distinguished CompositionSwComponentType hosts all SwComponentPrototypes of type AtomicSwComponentType to run on the EcuInstance. In the System with category ECU_EXTRACT the resulting RootSwCompositionPrototype is a flat structure where the included SwComponentPrototypes become real SWC instances, reflecting the actual resource needs on the targeted EcuInstance.

[TPS_SYST_01140] Ecu Extract contains only SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype ⌈The System with category ECU_EXTRACT only contains SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype which are effectively mapped onto the EcuInstance in focus.⌋

The transformation from hierarchical to flat Software Component structure includes a number of steps, to be performed per ECU. The list below outlining this process assumes that the extraction is done for the first time; if an System with category ECU_EXTRACT already exists from a previous development cycle, the extract shall merely be updated instead of created; for more details on iterative development see section 14.6.

- Create the one CompositionSwComponentType which will represent the ECU's SW subsystem (in further steps referred to as ECU flat view)

- To this ECU flat view, add a SwComponentPrototype for each instance of any AtomicSwComponentType mapped onto the EcuInstance. Copy all the identifiable information from the originating SwComponentPrototype, but assign an unique short name to the new element. The newly created SwComponentPrototypes are typed by the original AtomicSwComponentType.
- Unroll the connector paths leading to and from the included components:
  - For ECU internal communication, use AssemblySwConnector to connect PortPrototypes.
  - For ECU external communication, add delegated PortPrototypes to the ECU flat view CompositionSwComponentType. The delegated PPortPrototypes are allowed to contain a subset of the data that is defined by the inner PPortPrototype as specified by [TPS_SYST_02384]. The delegated RPortPrototypes are allowed to contain a superset of the data that is defined by the inner RPortPrototypes as specified by [TPS_SYST_02385]. Each delegated PortPrototype shall be connected via a DelegationSwConnector with PortPrototypes of the included SwComponentPrototypes that are used for the external communication.
  - VariableDataPrototypes and ClientServerOperations of the delegated PortPrototypes are mapped to SystemSignals.
- If the System with category SYSTEM_DESCRIPTION prescribes an Implementation for a SwComponentPrototype by using SwcToImplMapping, a corresponding constraint needs to be created in the System with category ECU_EXTRACT of the targeted EcuInstance. The SwcToImplMapping's component reference needs to be adjusted to the flat representation, while maintaining the original reference to the Implementation.
- Only ComSpecs on the PortPrototypes of atomic software components are relevant for the RTE generator (see [TPS_SWCT_01568]). The existence of ComSpecs on composition level can be taken into account for setting the values on the atomic level as the atomic level gets created. On the other hand there is no obligation to respect the ComSpec settings on the composition level for the creation of the atomic level. Finally the approach for the creation of ComSpec values on atomic level depends on OEM preferences.

[TPS_SYST_02384] Sending inner Ports may contain the superset of data provided on the outer delegation Port ⌈The outer PPortPrototype on the CompositionSwComponentType is allowed to contain a subset of the provided data that is defined by the inner PPortPrototype of the included SwComponentPrototype that is connected via a DelegationSwConnector to the outer PPortPrototype.⌋

[TPS_SYST_02385] Receiving outer Ports may contain the superset of data delegated to the inner Ports ⌈The inner RPortPrototype on the SwComponentPrototype that is included in the CompositionSwComponentType is allowed to

contain a subset of the received data that is defined by the outer RPortPrototype of the CompositionSwComponentType that is connected via a DelegationSwConnector to the inner RPortPrototype.

Figure 14.1 illustrates the process of flattening the hierarchical Software Composition into an ECU Flat View representation, as outlined in the previous paragraphs. The following sections explain the concrete transformation steps in more detail.

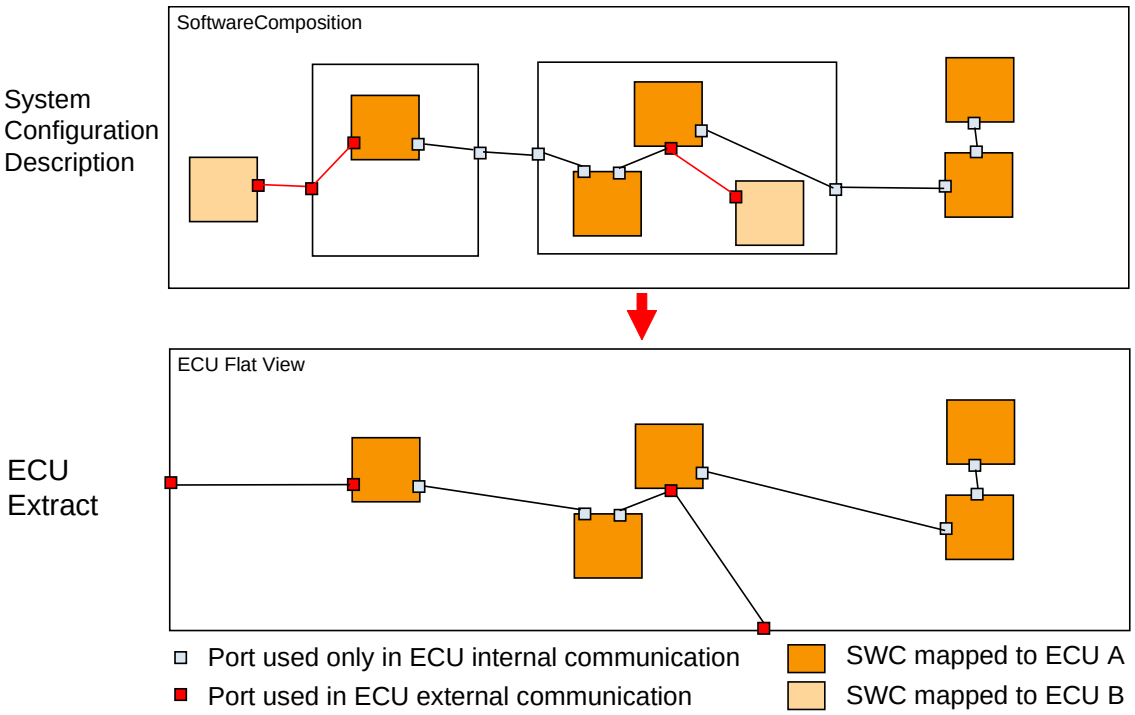

Figure 14.1: Flattening of a hierarchic Software Composition into an ECU Flat View, and the distinction between ports used in internal and those used in external communication.

Please note that instantiation specific scheduling of runnables shall be maintained when generating a System with category ECU_EXTRACT. This maintenance covers the rewrite of the instanceRef to the RTEEvent respectively the aggregation of the instantiationRTEEventProps to the next CompositionSwComponentType.

14.2.1 ECU Flat view

The first step of extracting the ECU specific Software View is the creation of a new CompositionSwComponentType (further referred to as ECU flat view). This new element serves as a container for collecting all SwComponentPrototypes of type AtomicSwComponentType deployed on the EcuInstance. In order to include the ECU flat view into the actual System with category ECU_EXTRACT, the System shall have its child class RootSwCompositionPrototype pointing to this ECU flat view.

Next, all SwcToEcuMappings present in the System with category SYSTEM_DESCRIPTION need to be analyzed according to the precedence rules (Section 5.1.1) in order to establish the exact set of AtomicSwComponentType instances to be included on this EcuInstance.

For each of these component instances, regardless of their order of depth in the System Configuration Description's Component hierarchy, exactly one SwComponentPrototype shall be created in the ECU flat view CompositionSwComponentType. The new element's description and type information shall be taken over from the original SwComponentPrototype as present in the System with category SYSTEM_DESCRIPTION. As an important exception to this rule, the SwComponentPrototype's shortName shall be unique in the name space formed by the ECU flat view.

The ParameterSwComponentTypes are handled in the same way. For ParameterSwComponentTypes there are SwConnectors defined but no communication is involved. For more details see [TPS_SWCT_01422] in the SwComponentTemplate [4]. For ServiceProxySwComponentTypes one small difference applies: This component type can be instantiated at most once per EcuInstance and for a given prototype in the System, instances on several EcuInstances can be created. The replication of ServiceProxySwComponentTypes on several EcuInstances does not require any special treatment of their communication properties.

14.2.2 Internal Communication

When flattening the RootSwCompositionPrototype for the System with category ECU_EXTRACT, not only all of the ECU's Software Components are to be collected in the ECU flat view, but also any connection existing between PortPrototypes of the included SwComponentPrototypes needs to be projected onto the same RootSwCompositionPrototype.

In the hierarchical RootSwCompositionPrototype, communication between Software Components is specified by a combination of AssemblySwConnectors and DelegationSwConnectors. Several DelegationSwConnectors may be combined in case of a multiple-level delegation, however there will always be exactly one AssemblySwConnector on the outermost CompositionSwComponentType the port is delegated to.

In the ECU flat view, any such number of stringed together SwConnectors effectively connecting two PortPrototypes of SwComponentPrototypes mapped to the same EcuInstance are resolved to exactly one AssemblySwConnector per connected port pair. As there are no additional levels of "inner SwComponentPrototypes". DelegationSwConnectors are only used to display the outside communication of an ECU in the ECU flat view.

[constr_3019] In the flat ECU extract each required interface shall be satisfied by connected provided interfaces

Imposition time: CP: IT_EcuExt

⌈In case of the flat System with category ECU_EXTRACT all VariableDataPrototypes specified by the SenderReceiverInterface of the RPortPrototype need to be supplied by some of the PPortPrototypes being connected with SwConnectors.⌋

For the System with category SYSTEM_DESCRIPTION, the Software Component Template Specification [4] allows a CompositionSwComponentType's outer PortPrototype to be connected to more than one inner port, observing a set of compatibility rules between the outer and the inner port's SenderReceiverInterfaces. Such a "merge" and "split" functionality for mixing VariableDataPrototypes is used to limit the number of SwConnectors required to connect PortPrototypes on higher VFB levels and thus reduce complexity in the wiring of such higher-level CompositionSwComponentTypes. On the other hand this means that an AssemblySwConnector in a hierarchical VFB may expand to more than one Port-Port pair. Naturally, in the ECU flat view such "hidden" additional connections need to be made explicit by unrolling them into concrete AssemblySwConnectors.

Additionally PassThroughSwConnector may be used to map PortInterface elements between require and provide outer ports of CompositionSwComponentTypes in order to use RTE features for mapping or conversion instead of real software components. The following paragraph suggests a way how such an unrolling of SwConnectors may be accomplished.

Starting with the top-level RootSwCompositionPrototype indicating the outermost CompositionSwComponentType, the hierarchical software model of SwComponentPrototypes is recursively iterated; for each prototype of CompositionSwComponentType, all its AssemblySwConnectors are being iterated. For each such found AssemblySwConnector both connector ends are evaluated for DelegationSwConnectors further delegating the connection: In order to consider the use cases of signal "merge" and "split", all possible communication partners need to be identified, recursively following DelegationSwConnectors in both directions. For each identified pair of PPortPrototypes and RPortPrototypes *actually exchanging* Information one AssemblySwConnector will be created in the ECU flat view.

In case that a string of SwConnectors started by AssemblySwConnector connects - directly or via DelegationSwConnectors - to a PassThroughSwConnector the SwConnector string is conjunct with the SwConnector string of the other end of the PassThroughSwConnector. Please note that the "merge" and "split" capability of DelegationSwConnectors and PassThroughSwConnectors requires an individual treatment of the single PortInterface elements for the evaluation of the SwConnector string.

The following rules shall be followed when PortInterfaceMappings are converted for the flat view. PortInterfaceMappings supports the connection of Ports typed

by two different PortInterfaces with unequal named PortInterface elements. More details can be found in [4].

- When unrolling a string of SwConnectors into a single SwConnectors all compatibility rules and PortInterfaceMappings of the individual SwConnector need to be considered for determining which VariableDataPrototypes are being transferred between provider and requester. If VariableDataPrototypes are to be filtered out a PortInterfaceMapping shall be provided to the flatten connector such that only the transferred VariableDataPrototypes are included in the mapping.
- When unrolling a string of SwConnectors into a single SwConnector all of the PortInterfaceMappings of the individual SwConnectors need to be considered for combining them into a single PortInterfaceMapping to be associated with a new SwConnector.

14.2.3 External Communication

In a System with category SYSTEM_DESCRIPTION , whenever two SwComponentPrototypes are specified to communicate across EcuInstances, the details of this communication need to be fully specified: VariableDataPrototypes of SenderReceiverInterfaces and ClientServerOperations of ClientServerInterfaces are mapped onto SystemSignals as carriers of information transported across the network. According to 5.2, each instance of a AutosarDataPrototype that is to be sent over AUTOSAR COM shall be mapped exactly once onto its individual SystemSignal, regardless of how many components receive the information or over how many PhysicalChannels the SystemSignal is transported.

As described above, deriving the System with category ECU_EXTRACT from System with category SYSTEM_DESCRIPTION or from System with category SYSTEM_EXTRACT means that all SwComponentPrototypes to be included in the Ecu extract are recreated in an ECU flat view. Consequently, each DataMapping concerning a SwComponentPrototype to be mapped onto the EcuInstance requires that a corresponding DataMapping be created in the System with category ECU_EXTRACT.

The ECU flat view contains delegated PortPrototypes to display the outside communication of an EcuInstance. VariableDataPrototypes and ClientServerOperations of these delegated PortPrototypes are mapped to SystemSignals. The original instance references indicating the mapped AutosarDataPrototype need to be adjusted to the new “flat” location in the ECU flat view.

While for the System with category SYSTEM_DESCRIPTION it is sufficient to describe DataMappings only on the provider side, the System with category ECU_EXTRACT additionally requires such DataMappings on the requiring side’s ports. In this case, a new DataMapping maps to the existing SystemSignal, previously defined in the System with category SYSTEM_DESCRIPTION on the provider side.

This is explained in more detail in figure 14.6, that is a continuation of the example from figure 13.3 in chapter 13.2.

To derive an ECU_EXTRACT from a System with category SYSTEM_DESCRIPTION or SYSTEM_EXTRACT unambiguous ClientServerToSignalMappings are required for inter-ECU n:1 client-server communication. In particular the communication path from the server to each client shall be uniquely mapped.

In this context, "communication path" encompasses the set of delegation/assembly connectors that connect the server (provide-port on SWC) through to the client (require-port on SWC).

[constr_3264] Server side ClientServerToSignalMappings in case of a n:1 inter-ECU client-server communication

Imposition time: CP: IT_SysDesc

⌈If within the System with category SYSTEM_DESCRIPTION or SYSTEM_EXTRACT the ClientServerToSignalMappings for inter-ECU n:1 client-server communication are placed on the provider (server) side, then each of these ClientServerToSignalMappings shall (in the hierarchy of SwComponentPrototypes) refer to a "unique communication path" w.r.t. the EcuInstances the client SwComponentPrototypes are mapped to.⌋

Note: A "unique communication path" has the property that, starting from the ClientServerOperation of a PortPrototype, a sequence of DelegationSwConnectors and AssemblySwConnectors leads to the client side and terminates at either at most one PortPrototype that is owned by the AtomicSwComponentType of the client's SwComponentPrototype or, if the path terminates at more than one PortPrototype, then the following shall hold: The clients' SwComponentPrototypes typed by AtomicSwComponentTypes owning these PortPrototypes shall be mapped to the same EcuInstance and the client identifier is used to distinguish the different clients (see [TPS_SYST_01087]).

The following example scenarios will show at which PortPrototypes the ClientServerToSignalMappings are allowed to be specified in a System with category SYSTEM_DESCRIPTION or SYSTEM_EXTRACT to derive an ECU_EXTRACT.

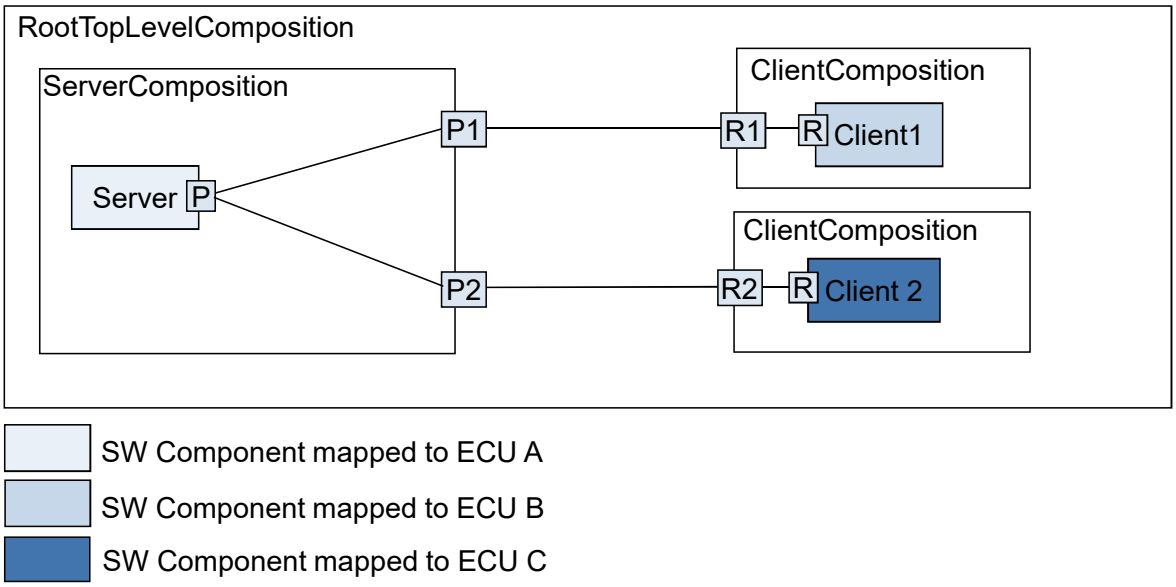

Figure 14.2: Client Server Scenario 1

For the scenario described in figure 14.2 the following statements apply:

- `ClientServerToSignalMapping`s for the provide-port Server.P are ambiguous and thus [constr_3243] exists to forbid this situation.
- `ClientServerToSignalMapping`s are permitted for ClientComposition.R1/ClientComponsition.R2 and Client1.R/Client2.R (client-side) or for ServerComposition.P1/ServerComposition.P2 (provider-side) since there is no ambiguity.

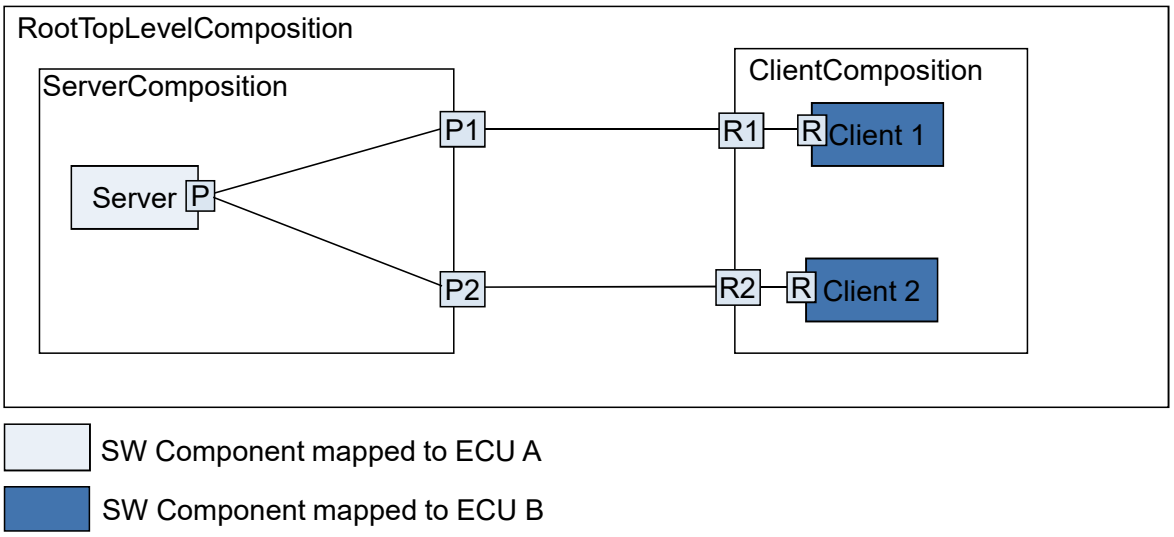

Figure 14.3: Client Server Scenario 2

For the scenario described in figure 14.3 the following statements apply:

- ClientServerToSignalMappings for the provide-port Server.P are not ambiguous (since although there is fork in the communication path, both sub-paths end up at the same ECU).
- ClientServerToSignalMappings are permitted for ClientComposition.R1/ClientComponsition.R2 and Client1.R/Client2.R (client-side) or for ServerComposition.P1/ServerComposition.P2 (provider-side) since there is no ambiguity.

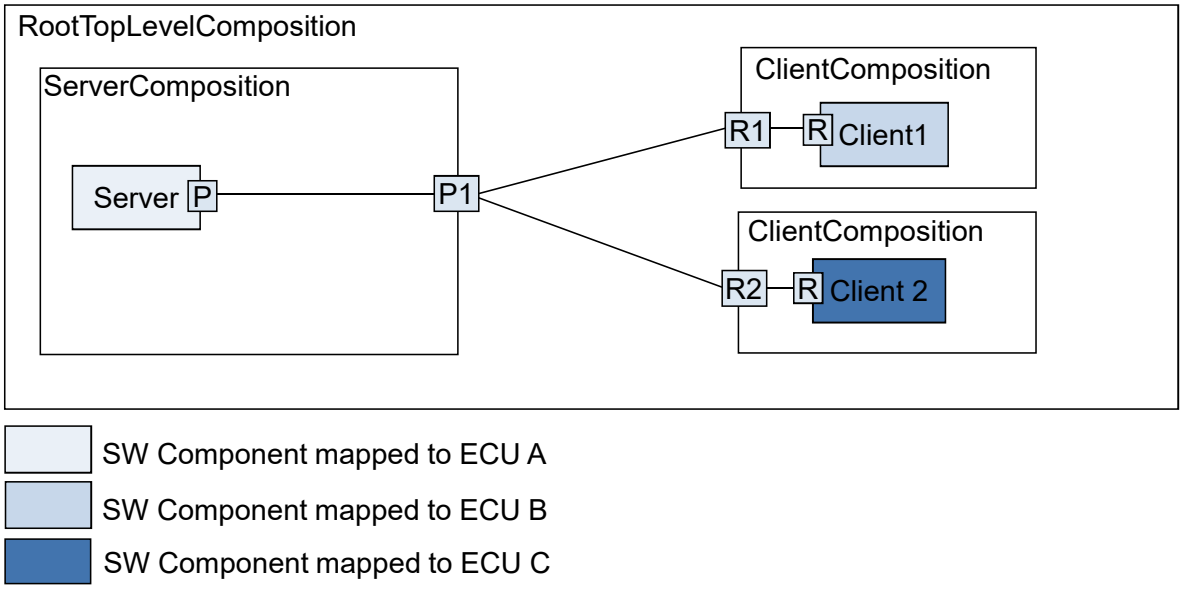

Figure 14.4: Client Server Scenario 3

For the scenario described in figure 14.4 the following statements apply:

- ClientServerToSignalMappings for the provide-ports Server.P and ServerComposition.P1 are ambiguous and thus [constr_3243] exists to forbid this situation.
- ClientServerToSignalMappings are permitted for ClientComposition.R1/ClientComposition.R2 and Client1.R/Client2.R (client-side) since there is no ambiguity.

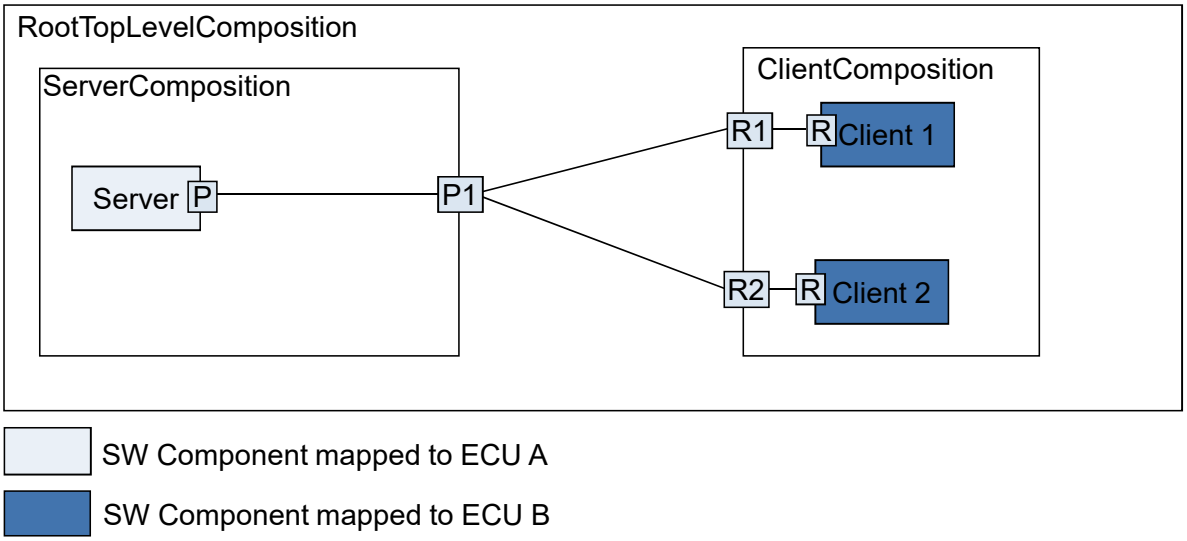

Figure 14.5: Client Server Scenario 4

For the scenario described in figure 14.5 the following statements apply:

- ClientServerToSignalMappings for the provide-ports Server.P and ServerComposition.P1 are not ambiguous (since although there is fork in the communication path, both sub-paths end up at the same ECU).
- ClientServerToSignalMappings are permitted for ClientComposition.R1/ClientComponsition.R2 and Client1.R/Client2.R (client-side) or for ServerComposition.P1/ServerComposition.P2 (provider-side) since there is no ambiguity.

[TPS_SYST_01145] PortInterfaceMappings in the ECU Extract ⌈In the System with category ECU_EXTRACT the missing PortInterfaceMappings on the complementary side needs to be supplemented to DelegationSwConnectors.⌋

[constr_5328] Ecu Extract shall only contain outerPort DataMappings

Imposition time: CP: IT_EcuExt

⌈The System with category ECU_EXTRACT shall only contain DataMappings for VariableDataPrototypes, ClientServerOperations or Triggers that are referenced in the context of a PortPrototype of the SwComponentType that in turn is referenced by the RootSwCompositionPrototype.⌋

In other words the ECU_EXTRACT shall not contain any DataMappings that are defined in the context of SwComponentPrototypes that are included in the RootSwCompositionPrototype. Only the outerPort DataMappings defined in the context of RootSwCompositionPrototype are considered by the RTE.

Figure 14.6 shows how the System with category ECU_EXTRACT for ECU A of the SW composition that is defined in figure 13.3 would look like: Only those

SwComponentPrototypes are included that are mapped to ECU A. The hierarchy present in the System with category SYSTEM_DESCRIPTION has been flattened into CompositionSwComponentType 'EcuAFlatView', including newly created SwComponentPrototype 'A1E', 'A2E', 'B1E' and 'C1E' for the component instances mapped to ECU A.

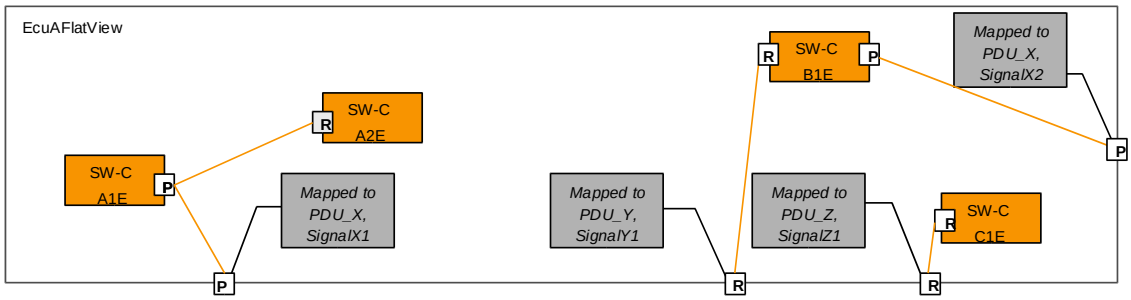

Figure 14.6: Example ECU extract for ECU A of above introduced composition

The SwConnectors to the outside ports (ECUFlatView composition ports) and SwConnectors that represent intra-ECU communication (in our example, only 'A1E' to 'A2E') are included. The VariableDataPrototypes and ClientServerOperations in the outside ports are mapped to SystemSignals. This DataMapping and the communication description is used to identify the source/destination of that data.

Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

The problem that new mapping rules have to be added arises with the mapping to 'PDU_Y', 'SignalY1': Since SW component 'A3', which was referenced in the original mapping, is no longer included, the DataMapping needs a new VariableDataPrototype in a PortPrototype to reference to. In the example, the data of the required port of 'B1E' is referenced, so that the ECU generator has the information that 'B1E' receives the data via 'PDU_Y'.

14.2.4 Port Groups

A SwComponentType can optionally define PortGroups which allow to group PortPrototypes according to logical criteria, e.g. according to shared communication resources (see [4]). A PortGroup of a CompositionSwComponentType can be linked to "inner" PortGroups of the aggregated SwComponentPrototypes. Since the main purpose of this grouping is to configure the behavior of mode managers on an EcuInstance, this information shall be preserved and broken down into the System with category ECU_EXTRACT.

The resulting CompositionSwComponentType in the ECU flat view will contain a set of PortGroups which refer to the linked inner port groups of the SwComponentPrototypes with AtomicSwComponentType. To get to this result, the following steps shall be applied in the extraction process:

- Recursively ignore all PortGroups in CompositionSwComponentTypes in the hierarchical structure, which are not linked to any inner groups to be mapped on this EcuInstance.
- In the remaining structure of linked PortGroups find out the top level PortGroups (i.e. which are not referred by any higher level PortGroup on this EcuInstance) and put an element representing each top level PortGroup into the CompositionSwComponentType of the ECU flat view. This can result in name conflicts, which should be resolved by a suitable algorithm.
- Link these top level PortGroups to the inner PortGroups of the atomic component instances of the flat view according to the links found in the hierarchical structure. Naturally, the top level PortGroups in the ECU flat view are not directly referring any PortPrototypes and due to the first step they should be linked to at least one inner PortGroup.
- The PortGroups in SwComponentPrototypes with an AtomicSwComponentType on the EcuInstance should be unchanged.

14.2.5 Service Needs

Each software component might need services which are provided by the ECU Basic Software through AUTOSAR Services. ServiceNeeds are used to provide detailed information what the software component expects from the AUTOSAR Services when integrated on an actual ECU (see SWComponentTemplate [4] for more details). If an ECU Extract is created the following rules apply to the existing ServiceNeeds:

[constr_3068] DoIpPowerModeStatusNeeds in the category ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈If and only if DoIP (i.e. any of the subclasses of DoIpServiceNeeds are present) is used on an Ecu then the DoIpPowerModeStatusNeeds shall exist exactly once in a System of category ECU_EXTRACT.⌋

[constr_1265] DoIpGidSynchronizationNeeds can only exist once per ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈Within the context of one System of category ECU_EXTRACT, there can only be at most one DoIpGidSynchronizationNeeds.⌋

[constr_1266] DoIpGidNeeds can only exist once per ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈Within the context of one System of category ECU_EXTRACT, there can only be at most one DoIpGidNeeds.⌋

[constr_1267] DoIpActivationLineNeeds can only exist once per ECU_EXTRACT

Imposition time: CP: IT_EcuExt

⌈Within the context of one System of category ECU_EXTRACT, there can only be at most one DoIpActivationLineNeeds.⌋

[constr_3083] Exactly one AtomicSwComponentType on an EcuInstance may use GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange

Imposition time: CP: IT_EcuExt

⌈The Dem only supports exactly one AtomicSwComponentType using GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange on one EcuInstance.⌋

[constr_3084] Service port in the role PowerTakeOff

Imposition time: CP: IT_EcuExt

⌈Within the context of one EcuInstance, there can only be one service port that uses the role PowerTakeOff in the RoleBasedPortAssignment.role.⌋

[constr_3085] Service port in the role CallbackDCMRequestServices

Imposition time: CP: IT_EcuExt

⌈Within the context of one EcuInstance, there can only be one service port that uses the role CallbackDCMRequestServices in the RoleBasedPortAssignment.role.⌋

14.3 Extending the ECU Software Composition

As explained in [4], Service Configuration takes place in ECU Configuration phase. In the ECU extract of the System, the Software Components and their ECU-internal connectors are represented as a flat set aggregated by RootSwCompositionPrototype as indicated in Figure 14.7.

ECU Configuration extends this aggregation by adding SwComponentPrototypes (each typed by a specific ServiceSwComponentType) and the required AssemblySwConnectors to the RootSwCompositionPrototype. This is possible without changing the initial artifacts of the ECU extract, because these aggregations are stereotyped as ≪atpSplitable≫ in the meta-model.

After this step, the RootSwCompositionPrototype (denoted by EcucValueCollection.ecuExtract.rootSoftwareComposition) represents the whole

Software Composition on the given ECU. This collection includes both the software components mapped to the ECU **and** the necessary service components represented as one SwComponentPrototype for each AUTOSAR Service utilized on the given ECU.

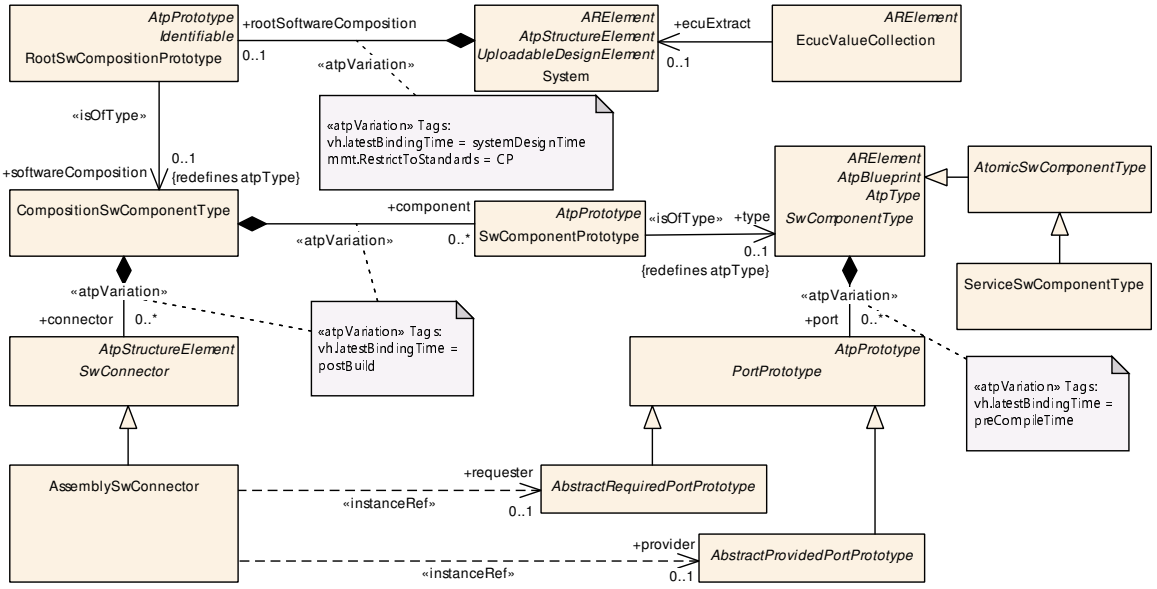

Figure 14.7: Usage of RootSwCompositionPrototype on an ECU

14.4 Communication

In explaining how SystemSignals are handled in the System with category ECU_EXTRACT, Section 14.2.3 touched on the topic of inter-ECU Communication. However, in order to enable the ECU Configuration of the COM-Stack, the relevant information of all layers of the AUTOSAR COM-Stack needs to be present in the System with category ECU_EXTRACT, including the central Communication classes ISignal, Pdu and Frame.

The above mentioned Communication elements have dependencies on each other, for ordinary COM-communication this means:

- Frames are assembled from one or more Pdus.
- ISignalIPdus carry their information in form of ISignals.
- ISignals as interaction points between RTE and COM refer to SystemSignals.

Note that the above list is not complete; TP and NM require additional elements. However, for the sake of clarity the following paragraphs describes the standard use case of a direct Signal-based communication between two EcuInstances. Once the handling of this case is understood, the additional model elements as NPdu, NmPdu, SystemSignalGroup etc. can be handled following the same basic principles.

For the System with category ECU_EXTRACT only the ECU-relevant subset of information present in the system-wide communication is to be considered. In order to establish this set of information, the dependencies in the list above are being followed.

14.4.1 Frame

In a complete System with category SYSTEM_DESCRIPTION, every outside communication of an EcuInstance will either be associated with an outgoing or and incoming Frame. The exact number and types of Frames to be received or sent by an EcuInstance is determined by the Communication Matrix (Chapter 6).

According to the selection rules for the Topology (14.1), the System with category ECU_EXTRACT contains all FrameTriggerings associated with Frames that are of any interest to the EcuInstance: If a particular FrameTriggering refers to a FramePort of type 'out' the associated Frame is to be sent by the EcuInstance, if it refers to an 'in' port the Frame is to be received. Therefore, the following selection rule applies:

- The System with category ECU_EXTRACT shall contain all Frame elements which are referenced by any included FrameTriggering.

14.4.2 PDU

Frames are assembled from one or more Pdus. In order to include all required Pdu elements, the following selection criteria apply:

- The System with category ECU_EXTRACT shall contain all Pdu elements which are referenced by any included Frame's PduToFrameMapping.
- The System with category ECU_EXTRACT shall contain all Pdu elements which are referenced by any included PduTriggering.
- For multiplexed Pdus, additionally all ISignalIPdus referenced by the MultiplexedIPdu's static and dynamic parts need to be included.

The second criterion is e.g. required in a pure post-build configuration scenario, where the frame-layout may not be completed at the time of System with category ECU_EXTRACT creation.

14.4.3 ISignals and ISignalGroups

ISignalIPdus carry their information in form of ISignals or ISignalGroups. In order to include all required ISignal and ISignalGroup elements, the following selection criteria apply:

- The System with category ECU_EXTRACT shall contain ISignal elements which are referenced by included ISignalIPdu's ISignalToIPduMapping. One exception are Pdu Gateways. Signal definitions that are not directly relevant for Gateways in case that the Pdu is routed as a whole (Pdu Routing) shall be omitted. See Section 14.4.5 for more details.
- The System with category ECU_EXTRACT shall contain all ISignal elements which are referenced by any included ISignalTriggering.
- The System with category ECU_EXTRACT shall contain ISignalGroup elements which are referenced by included ISignalIPdu's ISignalToIPduMapping. One exception are Pdu Gateways. Signal Group definitions that are not directly relevant for Gateways in case that the Pdu is routed as a whole (Pdu Routing) shall be omitted. See Section 14.4.5 for more details.
- The System with category ECU_EXTRACT shall contain all ISignalGroup elements which are referenced by any included ISignalTriggering.

Like in the case of the Pdu inclusion rules, the second and fourth criterion is required in scenarios with incomplete Pdu modeling due to post-build configurability of the communication matrix.

14.4.4 SystemSignal and SystemSignalGroup

Whereas the rules specified in Section 14.2.3 for the inclusion of SystemSignal comprise all SystemSignals that are being used by the Software Components in the ECU, the inclusion rules above stated for ISignalIPdus and ISignals may require the inclusion of additional SystemSignals. Also, strictly speaking both SystemSignals and SystemSignalGroup need to be considered. The complete inclusion rules for SystemSignals and SystemSignalGroups are:

- The System with category ECU_EXTRACT shall contain all SystemSignals and SystemSignalGroup elements which are referenced by any included DataMapping.
- The System with category ECU_EXTRACT shall contain all SystemSignal elements which are referenced by any included ISignal.
- The System with category ECU_EXTRACT shall contain all SystemSignalGroup elements which are referenced by any included ISignalGroup.

In addition on the receiving EcuInstance the following cases exist:

- only one SystemSignal out of the transmitted SystemSignalGroup is received: no SystemSignalGroup is required in the Ecu Extract of the receiving EcuInstance.
- more than one but not all SystemSignals out of the transmitted SystemSignalGroup are received: new SystemSignalGroup shall be created in the

System with category ECU_EXTRACT of the receiving EcuInstance containing the received SystemSignals.

- all SystemSignals out of the transmitted SystemSignalGroup are received: the original SystemSignalGroup shall be the taken over to the System with category ECU_EXTRACT of the receiving EcuInstance.

14.4.5 Gateways

Gateways that refer the EcuInstance shall be included in the System with category ECU_EXTRACT. The complete inclusion rules for Gateways are:

- The System with category ECU_EXTRACT shall contain all FrameMapping elements that are aggregated by the Gateway element.
- The System with category ECU_EXTRACT shall contain all IPduMapping elements that are aggregated by the Gateway element.
- The System with category ECU_EXTRACT shall contain all ISignalMapping elements that are aggregated by the Gateway element.
- ISignal definitions that are not directly relevant for the Gateway in case that the Pdu containing these ISignals is routed as a whole (Pdu Routing) shall be omitted .
- ISignalGroup definitions that are not directly relevant for the Gateways in case that the Pdu containing these ISignalGroups is routed as a whole (Pdu Routing) shall be omitted .

14.4.6 TP configuration

The TP-configuration element TpConfig and all its associated elements shall be included into the System with category ECU_EXTRACT if the EcuInstance has an TpAddress configured in this TpConfig.

14.4.7 NM configuration

The Nm configuration part of the System with category ECU_EXTRACT shall include the NmEcu that references the included EcuInstance. In addition a NmCoordinator composed by this NmEcu shall be included. Furthermore any NmNode referenced by the NmCoordinator shall be included. For each included NmNode the composing NmCluster shall be included. For each included NmCluster the composing NmConfig shall be included.

14.5 Naming Issues

[TPS_SYST_05015] Naming conventions ⌈The definition of naming conventions may facilitate the avoidance of name clashes to the further degree. However, these naming conventions can only be defined on the model level and the System Template does not define any specific naming conventions.⌋

Please note that a detailed information about mechanisms to resolve naming conflicts is given in [3]: [TR_METH_03005], [TR_METH_03006], [TR_METH_03007], [TR_METH_03008], [TR_METH_03009], [TR_METH_03010].

14.5.1 Package Structure

As detailed in the sections above, extracting information from the System with category SYSTEM_DESCRIPTION into an System with category ECU_EXTRACT is a non-trivial transformation: While some of the model elements are simply copied verbatim into the System with category ECU_EXTRACT, it is additionally necessary to create new elements reducing parts of system-wide structures, most noticeably in flattening of the hierarchical VFB view to the ECU Flat View.

All such elements being created or modified in the process of generating the System with category ECU_EXTRACT shall reside in the same ARPackage. In order to avoid namespace conflicts with existing elements, the package shall exclusively be used for this purpose.

By creating derivation elements from elements originally contained in the System with category SYSTEM_DESCRIPTION package structure, duplications of names may occur. This kind of name clashes shall be resolved by a suitable naming algorithm (see section 14.5.3).

All Elements that are taken over from the System with category SYSTEM_DESCRIPTION unchanged (e.g. AtomicSwComponentType, PortInterface, ApplicationDataType, EcuInstance, CommunicationCluster) shall remain in their original packages.

ARElements not used in the System with category ECU_EXTRACT shall not be copied to the ECU Extract XML file.

In more detail, ARPackages taken over from System with category SYSTEM_DESCRIPTION will not be altered by the ECU extraction process, except that some ARElements will not be included in the actual XML file of the extract: ARElements which exist in the System with category SYSTEM_DESCRIPTION but have been stripped for the System with category ECU_EXTRACT are not actually deleted from their ARPackage, but merely are skipped in the XML file forming the extract. Note that having such a partial view on an ARPackage doesn't break the original ARPackage definition because the composition of PackageableElement, responsible for

adding ARElements to ARPackage, is stereotyped ≪splitable≫; this means several XML files can contribute to an ARPackage, or in case of the ECU Extract an AUTOSAR description file may contain only a subset of the complete ARPackage.

14.5.2 Naming of Measurement and Calibration Data

The software component descriptions provide several means to declare data prototypes which have to be available for measurement and calibration (MCD) tools on the EcuInstance. Together with the System with category ECU_EXTRACT it is required to provide a list of references to the description of these data for further processing in the scope of the EcuInstance. In addition, the MCD tools need a unique name for each instance of such a data prototype. Since the data descriptions are part of the nested composition structure and are contained in reusable types (components or port interfaces), the system description itself does in general not provide unique names for those.

This means, providing such a list with references and unique names for MCD data is also a task of the ECU extractor tool. This list is part of the artifact ECU Flat Map, which is further explained below.

14.5.3 Naming of Derived Elements

When performing the extract process, name clashes may occur, necessitating a naming scheme for elements derived in ECU generation: By flattening the Software Composition hierarchy all component instances present on the considered EcuInstance are put in one ECU-wide software composition. Name clashes may occur for the following reasons:

1. SwComponentPrototypes taken from different Software Compositions are allowed to have identical short names in the hierarchical structure. As all SwComponentPrototypes will be located in the same ECU Flat View, the original name spaces separation no longer exists.
2. Multiple instances of the same CompositionSwComponentType are mapped to an EcuInstance: In this case, duplicates of all contained SwComponentPrototypes will be placed next to each other in the ECU flat composition.
3. The two mechanisms just mentioned may also lead to name clashes in AutosarDataPrototypes if their names shall be used as MCD data names. In addition, reuse of a PortInterface can also lead to name clashes if it provides data elements to be used by MCD.
4. The setup of PortGroups in the ECU flat view can result in name clashes, because two port groups originating from different component types (i.e. different name spaces) may be aggregated within the flat view.

Therefore the System with category ECU_EXTRACT generator shall take care that all elements derived or created during the extraction process have unique short names. These unique names shall be created in an initial step of the extraction process which leads to the creation of an initial ECU Flat Map. Some ways to satisfy this requirement may be:

- Use globally unique identifiers (GUID) for generating short names.
- Add a number to the original name; if done consistently the flat map approach makes this reproducible.
- Expand the name recursively by the names of the containing elements (e.g. compositions) until is is unique.
- Allow human interaction (this may be combined with an initially proposed name expansion).

The creation of a new short name is compulsory only if otherwise a clash would occur.

[constr_2025] Uniqueness of symbol attributes

Imposition time: CP: IT_EcuExt

⌈With the exception of RunnableEntities that are subject to [constr_1234] (RunnableEntities owned by NvBlockSwComponentTypes), in the context of a single EcuInstance the values of the RunnableEntity.symbol in combination with the attribute symbol of the meta-class SymbolProps owned by AtomicSwComponentType of all deployed RunnableEntities shall be unique such that no two (or more) combinations of RunnableEntity.symbol and the symbol of the meta-class SymbolProps owned by AtomicSwComponentType in the role symbolProps share the same value.⌋

14.5.4 Re-use of short names assigned in previous iterations

As described in the previous section, potential name clashes during ECU extraction shall be avoided by assigning unique names to the elements specifically created for the System with category ECU_EXTRACT and for the list of MCD data per EcuInstance. Considering the use case of iterative development (also see Section 14.6), the same names shall be assigned to existing elements in consecutive iterations. Elements which have been modified or newly introduced between two ECU extract iterations shall not use an existing short name. Additionally, the ECU extractor tool shall not re-use any short name used in any iteration from previous development phases if the meaning of the element is not exactly the same (i.e. the element's back reference into the System Configuration Description is not the same.)

14.6 ECU Extract in subsequent Cycles of Iterative Development

14.6.1 Traceability of model elements created in ECU Extract

For development scenarios in real life projects iterative development shall be supported.

The following use case shall be considered:

Changes in the System with category SYSTEM_DESCRIPTION require the recreation of an System with category ECU_EXTRACT. In the successive re-run of ECU configuration, ECU configuration parameters which were configured based on the previous System with category ECU_EXTRACT need to be maintained for those parts in the System with category ECU_EXTRACT that didn't change between iterations.

Consequently, there are two requirements on the extraction process:

- Elements that are present in both versions of the System with category SYSTEM_DESCRIPTION shall not change their short names between the two ECU Extracts either.
- If changes between the two versions of the System with category SYSTEM_DESCRIPTION lead to the creation of new model elements in the System with category ECU_EXTRACT, then these newly created elements shall have new names that have not been used in previous iterations of the System with category ECU_EXTRACT. (See also Section 14.5.4).

In order to fulfill these requirements, a back-tracing of the relevant model elements in the System with category ECU_EXTRACT to their counterparts in the System with category SYSTEM_DESCRIPTION shall be established. Based on these back references, short names shall consistently be re-used in iterations. Relevant elements are all those which potentially have been modified in the extraction process.

All back-tracing references are collected in one central table per System with category ECU_EXTRACT based on the meta-class FlatMap. This table collects "instance" entries for each Ecu Extract element that is being created in the System with category ECU_EXTRACT transformation and for each MCD data object that has to be available in the EcuInstance. These entries are called FlatInstanceDescriptor.

Each mapping entry owns two references per mapped element, one reference pointing to the target element in the System with category ECU_EXTRACT, the other one pointing to the origin upstream description (for more details about allowed upstream description see FlatInstanceDescriptor.upstreamReference). Both of these references are deep "instance" references, requiring a tuple of context/target description.

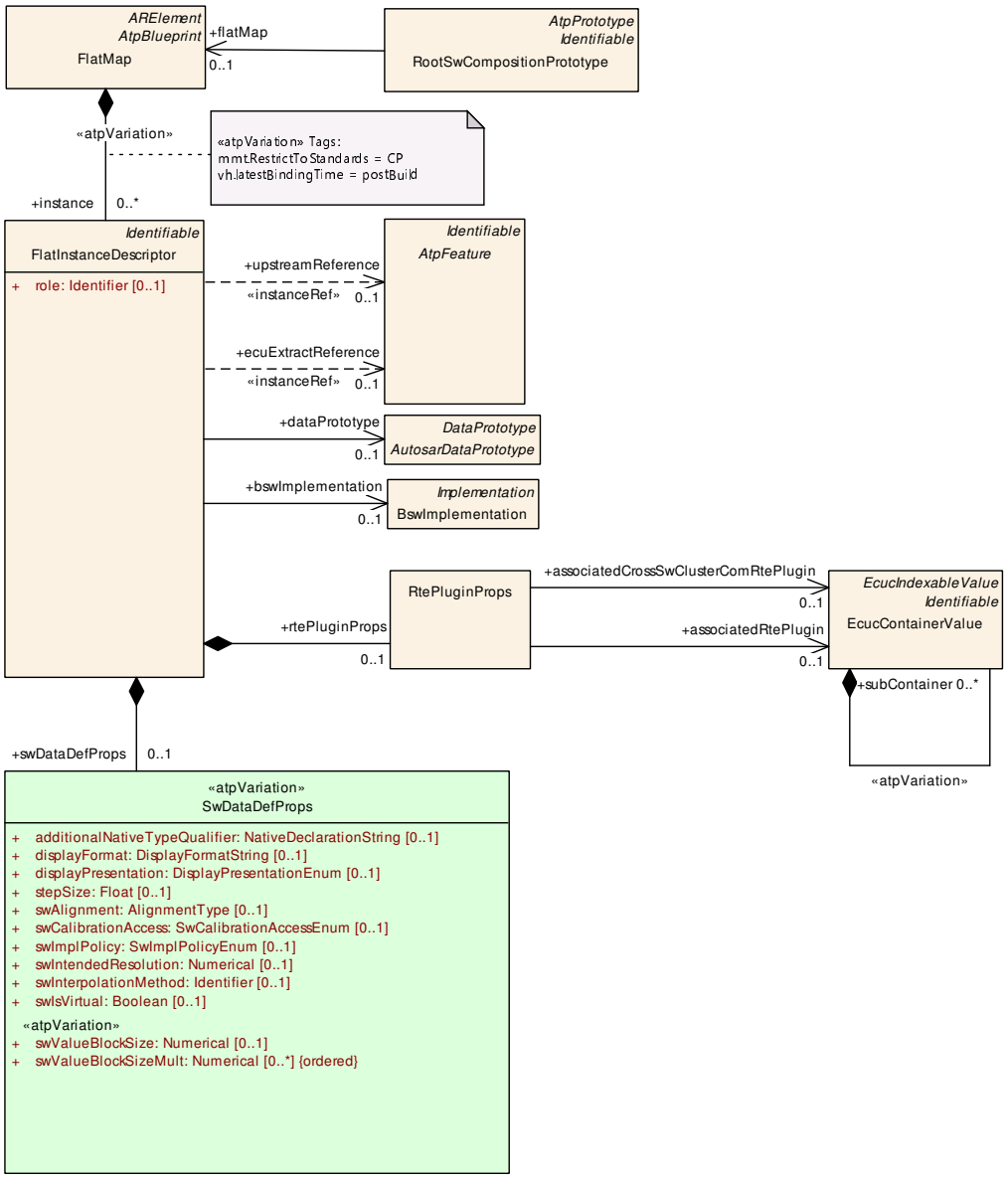

Figure 14.8: Flat Map (CommonStrucure: FlatMap)

Class	**FlatMap**
Note	Contains a flat list of references to software objects. This list is used to identify instances and to resolve name conflicts. The scope is given by the RootSwCompositionPrototype for which it is used, i.e. it can be applied to a system, system extract or ECU-extract. An instance of FlatMap may also be used in a preliminary context, e.g. in the scope of a software component before integration into a system. In this case it is not referred by a RootSwComposition Prototype. **Tags:** atp.recommendedPackage=FlatMaps This Class is only used by the AUTOSAR Classic Platform.
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

△

Class	**FlatMap**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
instance	FlatInstanceDescriptor	*	aggr	A descriptor instance aggregated in the flat map. The variation point accounts for the fact, that the system in scope can be subject to variability, and thus the existence of some instances is variable. The aggregation has been made splitable because the content might be contributed by different stakeholders at different times in the workflow. Plus, the overall size might be so big that eventually it becomes more manageable if it is distributed over several files. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instance, instance.shortName, instance.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 14.1: FlatMap

Class	**FlatInstanceDescriptor**			
Note	Represents exactly one node (e.g. a component instance or data element) of the instance tree of a software system. The purpose of this element is to map the various nested representations of this instance to a flat representation and assign a unique name (shortName) to it. Use cases: • Specify unique names of measurable data to be used by MCD tools • Specify unique names of calibration data to be used by MCD tool • Specify a unique name for an instance of a component prototype in the ECU extract of the system description Note that in addition it is possible to assign alias names via AliasNameAssignment. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlatMap.instance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bsw Implementation	BswImplementation	0..1	ref	Reference to BswImplementation that defines the context for the AutosarDataPrototype that is referenced by Flat InstanceDescriptor.dataPrototype. **Tags:** atp.Splitkey=bswImplementation
dataPrototype	AutosarDataPrototype	0..1	ref	Reference to a DataPrototype that is defined in the Bsw InternalBehavior in the context of a BswImplementation that is defined by the FlatInstanceDescriptor.bsw Implementation reference. **Tags:** atp.Splitkey=dataPrototype

▽

△

Class	**FlatInstanceDescriptor**			
ecuExtract Reference	AtpFeature	0..1	iref	Refers to the instance in the ECU extract. This is valid only, if the FlatMap is used in the context of an ECU extract. The reference shall be such that it uniquely defines the object instance. For example, if a data prototype is declared as a role within an SwcInternalBehavior, it is not enough to state the SwcInternalBehavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying instance of the component prototype and the Atomic SoftwareComponentType, which is refered by the particular SwcInternalBehavior. **Tags:** atp.Splitkey=ecuExtractReference, ecuExtract Reference.contextElement, ecuExtractReference.target xml.sequenceOffset=40 **InstanceRef implemented by:** AnyInstanceRef
role	Identifier	0..1	attr	The role denotes the particular role of the downstream memory location described by this FlatInstanceDescriptor. It applies to use case where one upstream object results in multiple downstream objects, e.g. ModeDeclaration GroupPrototypes which are measurable. In this case the RTE will provide locations for current mode, previous mode and next mode.
rtePluginProps	RtePluginProps	0..1	aggr	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in. **Tags:** atp.Splitkey=rtePluginProps
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this FlatInstanceDescriptor. **Tags:** atp.Splitkey=swDataDefProps
upstream Reference	AtpFeature	0..1	iref	Refers to the instance in the context of an "upstream" description, which could be: the SYSTEM_ DESCRIPTION, or SYSTEM_EXTRACT, or ECU_ SYSTEM_DESCRIPTION, or SW_CLUSTER_SYSTEM_ DESCRIPTION, or the basic software module description (in this case only the target reference of the AnyInstance Ref is needed), or (if a flat map is used in preliminary context) a description of an atomic component or composition. This reference is optional in case the flat map is used in ECU context. The reference shall be such that it uniquely defines the object instance in the given context. For example, if a data prototype is declared as a role within an SwcInternal Behavior, it is not enough to state the Swc Internal Behavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying the instance of the component prototype that contains the particular instance of Swc InternalBehavior. **Tags:** atp.Splitkey=upstreamReference, upstream Reference.contextElement, upstreamReference.target xml.sequenceOffset=20 **InstanceRef implemented by:** AnyInstanceRef

Table 14.2: FlatInstanceDescriptor

[constr_9300] Existence of FlatMap.instance

Imposition time: CP: IT_EcuExt

⌈For each FlatMap, at least one FlatInstanceDescriptor shall be aggregated by FlatMap in the role instance.⌋

[TPS_SYST_01000] FlatInstanceDescriptor roles ⌈If a ModeDeclarationGroupPrototype is measurable the FlatMap shall contain three entries where the particular roles are set to

- CURRENT_MODE specifies the FlatInstanceDescriptor applicable for current mode value of the ModeDeclarationGroupPrototype
- PREVIOUS_MODE specifies the FlatInstanceDescriptor applicable for previous mode value of the ModeDeclarationGroupPrototype
- NEXT_MODE specifies the FlatInstanceDescriptor applicable for next mode value of the ModeDeclarationGroupPrototype

Please note that these entries may exist in a FlatMap even if the ModeDeclarationGroupPrototype is not measurable.⌋

[TPS_SYST_02421] Semantics of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation ⌈In case that an AutosarDataPrototype of a BswInternalBehavior needs to be referenced by the FlatInstanceDescriptor the usage of the FlatInstanceDescriptor.ecuExtractReference is not possible since the BswImplementation cannot be defined as a context reference. In this case the plain references FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation shall be used instead. The BswInternalBehavior in which the AutosarDataPrototype is defined can be followed via the BswImplementation.behavior reference.⌋

[constr_9361] FlatInstanceDescriptor.dataPrototype reference restriction

Imposition time: CP: IT_EcuExt

⌈The FlatInstanceDescriptor.dataPrototype reference shall not be used to reference an ArgumentDataPrototype.⌋

[constr_9362] Mutually exclusive existence of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation vs. FlatInstanceDescriptor.ecuExtractReference

Imposition time: CP: IT_EcuExt

⌈If a FlatInstanceDescriptor references an AutosarDataPrototype with role dataPrototype and a BswImplementation in the role bswImplementation then the same FlatInstanceDescriptor shall not reference an AtpFeature in the role ecuExtractReference.⌋

[constr_9363] **`FlatInstanceDescriptor.dataPrototype` and `FlatInstanceDescriptor.bswImplementation`**

Imposition time: CP: `IT_EcuExt`

⌈If a `FlatInstanceDescriptor` references an `AutosarDataPrototype` with role `dataPrototype` then the `FlatInstanceDescriptor` shall also reference a `BswImplementation` in the role `FlatInstanceDescriptor.bswImplementation`.⌋

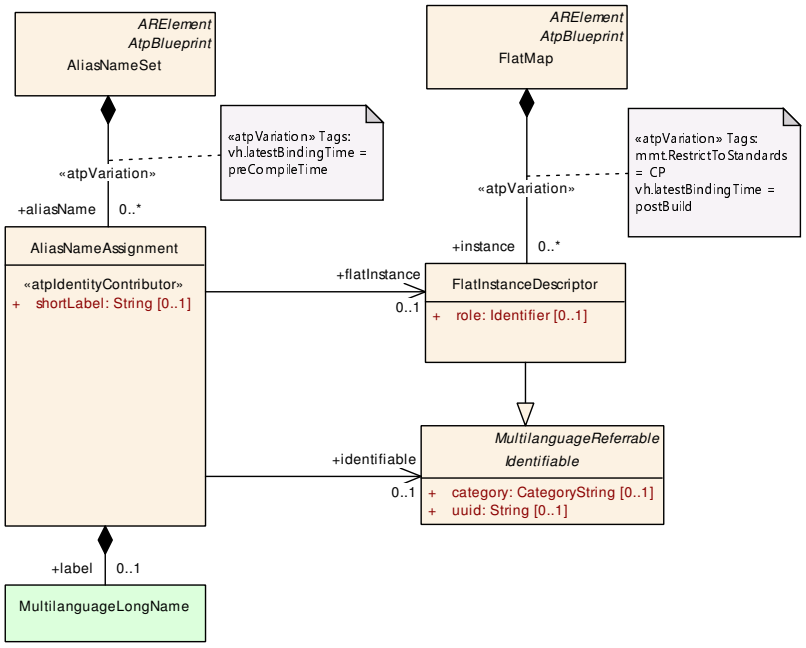

Figure 14.9: Alias Name Assignment (CommonStrucure: AliasNameAssignment)

Class	**AliasNameSet**			
Note	This meta-class represents a set of AliasNames. The AliasNameSet can for example be an input to the A2L-Generator. **Tags:** atp.recommendedPackage=AliasNameSets			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aliasName	AliasNameAssignment	*	aggr	AliasNames contained in the AliasNameSet. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=aliasName, aliasName.shortLabel, aliasName.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 14.3: AliasNameSet

Class	**AliasNameAssignment**			
Note	This meta-class represents the ability to associate an alternative name to a flat representations or an Identifiable. The usage of this name is defined outside of AUTOSAR. For example this name can be used by MCD tools or as a name for component instances in the ECU extract. Note that flatInstance and identifiable are mutually exclusive.			
Base	*ARObject*			
Aggregated by	AliasNameSet.aliasName			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
flatInstance	FlatInstanceDescriptor	0..1	ref	Assignment of a unique name to a flat representation. **Tags:** atp.Splitkey=flatInstance xml.sequenceOffset=60 This Attribute is only used by the AUTOSAR Classic Platform.
identifiable	Identifiable	0..1	ref	Assignment of a unique name to an Identifiable. **Tags:** atp.Splitkey=identifiable xml.sequenceOffset=50
label	MultilanguageLongName	0..1	aggr	This represents an "Alias LongName". **Tags:** atp.Splitkey=label xml.sequenceOffset=20
shortLabel	String	0..1	attr	This attribute represents the alias name. It is modeled as string because the alias name is used outside of AUTOSAR and therefore no naming conventions can be applied within AUTOSAR. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10

Table 14.4: AliasNameAssignment

[constr_9301] Existence of `AliasNameAssignment.shortLabel`

Imposition time: CP: `IT_EcuExt`

⌈For each `AliasNameAssignment`, the attribute `shortLabel` shall exist.⌋

During the ECU extraction process, the ECU `FlatMap` will be processed in the following steps:

1. Create the entries `shortName` and `upstreamReference` of the `FlatMap` or, if a previous version exists, try to reuse them. Resolve name conflicts.
2. Generate the ECU Software Composition.
3. Create the entries `ecuExtractReference` of the ECU `FlatMap`.

More details are define be the AUTOSAR methodology, see [3]. The methodology also allows to have a `FlatMap` for the whole system. This System `FlatMap` can be created and maintained independently from the ECU extraction process, but can be used as an input for the creation of the ECU `FlatMap`.

[constr_3378] Maximal one `AliasNameAssignment` allowed per `FlatInstanceDescriptor`

Imposition time: CP: `IT_EcuExt`

⌈In a given instance of `AliasNameSet` in the bound system there shall be at most one `aliasName` per `FlatInstanceDescriptor`.⌋

14.6.2 Mapping of AUTOSAR attributes to ASAM ASAP2

With the MC Support information AUTOSAR builds a bridge to tools processing ASAM ASAP2 files. In order to support the interoperability of converter tools the following mapping of AUTOSAR attributes to ASAM ASAP2 [24] (also known as "A2l" respectively "ASAM MCD 2MC") is recommended:

- If the `FlatInstanceDescriptor` references `DataPrototypes`:

  `FlatInstanceDescriptor.shortName` ->
  MEASUREMENT Name
  CHARACTERISTIC Name

  `FlatInstanceDescriptor`.(longName + desc |upstreamReference.desc) ->
  MEASUREMENT LongIdentifier
  CHARACTERISTIC LongIdentifier

  `AliasNameAssignment.shortLabel` ->
  MEASUREMENT [-> DISPLAY_IDENTIFIER]
  CHARACTERISTIC [-> DISPLAY_IDENTIFIER]

  `AliasNameAssignment.label`(if provided) +
  `FlatInstanceDescriptor`.(desc |upstreamReference.desc) ->
  MEASUREMENT LongIdentifier
  CHARACTERISTIC LongIdentifier

- If `AliasNameAssignment` references a `SwSystemconstant`:

  `AliasNameAssignment.shortLabel` ->
  SYSTEM_CONSTANT -> Name for SwSystemconstants

- If `AliasNameAssignment` references a `Unit`:

  `AliasNameAssignment.shortLabel` ->
  UNIT -> Name for Units

14.6.3 Assigning communication graphs to RTE Implementation Plug-Ins

When `RTE Implementation Plug-Ins` are used to modularize the RTE implementation, it's required to decide which communication graphs are implemented by the RTE

or by an specific RTE Implementation Plug-In. Thereby an RTE Implementation Plug-In is a part of the overall RTE implementation which is not provided by the RTE Generator but from an additional source (e.g. a Plug-In Generator or a manually implemented source code).

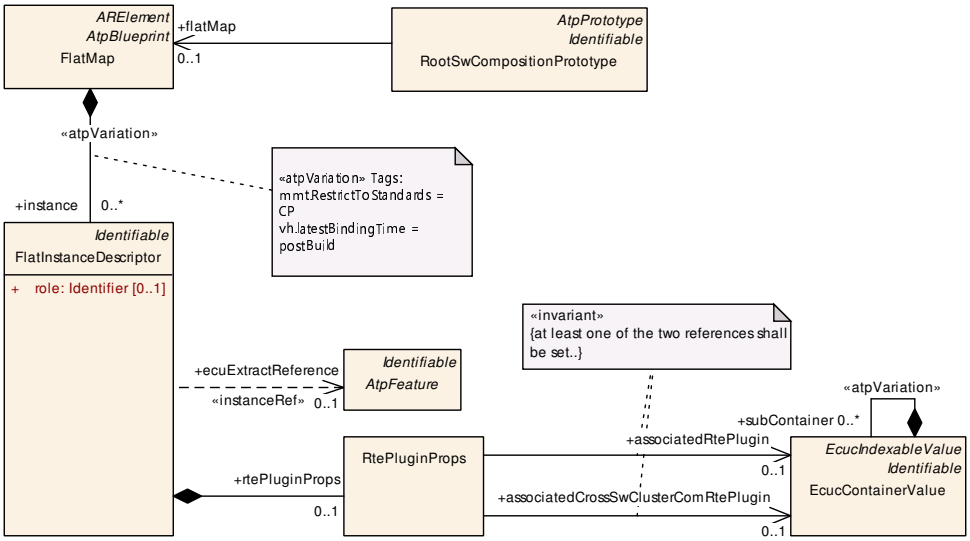

Figure 14.10: ECU Flat Map and rtePluginProps

Class	**RtePluginProps**			
Note	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	FlatInstanceDescriptor.rtePluginProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
associated CrossSwCluster ComRtePlugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling cross Software Cluster communication. **Tags:** atp.Splitkey=associatedCrossSwClusterComRte Plugin
associatedRte Plugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling local Software Cluster communication or communication in a non-cluster ECU. **Tags:** atp.Splitkey=associatedRtePlugin

Table 14.5: RtePluginProps

This assignment is described with the FlatInstanceDescriptor.rtePluginProps, where the RtePluginProps.associatedRtePlugin or RtePluginProps.associatedCrossSwClusterComRtePlugin references the EcucContainerValue representing the identity of an RTE Implementation Plug-In.

Assigning an communication graphs has following underlying semantic:

[TPS_SYST_02197] Assigning communication graphs to RTE Implementation Plug-Ins ⌈The FlatInstanceDescriptor.ecuExtractReference points to an

instance of a VariableDataPrototype and the FlatInstanceDescriptor.rtePluginProps.associatedRtePlugin or RtePluginProps.associatedCrossSwClusterComRtePlugin references the EcucContainerValue which defines the identity of the RTE Implementation Plug-In. This assigns the full communication graph to the specific RTE Implementation Plug-Ins represented by according EcucContainerValue.⌋

For instance the FlatInstanceDescriptor.ecuExtractReference points to instance of a VariableDataPrototype defined by the AnyInstanceRef using

- contextElement: RootSwCompositionPrototype
- contextElement: SwComponentPrototype
- contextElement: PPortPrototype
- target: VariableDataPrototype

According the AUTOSAR Meta-Model various further model elements are exist to describe the complete communication graph, for instance with the means of AssemblySwConnectors, SwComponentType.ports, RunnableEntitys and VariableAccesses. Nevertheless all such related elements of this communication graph are addressed by this single FlatInstanceDescriptor and all access of ExecutableEntitys to this communication graph are handled by the associated RTE Implementation Plug-In.

[constr_3458] FlatInstanceDescriptor.rtePluginProps shall only reference a EcucContainerValue representing a RteRipsPlugin

Imposition time: CP: IT_EcuExt

⌈FlatInstanceDescriptor.rtePluginProps shall only reference an EcucContainerValue which defines the identity of the RTE Implementation Plug-In. This requires that the according EcucContainerValue's definition references a EcucContainerDef having a destinationUri set to /AUTOSAR/EcucDestinationUriDefSets/RteRipsUriDefSet/RteRipsPlugin⌋

[constr_5175] RtePluginProps shall reference at least one EcucContainerValue representing a RteRipsPlugin

Imposition time: CP: IT_EcuExt

⌈If a FlatInstanceDescriptor owns are RtePluginProps this RtePluginProps shall define the associatedRtePlugin reference and/or the associatedCrossSwClusterComRtePlugin reference.⌋

To support different work-flows the FlatInstanceDescriptor.rtePluginProps is defined as ≪atpSplitable≫. Therefore it's possible the do the assignment of communication graphs immediately during the creation of the ECU Flat Map or in a second processing step after the ECU Flat Map is already created.

Further information, specifications, and applicable constraints on assignments of communication graphs are provided in the document SWS RTE [23] at which specific anchor points of an communication graphs the assignment shall be described.

Some further notes about the chosen modeling pattern

In general it is an unusual pattern, that a meta class not being related to ECU configuration references an `EcucContainerValue`. But the `FlatInstanceDescriptor` of the ECU Extract is in any case closely related to the configuration of the ECU. Furthermore in case of data conversion it's mandatory to provide for each different representation a `FlatInstanceDescriptor` for the communication graph. Further information about such configurations is provided in the document SWS RTE [23].

The alternative approach to describe the `RTE Implementation Plug-In` as meta class is not the right approach since only very few properties of `RTE Implementation Plug-In`s are standardized. There is also no need to exchange information between different development parties about those properties. Due to this reason `RTE Implementation Plug-In`s are described by the means of ECU Configuration elements.

The alternative approach to model the relationship between `FlatInstanceDescriptor`s and the container which represents the `RTE Implementation Plug-In` with a mapping pattern was rejected due to the very high number of expected configuration elements.

14.7 Variant Handling in ECU Extract

The System Template supports the creation of variants in many of its model elements. Depending on the binding time, some of this variability may have been already resolved within the `System` with `category` `SYSTEM_DESCRIPTION` at the time of creating the `System` with `category` `ECU_EXTRACT`, and a cleanup step may have removed some of the complexity by removing the out-configured variability.

If however binding of a concrete variation condition happens in a later stage of the AUTOSAR methodology (e.g. during ECU Configuration or even post build), or if for other process reasons such a cleanup step is not applicable, the variability needs to be carried over to the `System` with `category` `ECU_EXTRACT`.

14.7.1 System Constants

In the AUTOSAR variant handling concept, `SwSystemconst` represents a variant selector which needs to have its value assigned latest at binding time of any expression which refers to it. Such a value assignment may be done literally using a fixed value, or by specifying a formula, depending on the values of other variant selectors. The elements to do this are collected in a `SwSystemconstantValueSet`, aggregating individual value assignment expressions in the form of `SwSystemconstValue`.

In the System with category ECU_EXTRACT, all SwSystemconst elements are included that influence its variable content. In detail the following rules for the inclusion of SwSystemconst apply:

- System with category ECU_EXTRACT shall contain all SwSystemconst elements that are being referenced directly by variable elements contained in the System with category ECU_EXTRACT.
- Additionally, whenever a SwSystemconst's value is assigned indirectly using an SwSystemconstValue's ConditionByFormula expression, each SwSystemconstValue referred to in the assignment formula needs to be included, too. As such assignments may be nested in multiple levels, the whole directed acyclic graph of SwSystemconst elements influencing the System with category ECU_EXTRACT variability need to be included.

Additionally to the SwSystemconst elements also all relevant SwSystemconstValue assignments need to be included. As they are aggregated by SwSystemconstantValueSet, the whole Value Set is included whenever one of its SwSystemconstValue assignments is relevant for the System with category ECU_EXTRACT.

Note: Typically, the assignment of Variants ("Binding") will be done in a Variant Configuration work product, separate from the actual System with category ECU_EXTRACT. In this case, the relevant information from the Variant Configuration also needs to be extracted and delivered in combination with the System with category ECU_EXTRACT. From the model point of view it doesn't matter whether System with category ECU_EXTRACT and Variant Configuration are contained in the same file or in separate files.

14.7.2 Nested Whole/Part class variants

In case of flattening the hierarchical VFB view to the ECU flat view representation, the case may appear that one conditional SwComponentPrototype is nested within another SwComponentPrototype depending on another variance condition. As the resulting ECU flat view only has a flat representation of SwComponentPrototypes, such a double condition needs to be resolved to a single condition in the resulting SwComponentPrototypes.

In this case, the variation condition formula needs to be altered such that the two (or more) individual conditions are combined in a boolean AND function.

14.7.3 Multiple instances of calibration parameters in system scope

Use case: In complex systems the problem occurs that parameter values may depend on the configuration of the vehicle due to functional side effects. E.g. the calibration

of a lambda sensor depends from the kind of transmission due to mechanical impacts (e.g due to additional / different curvatures in the exhaust pipe)

The difficulty is that those dependencies are typically detected after design of the software components and shall not change the software component design. Furthermore this is typical use case for post build variability since the ECU SW should not change due to environmental variability.

[TPS_SYST_02029] Multiple `ParameterDataPrototype` instances in an EcuExtract ⌈It shall be possible to instruct the RTE Generator to provide various instances for a `ParameterDataPrototype` in the `System` with `category` `ECU_EXTRACT`. Therefore one `FlatInstanceDescriptor` per expected data instance has to point to the `ParameterDataPrototype` as an atpTarget.⌋

[constr_3114] `FlatInstanceDescriptor`s pointing to the same `ParameterDataPrototype` shall have different `postBuildVariantCondition`s

Imposition time: CP: `IT_EcuExt`

⌈`FlatInstanceDescriptor`s that are pointing as an atpTarget to the same `ParameterDataPrototype` instance shall have different `postBuildVariantCondition`s.⌋

Note: When several instances of a `ParameterDataPrototype` are created it shall be ensured that at most one parameter instance is active in a post build variant.

[constr_3115] `FlatInstanceDescriptor`s pointing to the same `ParameterDataPrototype` instance

Imposition time: CP: `IT_EcuExt`

⌈When several `FlatInstanceDescriptor`s point to the same `ParameterDataPrototype` instance as an atpTarget in the context of a `ParameterInterface` the different `FlatInstanceDescriptor`s shall point to the `PPortPrototype` of the owning `ParameterSwComponentType`. In this case the `PPortPrototype` typed by the `ParameterInterface` is part of the context of the according `AnyInstanceRef`.⌋

Please note that the individual `FlatInstanceDescriptor`s are utilized to provide unique names for the MCD tool as well as individual `CalibrationParameterValue`s typically refer to the `FlatInstanceDescriptor`s to provide instance specific initialization values.

15 Supported special use-cases

The description means of the communication matrix in the System Template potentially support a variety of use-cases. Some combinations of description means are explicitly ruled-out by semantical constraints. But the remaining space for the possible descriptions is so huge, that certain use-cases are actually not supported by tool-vendors because they did not consider them. This chapter describes special use-cases that can be specified in the System Template in order to get a harmonized support by tools.

15.1 Support of sending / receiving same Can/Flexray Frame on same channel (Pdu Gateway Use-Case)

Description: The System Template supports the definition of a communication where the same Can/Flexray `FrameTriggering` is sent and received on the same `PhysicalChannel` of one Pdu Gateway `EcuInstance`.

Rationale: This use-case occurs in gateway `EcuInstance`s which are used in several vehicle platforms.

Implementation: This usage shall be supported by defining one `Frame` and one `FrameTriggering` with different directions on the referenced `FramePort`s for the same `PhysicalChannel`. Also one `Pdu` and one `PduTriggering` with different directions on the referenced `IPduPort`s for the same `PhysicalChannel` shall be used.

Example: In figure 15.1 a sample network setup is shown. The ECU1 is designed to send the Frame_X on the `PhysicalChannel`. The ECU2, ECU3 and ECU4 do receive the information. But since ECU1 is optional, ECU4 is also designed to send the Frame_X on the network (in case ECU1 is not present). Please note that in in this example ECU4 is a gateway `EcuInstance` that is connected to an additional channel.

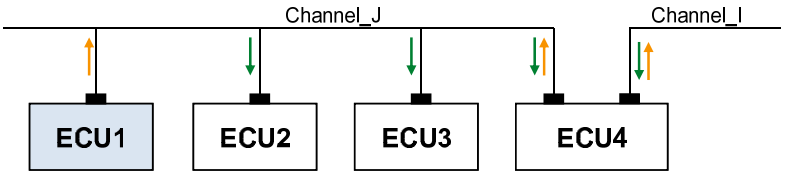

Figure 15.1: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the Frame_X and one `FrameTriggering` for the `PhysicalChannel` (figure 15.2). Each `EcuInstance` sending or receiving the `FrameTriggering` does define one `FramePort` per direction, thus for ECU4 there are two `FramePort`s defined.

For each Pdu mapped to the Frame there exists one definition for the Pdu_X and one PduTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the Pdu does define one IPduPort per direction, thus for ECU4 there are two IPduPorts defined.

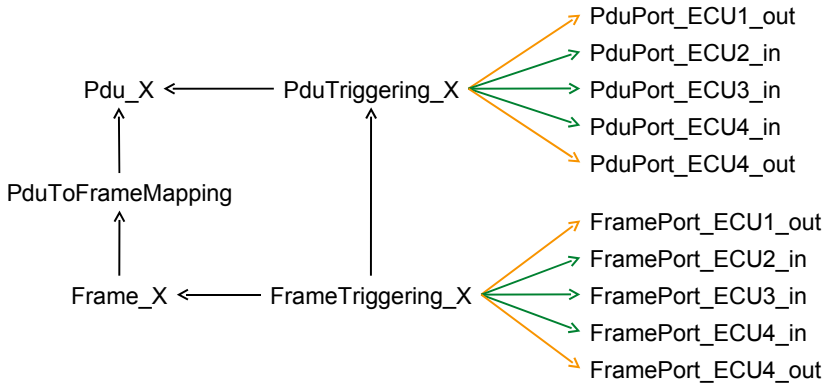

Figure 15.2: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent by the same Gateway ECU

In case a System Extract / ECU Extract is build, only the relevant FramePorts and IPduPorts for the corresponding EcuInstance are extracted. Especially in case an additional EcuInstance is designed to send and receive the same Frame all the other ECU extracts will not be affected by this change.

15.2 Support of sending / receiving same Can/Flexray Frame on same channel (bidirectional routing in COM)

Description: The System Template supports the definition of a communication where the same Can/Flexray FrameTriggering is sent and received on the same PhysicalChannel of one EcuInstance and the content of this Frame is processed by an Application. Please note that this use case is only applicable for legacy communication over COM (excluding the Large Data Path on COM).

Rationale: This use-case occurs in case of runtime variation where the same data is transmitted or received by the same ECU.

Implementation in a System Description: This use-case is supported with the following modelling:

- One Frame and one FrameTriggering with different directions on the referenced FramePorts for the same PhysicalChannel shall be defined.
- One Pdu and one PduTriggering with different directions on the referenced IPduPorts for the same PhysicalChannel shall be defined.
- One ISignal and one ISignalTriggering with different directions on the referenced ISignalPorts for the same PhysicalChannel shall be defined.

Please note that in case of a bidirectional routing on the ISignal level the COM Configuration (ComIPdus) needs to be derived from the PduTriggering and from IPduPorts.

Example: In figure 15.3 a sample network setup is shown. The same data (Frame_X) is transmitted by Ecu4 and by Ecu1 (runtime variation). Ecu4 is designed to send and to receive the Frame_X on the network. For Ecu2 and Ecu3 it is transparent from which sender (Ecu1 or Ecu4) the data is transmitted.

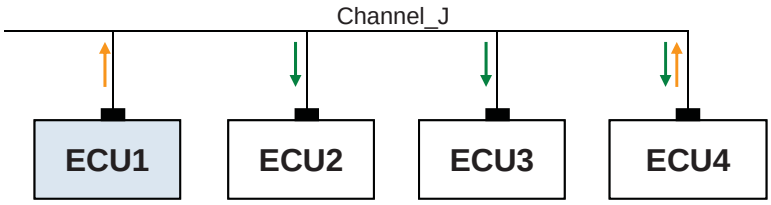

Figure 15.3: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the Frame_X and one FrameTriggering for the PhysicalChannel (figure 15.4). Each EcuInstance sending or receiving the FrameTriggering does define one FramePort per direction, thus for ECU4 there are two FramePorts defined.

For each Pdu mapped to the Frame there exists one definition for the Pdu_X and one PduTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the Pdu does define one IPduPort per direction, thus for ECU4 there are two IPduPorts defined.

For each ISignal mapped to the Pdu there exists one definition for the Signal_X and one ISignalTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the ISignal does define one ISignalPort per direction, thus for ECU4 there are two ISignalPorts defined.

Example 15.4 shows a System Description where only the DataMapping for the PPorts is defined. Please note that in the COM configuration a ComIPdu has a ComIPduDirection. Therefore two ComIPdus (Tx and Rx) need to be created from such a System Description.

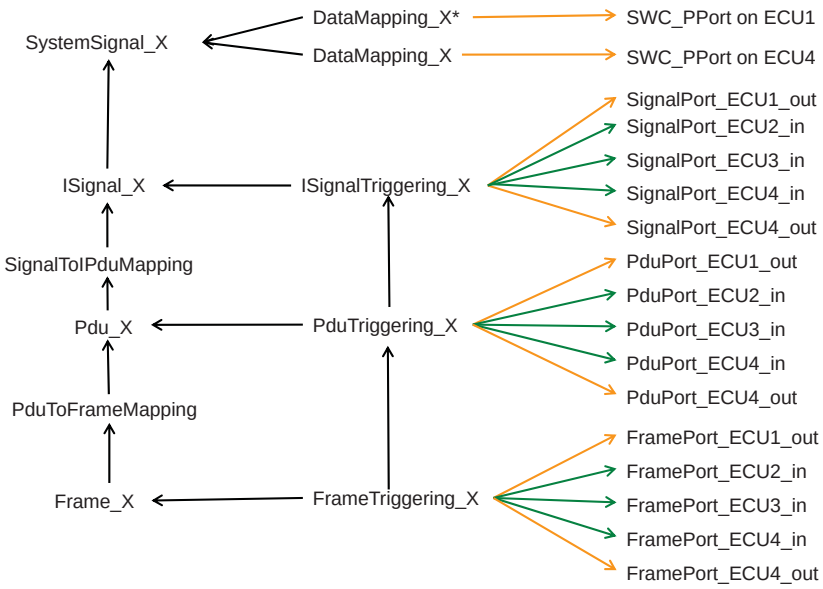

Figure 15.4: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Description with ECU1, ECU2, ECU3 and ECU4)

In case a System Extract / ECU Extract is build, only the relevant `FramePort`s, `IPduPort`s and `ISignalPort`s for the corresponding `EcuInstance` are extracted. Especially in case an additional `EcuInstance` is designed to send and receive the same `Frame` all the other ECU extracts will not be affected by this change. Figure 15.5 shows a System Extract where only the description for ECU4 is available. Please note that in this example the `VariableDataPrototype` in the PPort and the `VariableDataPrototype` in the RPort of the Software Component are mapped to the same `SystemSignal`.

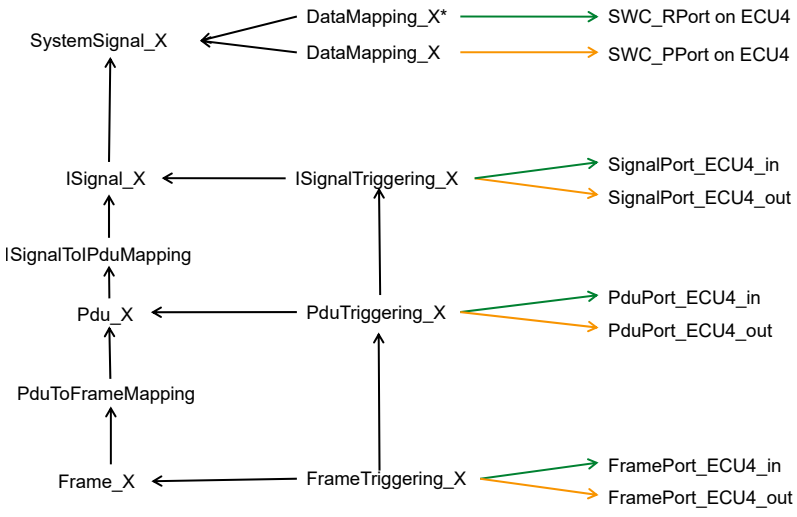

Figure 15.5: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Extract with ECU4 only)

15.3 Support of dynamic CAN IDs

To support efficient diagnostics with on-board clients, efficient routing, and efficient SAE J1939 transport protocol and request handling, AUTOSAR provides access to dynamic CAN identifier parts in upper layers of the COM stack. This is achieved by appending parts of the identifier (or the complete identifier) as MetaData to the Pdu payload. The usage of MetaData is an Ecu Configuration decision. A System Description does not define whether MetaData shall be used or not.

The System Template uses the following attributes for the configuration of dynamic CAN IDs:

- The rxMask of a CanFrameTriggering defines the relevant bits in a CAN identifier and thus defines a range of CAN identifiers that match these bits and may vary in the other bits.
- The txMask of a CanFrameTriggering defines the static bits in a CAN identifier and thus allows to set the other bits using the data appended to the payload.

These parameters are sufficient to support the following scenarios:

- A Pdu is transmitted from one AUTOSAR node to another with variable ID parts. In this case, rxMask and txMask will be identical, and the variable identifier parts placed in the Pdu MetaData by the sender will be routed transparently and received in the same way.
- A Pdu is transmitted by one node with a static identifier and received using the rxMask. In this case, the MetaData is not used, and the receiver is tolerant regarding dynamic address parts.
- J1939 Pdu is sent with fixed priority, but priority is ignored by the receiver. Here, the MetaData may or may not be used, and the rxMask differs from the txMask just in the three priority bits.

15.4 N:1 Sender Receiver communication description in a System Extract over one PhysicalChannel

Description: The System Template supports a System Extract description of a n:1 sender-receiver communication over one PhysicalChannel where each sender and the receiver are located on different Ecus. Each sender Ecu sends the same data marked with a different frame identifier (e.g. CAN Identifier) to the receiver Ecu over the PhysicalChannel.

Implementation: This usage shall be supported by defining one Frame and several FrameTriggerings on the same PhysicalChannel. Each defined FrameTriggering refers to the same Frame. The senders and receivers of a specific FrameTriggering are defined with references to FramePorts.

For every defined Pdu that is contained in the Frame exactly one PduTriggering is defined. This also means that all defined FrameTriggerings refer to the same PduTriggerings with the FrameTriggering.pduTriggering reference.

The communication direction of the Pdu is defined by PduTriggering references to IPduPorts. All sender IPduPorts and receiver IPduPorts are referenced by the same PduTriggering.

The description of ISignals and ISignalTriggerings shall be defined accordingly. Please also note that in case of n:1 sender-receiver communication each sender shall be represented by the same SystemSignal according to [constr_3086].

Example: In figure 15.6 a small example is shown. Three different Ecus (Ecu1, Ecu2, Ecu3) are sending the same Frame to Ecu4.

- Ecu1 sends the Frame with CanId = 3 as described with FrameTriggering3.
- Ecu2 sends the Frame with CanId = 2 as described with FrameTriggering2.
- Ecu3 sends the Frame with CanId = 1 as described with FrameTriggering1.

The Frame contains one single Pdu. Only one PduTriggering is defined here that refers to three IPduPorts with communicationDirection "out" (Ecu1OutPort, Ecu2OutPort and Ecu3OutPort) and to one IPduPort with communicationDirection "in" (Ecu4InPort). Please note that the references between the Triggering elements (FrameTriggering.pduTriggering and PduTriggering.iSignalTriggering) are not visible in figure 15.6 for the sake of clarity.

The description of the ISignal that is included in the Pdu and the ISignalTriggering is defined accordingly.

Upstream Mapping: In the basic Ecu configuration for the receiving Ecu that is derived from such a System Extract all FrameTriggerings shall be mapped to the same Pdu that is passed to a upper layer module (e.g. Nm, PduR). This corresponds to the upstream mapping rules for COM Signals defined in [TPS_SYST_01066] and [TPS_SYST_01067].

- CanIf: several CanIfRxPduCfg containers need to be created with different CanIfRxPduCanIds that all point to the same Pdu (CanIfRxPduRef).
- FrIf: several FrIfFrameTriggering containers need to be created that all point to the same Pdu (FrIfFrameStructure/FrIfPdusInFrame/FrIfPduRef)

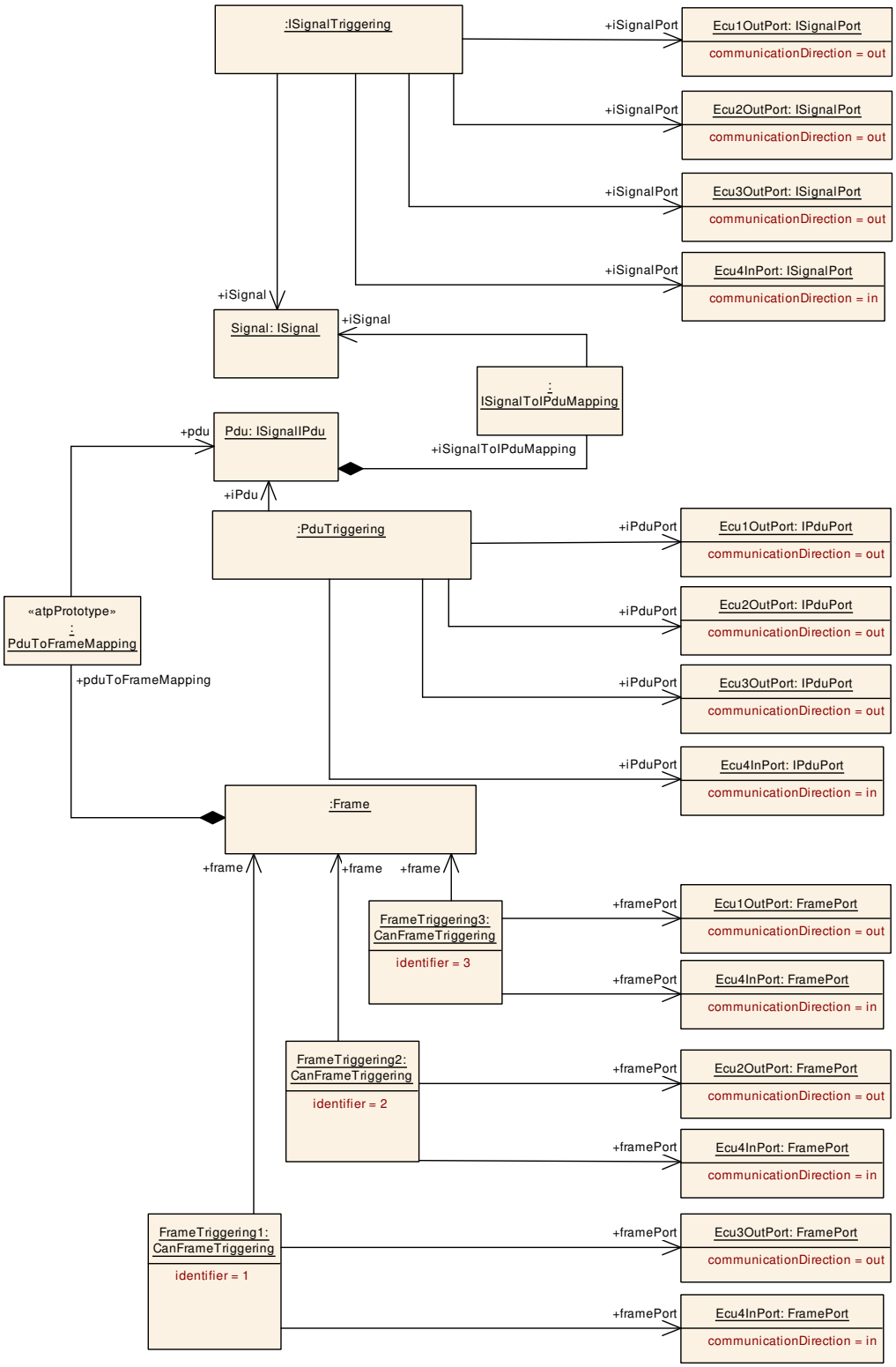

Figure 15.6: Example for a N:1 Sender Receiver communication description in a System Extract

A Reference Material

This chapter contains some relevant reference material for this specification.

A.1 Abbreviations

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This section describes abbreviations that are specific to the System Template Specification and that are not part of the official AUTOSAR Glossary [54].

Following abbreviations are mentioned that are specifically used in this specification:

CAS	Collision Avoidance Symbol
CBV	Control Bit Vector
CMAC	Cipher-based message authentication code
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
DoIp	Diagnostics over IP
FIBEX	Field Bus Exchange Format
I^2C	Inter-Integrated Circuit
ISG	Inter-slot Gap
KaY	MACsec Key Agreement Entity
MAC Address	media access control address
MACsec	Media Access Control security
NAD	Node Address for Diagnostic
NID	NOde Identification
NIT	Network Idle Time
PAE	MACsec Port Access Entity
PLCA	Physical Layer Collision Avoidance
POC	Protocol Operation Control
SAK	MACsec Secure Association key
SecY	MACsec Security Entity
SID	Service Identifier
VDP	Vehicle Data Protocol

Table A.1: Abbreviations used in the scope of this Document

A.2 Imposition Times of Constraints

The constraints formulated in this document have different *actual* imposition times which denote the steps in the workflow when the respective constraint has to be imposed.

Some imposition times “include” other imposition times, an example for this relation is discussed in the Table A.2

The imposition times that are considered applicable in the scope of this document[1] are listed in Table A.2.

Please note that the imposition times are intentionally rendered as technical terms such that it is possible to link back from each constraint to the definition of the affected imposition time in Table A.2.

This document has been created to apply primarily for the *AUTOSAR classic platform* and therefore the discussed imposition times also apply exclusively to the *AUTOSAR classic platform*.

Some constraints, however, *may* also be meaningful in the context of other imposition times, applicable for other *AUTOSAR platforms*.

Imposition Time	Description	Motivation
IT_BinObjMetaData	The definition of binary object meta-data is finished	This imposition time denotes the step in the workflow where the description of `CpSoftwareClusterBinaryManifestDescriptor` is considered complete so that that the Software Cluster Binary Manifest can be defined during the integration.
IT_EcuExt	ECU_EXTRACT is completed	This imposition time denotes the step in the workflow where the ECU_EXTRACT model is considered complete such that it can be used as input for the generation of the RTE.
IT_Mani	Creation of the manifest is finished	This imposition time denotes the step in the workflow, where the manifest is considered complete such that the installation on a target platform can be started.
IT_ResPool	Definition of the resource pool is finished	This imposition time denotes the step in the workflow where the pool of resources which can be provided or required by Software Clusters is considered complete such that such that the development and integration of the Software Cluster can start.
IT_SwCluSysDesc	SW_CLUSTER_SYSTEM_ DESCRIPTION is completed	This imposition time denotes the step in the workflow where the SW_CLUSTER_SYSTEM_DESCRIPTION model is considered complete such that the development and integration of the Software Cluster can start.
IT_SysDes	System design is completed	This imposition time denotes the step in the workflow, where the system design is about to be finished.
IT_SysDesc	SYSTEM_DESCRIPTION is completed	This imposition time is aimed at the time when a system description (e.g. SYSTEM_DESCRIPTION or SYSTEM_ EXTRACT or ECU_SYSTEM_DESCRIPTION) is complete for exchange between parties and is ready as input for Ecu Extract and Com Stack generators.

Table A.2: Imposition Times of constraints in this document

From the appearance of an imposition time that only applies to the *AUTOSAR classic platform* in the text of a constraint in this document, it **shall not be concluded that the constraint is exclusively applicable to the *AUTOSAR classic platform***.

[1] Different imposition times may be defined in the context of other AUTOSAR standard documents

B Detailed Representation of InstanceRef Associations in the System Template

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [1]. This chapter contains the detailed InstanceRef Diagrams.

B.1 Usage of InstanceRefs in Data Mapping diagrams

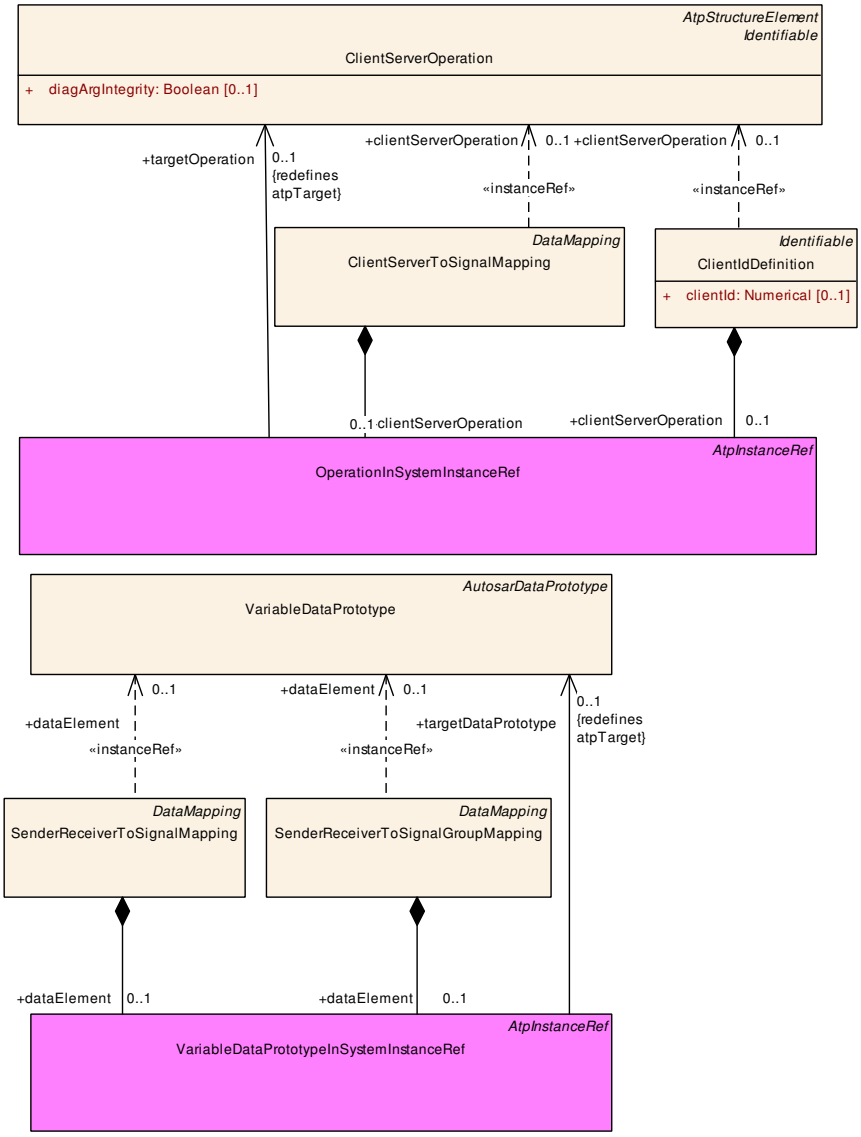

Figure B.1: Data Mapping Instance Ref Usage

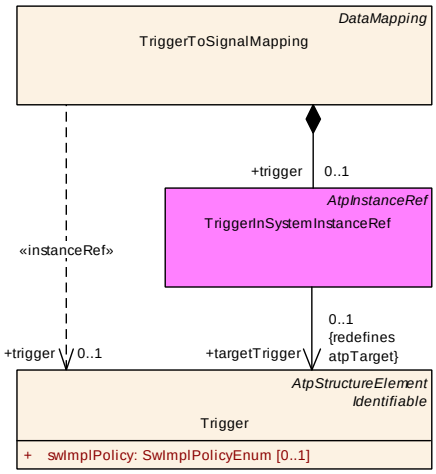

Figure B.2: Modeling of InstanceRef usage for `TriggerInSystemInstanceRef`

B.2 Usage of InstanceRefs in SW Mapping diagrams

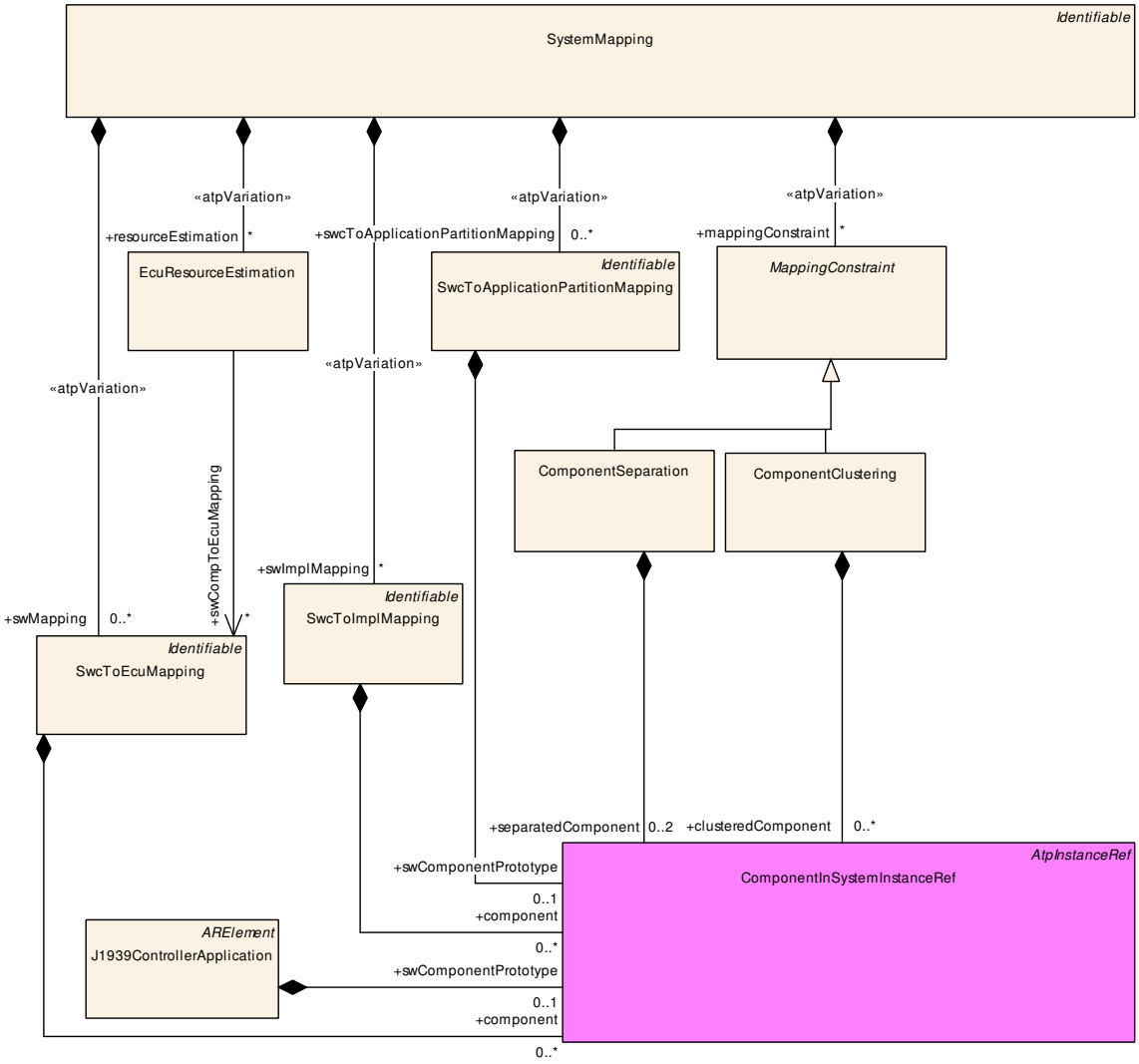

Figure B.3: SW Mapping Instance Ref Usage

B.3 Usage of InstanceRefs in Signal Path Constraint diagrams

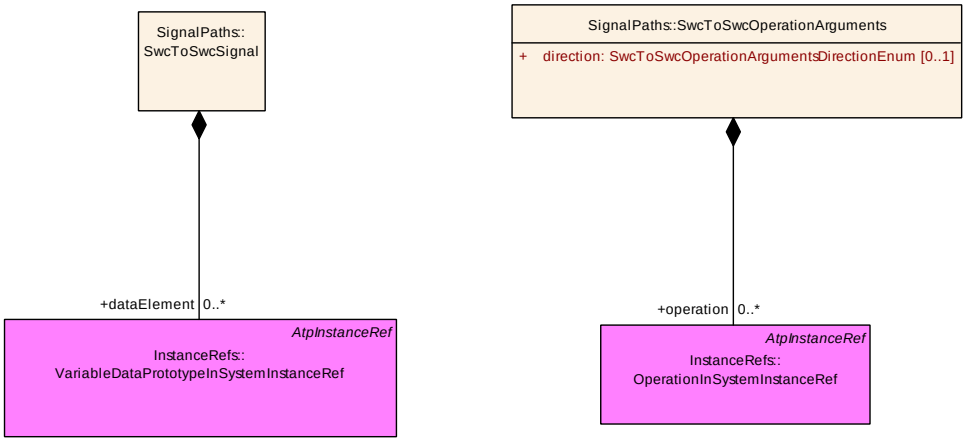

Figure B.4: SW Mapping Instance Ref Usage

B.4 Usage of InstanceRefs in PncMapping

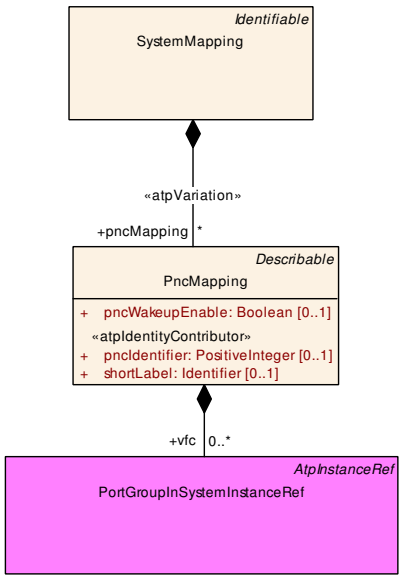

Figure B.5: Partial Network Mapping Instance Ref Usage

B.5 Usage of InstanceRefs in ComManagementMapping

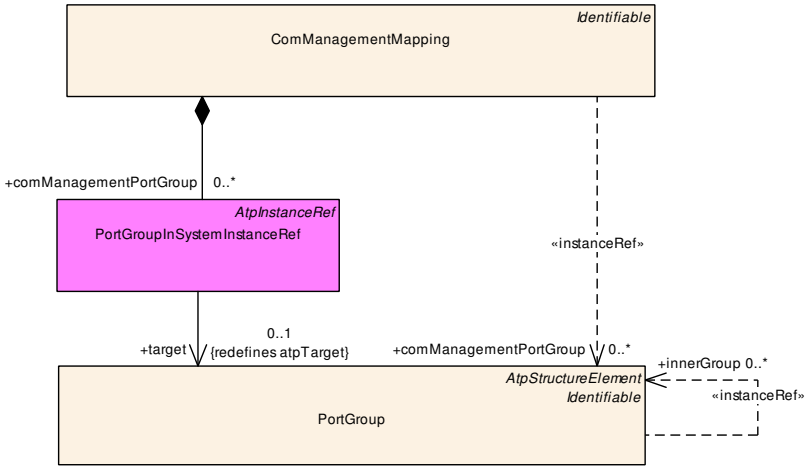

Figure B.6: ComManagementMapping Instance Ref Usage

B.6 “SWC in System” InstanceRef

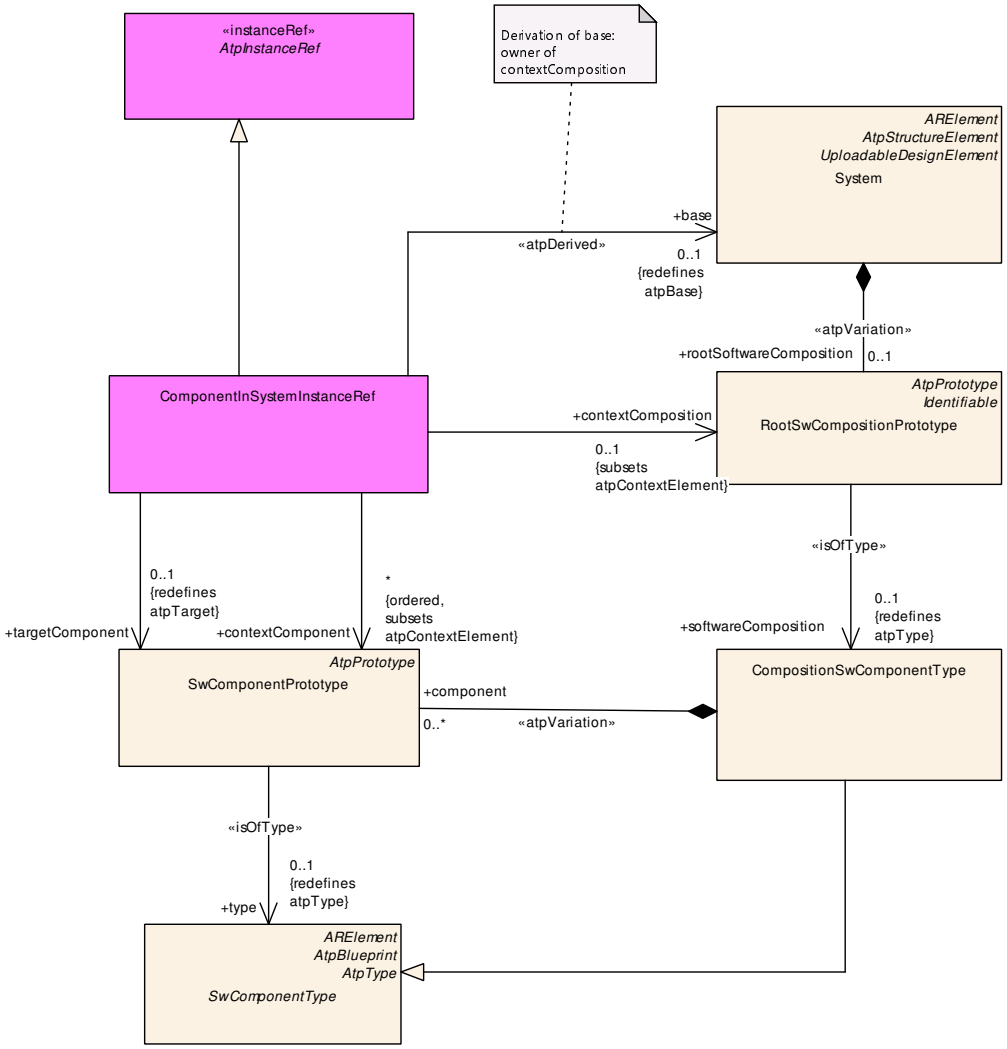

Figure B.7: ComponentInSystem InstanceRef

Class	**ComponentInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ComponentClustering.clusteredComponent, ComponentSeparation.separatedComponent, J1939 ControllerApplication.swComponentPrototype, SwComponentPrototypeAssignment.swComponent, Swc ToApplicationPartitionMapping.swComponentPrototype, SwcToEcuMapping.component, SwcToImpl Mapping.component			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30

△

Class	**ComponentInSystemInstanceRef**			
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
target Component	SwComponent Prototype	0..1	ref	**Tags:** xml.sequenceOffset=40

Table B.1: ComponentInSystemInstanceRef

B.7 “ModeDeclarationGroupPrototype in System” InstanceRef

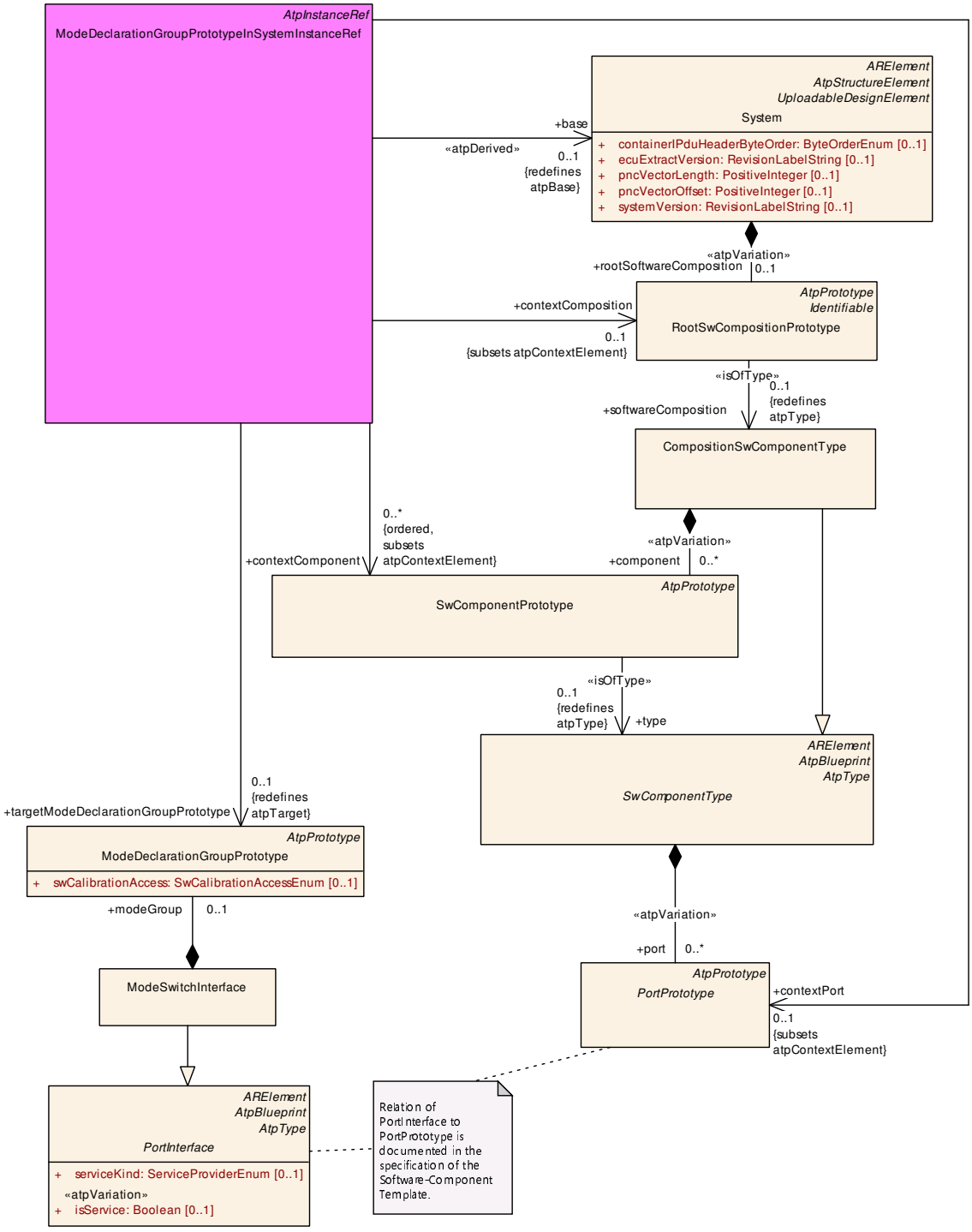

Figure B.8: ModeDeclarationGroupPrototypeInSystem InstanceRef

B.8 “Operation in System” InstanceRef

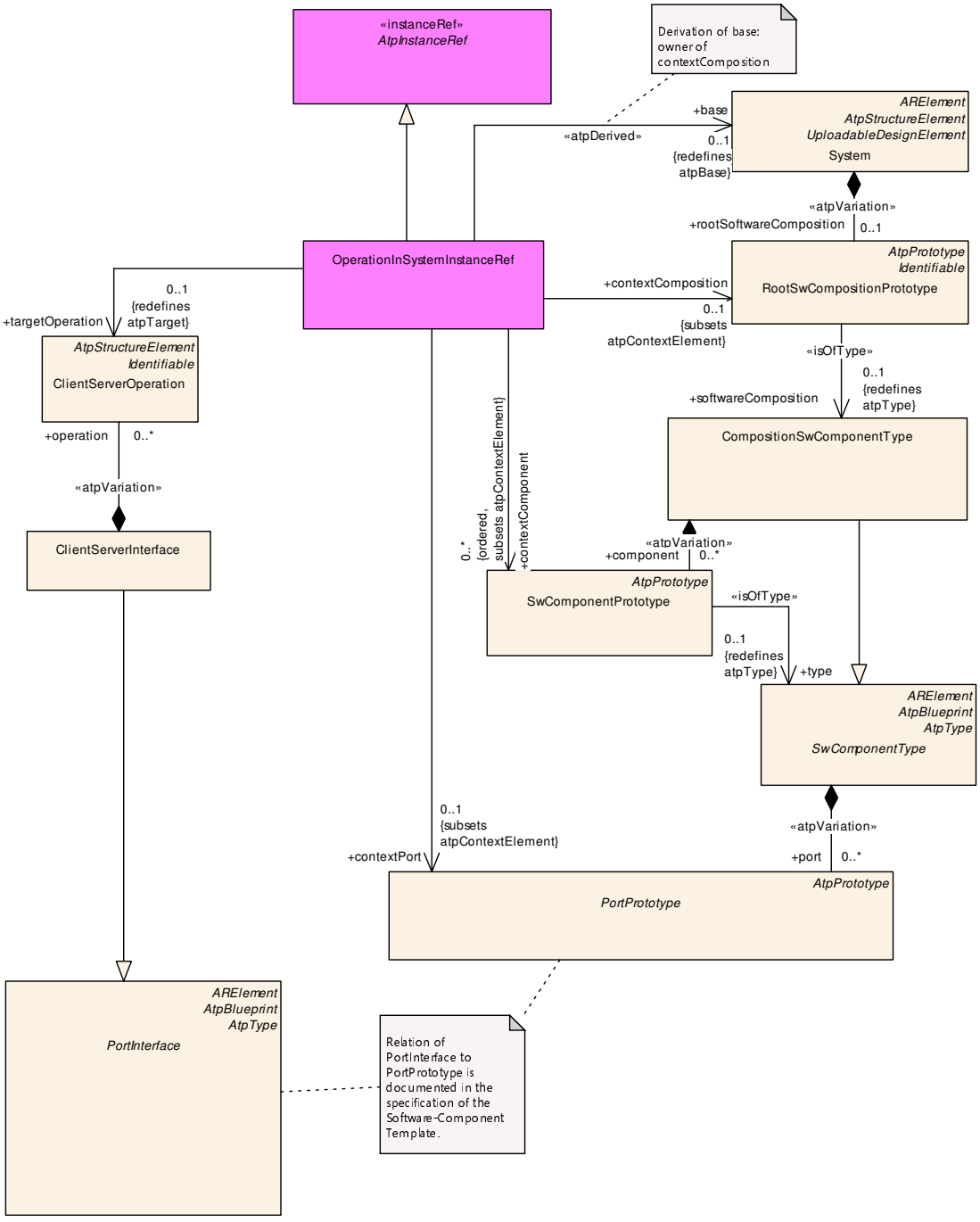

Figure B.9: OperationInSystem InstanceRef

Class	**OperationInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ClientIdDefinition.clientServerOperation, ClientServerToSignalMapping.clientServerOperation, PortElementToCommunicationResourceMapping.clientServerOperation, ServerComSpecProps.clientServerOperation, SwcToSwcOperationArguments.operation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	0..1	ref	**Tags:** xml.sequenceOffset=40
targetOperation	ClientServerOperation	0..1	ref	**Tags:** xml.sequenceOffset=50

Table B.2: OperationInSystemInstanceRef

B.9 "VariableDataPrototype" InstanceRef

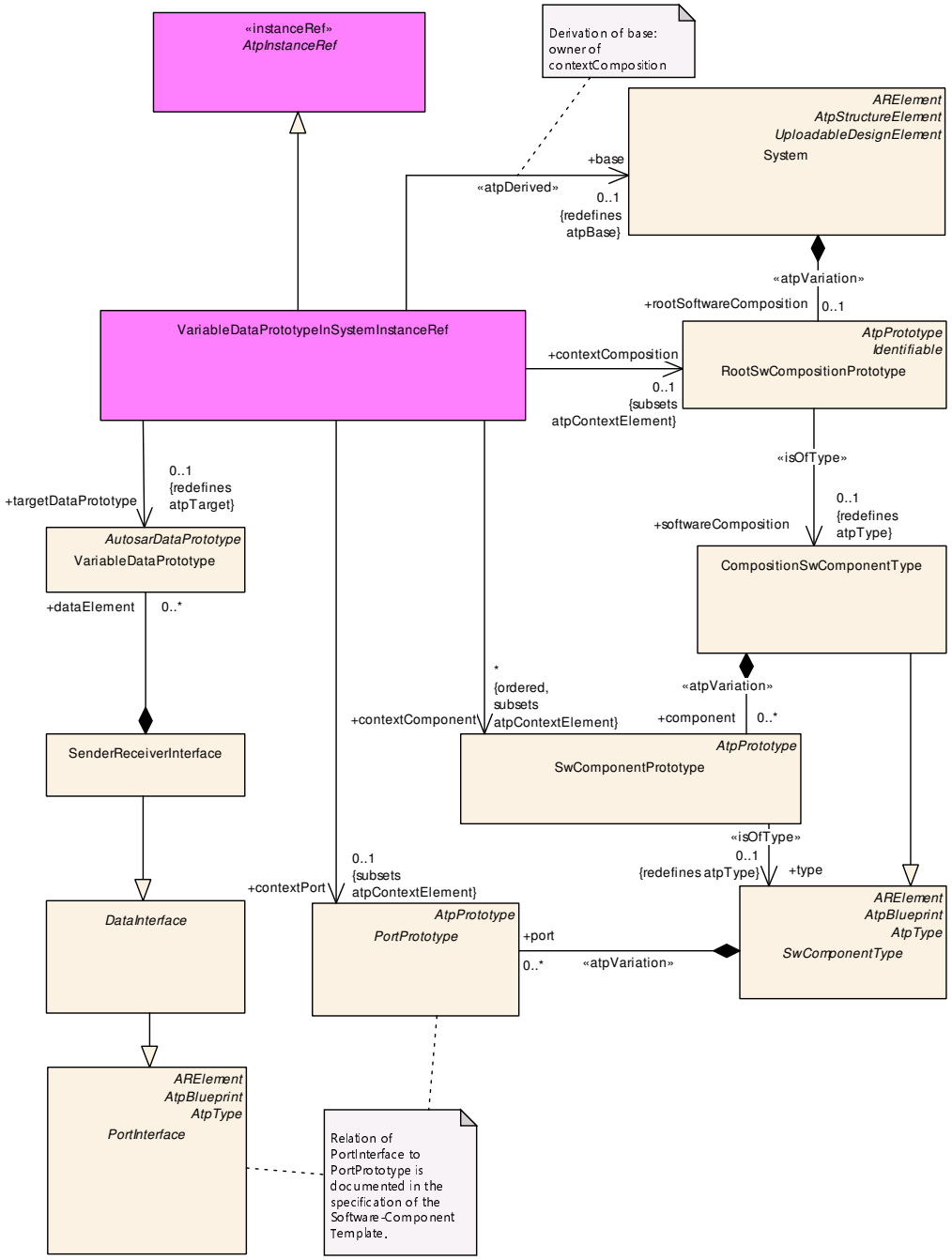

Figure B.10: VariableDataPrototypeInSystem InstanceRef

Class	**VariableDataPrototypeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype, PortElementToCommunicationResourceMapping.variableDataPrototype, QueuedReceiverComSpecProps.dataElement, SenderReceiverToSignalGroupMapping.dataElement, SenderReceiverToSignalMapping.dataElement, SignalServiceTranslationEventProps.translationTarget, SwcToSwcSignal.dataElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived
context Component (ordered)	SwComponent Prototype	*	ref	
context Composition	RootSwComposition Prototype	0..1	ref	
contextPort	PortPrototype	0..1	ref	
targetData Prototype	VariableDataPrototype	0..1	ref	

Table B.3: VariableDataPrototypeInSystemInstanceRef

Class	**ParameterDataPrototypeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	PortElementToCommunicationResourceMapping.parameterDataPrototype			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	0..1	ref	**Tags:** xml.sequenceOffset=40
targetParameter DataPrototype	ParameterData Prototype	0..1	ref	**Tags:** xml.sequenceOffset=50

Table B.4: ParameterDataPrototypeInSystemInstanceRef

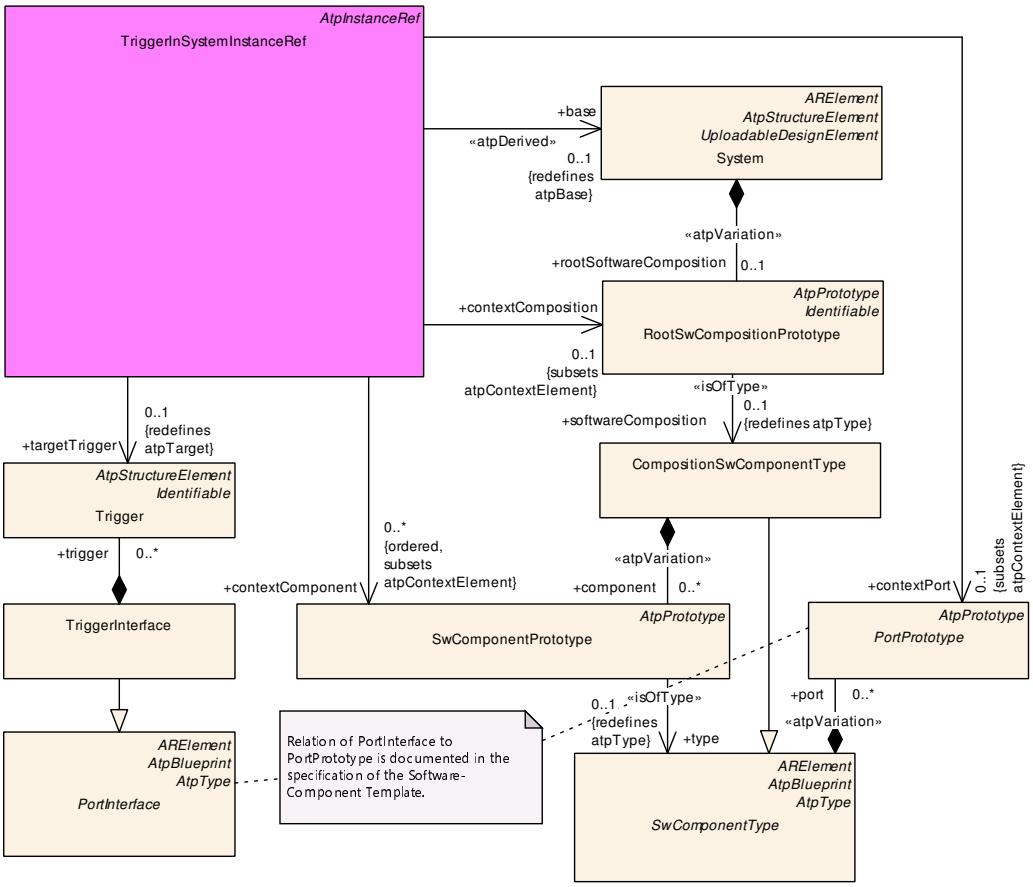

Figure B.11: `TriggerInSystemInstanceRef`

Class	**TriggerInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	PortElementToCommunicationResourceMapping.trigger, TriggerToSignalMapping.trigger			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	This represents that base of the InstanceRef **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	This represents the set of context components. The association is ordered because it needs to respect the nesting order. **Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	This represents the reference to the RootSw Compositiontype representing a context of the Instance Ref. **Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	0..1	ref	This represents the PortPrototype in which the target Trigger is located. **Tags:** xml.sequenceOffset=40
targetTrigger	Trigger	0..1	ref	This represents the target Trigger. **Tags:** xml.sequenceOffset=50

Table B.5: TriggerInSystemInstanceRef

B.10 "PortGroup in System" InstanceRef

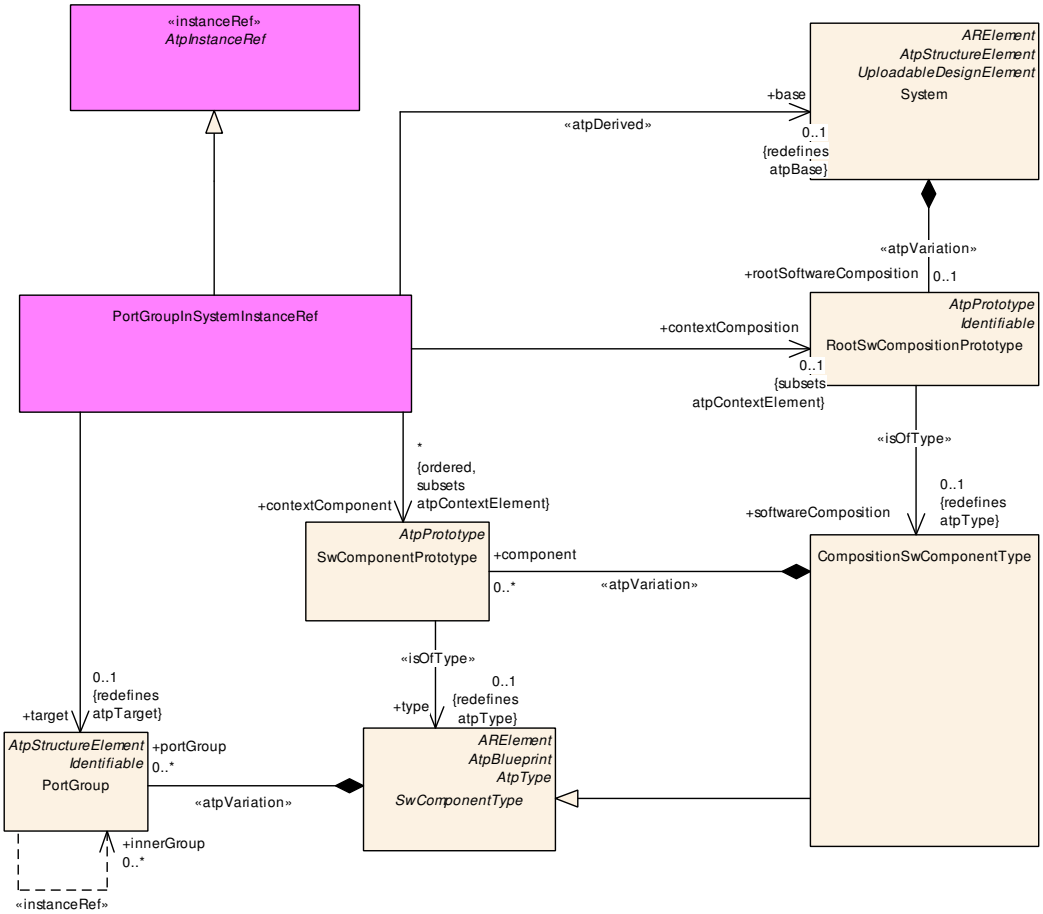

Figure B.12: PortGroupInSystem InstanceRef

Class	**PortGroupInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ComManagementMapping.comManagementPortGroup, PncMapping.vfc			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
target	PortGroup	0..1	ref	Link to a PortGroup that is defined in a component which is part of this CompositionSwComponentType. **Tags:** xml.sequenceOffset=40

Table B.6: PortGroupInSystemInstanceRef

B.11 "DataPrototype in PortInterface" InstanceRef

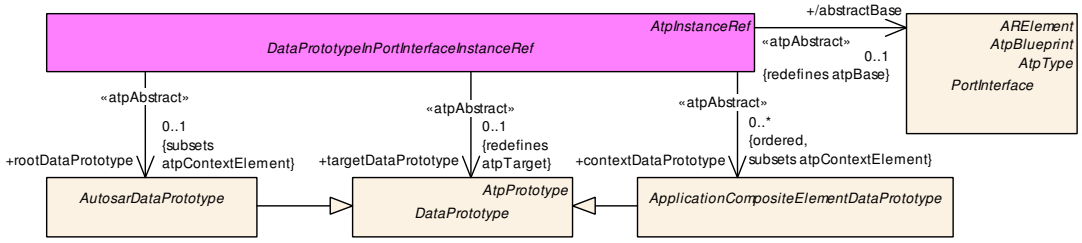

Figure B.13: DataPrototypeInPortInterfaceInstanceRef InstanceRef

Class	***DataPrototypeInPortInterfaceInstanceRef*** (abstract)			
Note	This meta-class represents the ability to: • refer to a DataPrototype in the context of a PortInterface. • refer to the internal structure of a DataPrototype which is typed by an ApplicationDatatype in the context of a PortInterface.			
Base	*ARObject*, *AtpInstanceRef*			
Subclasses	DataPrototypeInClientServerInterfaceInstanceRef, DataPrototypeInSenderReceiverInterfaceInstanceRef			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
abstractBase	PortInterface	0..1	ref	**Stereotypes:** atpAbstract
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Stereotypes:** atpAbstract **Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	**Stereotypes:** atpAbstract **Tags:** xml.sequenceOffset=10
targetData Prototype	DataPrototype	0..1	ref	**Stereotypes:** atpAbstract **Tags:** xml.sequenceOffset=30

Table B.7: DataPrototypeInPortInterfaceInstanceRef

B.12 Affinity Constraints InstanceRef

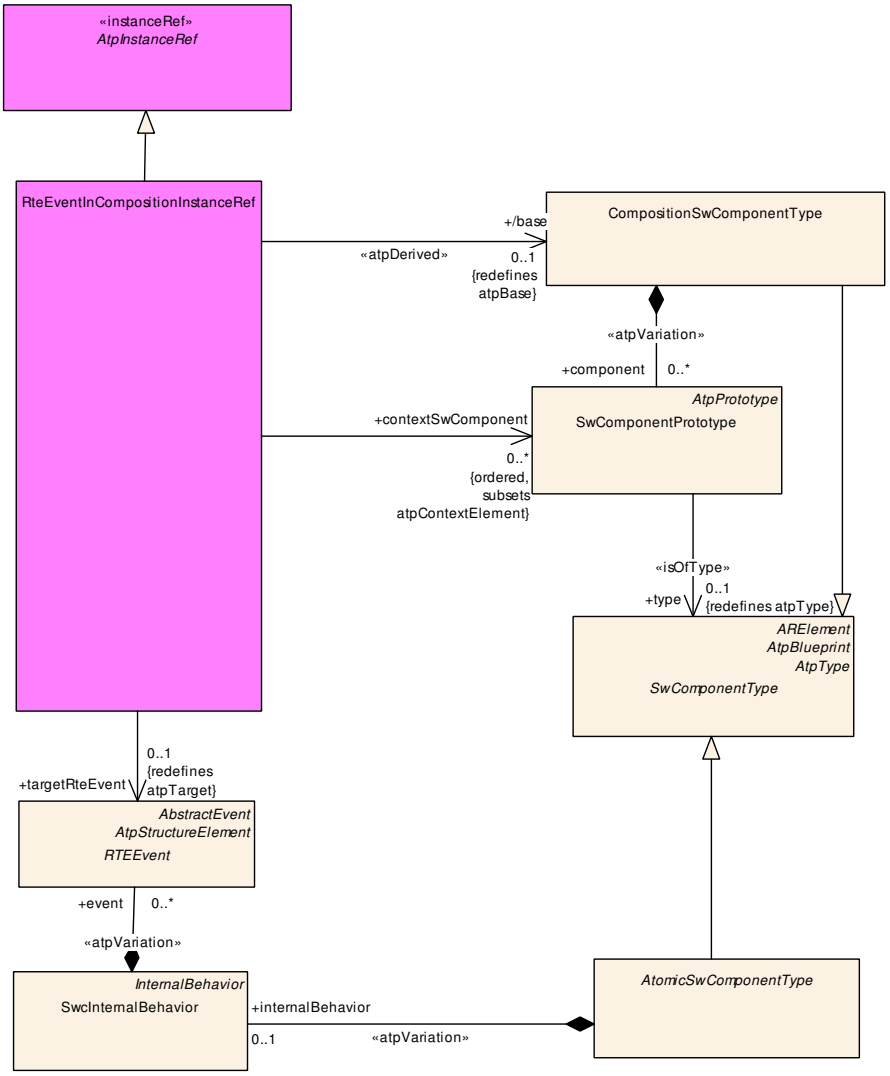

Figure B.14: RteEventInCompositionInstanceRef

Class	**RteEventInCompositionInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	RteEventInCompositionSeparation.rteEvent, RteEventInCompositionToOsTaskProxyMapping.rteEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	CompositionSw ComponentType	0..1	ref	**Stereotypes:** atpDerived
contextSw Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=20
targetRteEvent	RTEEvent	0..1	ref	**Tags:** xml.sequenceOffset=30

Table B.8: RteEventInCompositionInstanceRef

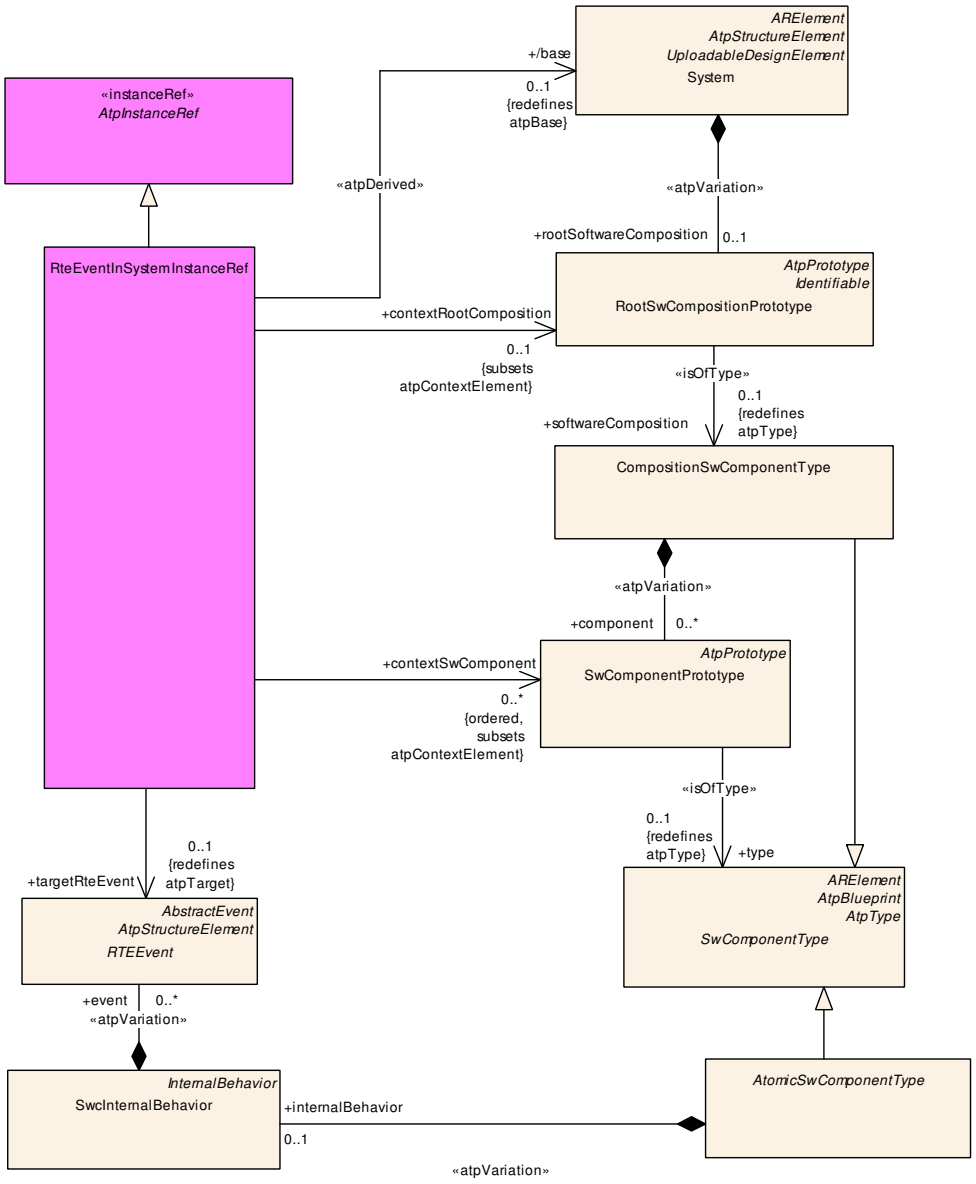

Figure B.15: RteEventInSystemInstanceRef

Class	**RteEventInSystemInstanceRef**			
Note	This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	RteEventInSystemSeparation.rteEvent, RteEventInSystemToOsTaskProxyMapping.rteEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived
contextRoot Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextSw Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
targetRteEvent	RTEEvent	0..1	ref	**Tags:** xml.sequenceOffset=40

Table B.9: RteEventInSystemInstanceRef

C Harmonization between Upstream Templates and ECU Configuration

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template and ECU Resource Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

- How shall a supplier use the information in a System Description in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an ECU Configuration Description out of ECU Extract of System Description?

In addition to adhering to the mapping rules defined in this appendix an automated generation of an ECU Configuration Description out of ECU Extract of System Description should apply a certain implementation-specific name mangling when deriving the `shortName` of the `EcucContainerValue` elements to ensure that the resulting ECU Configuration Description is valid with respect to constr_2508 of [1].

Please note that the tables contain the following columns:

bsw module: Name of BSW module

bsw context: Reference to parameter container

bsw type: Type of parameter

bsw param: Name of the BSW parameter

bsw desc: Description from the configuration document

m2 template: System Template, SW Component Template, ECU Resource Template

m2 param: Name of the upstream template parameter

m2 description: Description from the upstream template definition

mapping rule: Textual description on how to transform between M2 and BSW domains

mapping type:

- local: no mapping needed since parameter local to BSW
- partial: some data can be automatically mapped but not all
- full: all data can be automatically mapped

C.1 Can

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerBaudRate	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the baudrate of the controller in kbps.	
Template Description	
Channels speed in bits/s.	
M2 Parameter	
CommunicationCluster.baudrate	
Mapping Rule	**Mapping Type**
SystemTemplate speed is in bps, so divide it by 1000 to get kbps	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00005]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerFdBaudRate	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the data segment baud rate of the controller in kbps.	
Template Description	
Specifies the data segment baud rate of the controller in bits/s.	
M2 Parameter	
AbstractCanCluster.canFdBaudrate	
Mapping Rule	**Mapping Type**
SystemTemplate speed is in bps, so divide it by 1000 to get kbps	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00481]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerPropSeg	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies propagation delay in time quantas.	
Template Description	
Specifies propagation delay in time quantas.	
M2 Parameter	
CanControllerFdConfiguration.propSeg	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00476]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter	**BSW Type**	
CanControllerSeg1	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Specifies phase segment 1 in time quantas.		
Template Description		
Specifies phase segment 1 in time quantas.		
M2 Parameter		
CanControllerFdConfiguration.timeSeg1		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Can_00477]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter	**BSW Type**	
CanControllerSeg2	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Specifies phase segment 2 in time quantas.		
Template Description		
Specifies phase segment 2 in time quantas.		
M2 Parameter		
CanControllerFdConfiguration.timeSeg2		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Can_00478]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter	**BSW Type**	
CanControllerSspOffset	ECUC-INTEGER-PARAM-DEF	
BSW Description		



△

Specifies the Transmitter Delay Compensation Offset in minimum time quanta (see [17]). Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified, Transmitter Delay Compensation is disabled. Note: MTQ == Minimum Time Quanta in seconds == 1/(frequency of the CAN controller clock) Secondary Sample Point Offset in seconds = CanControllerSspOffset * MTQ Example: • CAN controller clock frequency = 20MHz => MTQ = 1/20 * 10^(-6) s = 0,05 us = 50ns • Baud rate = 1MBit/s => BitTime = 1/(1 * 10^6) s/Bit = 1 * 10^(-6) = 1us/Bit • SSP = 75% => SSP in seconds = 0,75 * 1us = 750 ns • CanControllerSspOffset in MTQ = 750ns / 50ns = 15 Note: Please consider the minimum range (0..63) stated in [17] and the range definition (0..127) used as per [19].
Template Description
Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
M2 Parameter
`CanControllerFdConfiguration.sspOffset`

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00494]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig

BSW Parameter	**BSW Type**
CanControllerSyncJumpWidth	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the synchronization jump width for the controller in time quantas.	
Template Description	
Specifies the synchronization jump width for the controller in time quantas.	
M2 Parameter	
`CanControllerFdConfiguration.syncJumpWidth`	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00479]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig

BSW Parameter	**BSW Type**
CanControllerTxBitRateSwitch	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if the bit rate switching shall be used for transmissions. If FALSE: CAN FD frames shall be sent without bit rate switching.	
Template Description	

▽

△

Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.	
M2 Parameter	
CanControllerFdConfiguration.txBitRateSwitch	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00475]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerPropSeg	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies propagation delay in time quantas.	
Template Description	
Specifies propagation delay in time quantas.	
M2 Parameter	
CanControllerConfiguration.propSeg	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00073]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerSeg1	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies phase segment 1 in time quantas.	
Template Description	
Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1	
M2 Parameter	
CanControllerConfiguration.timeSeg1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00074]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerSeg2	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies phase segment 2 in time quantas.	
Template Description	
Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2	
M2 Parameter	
CanControllerConfiguration.timeSeg2	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00075]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerSyncJumpWidth	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the synchronization jump width for the controller in time quantas.	
Template Description	
The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.	
M2 Parameter	
CanControllerConfiguration.syncJumpWidth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00383]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController
BSW Parameter	**BSW Type**
CanControllerDefaultBaudrate	ECUC-REFERENCE-DEF
BSW Description	
Reference to baudrate configuration container configured for the Can Controller.	
Template Description	
Channels speed in bits/s.	
M2 Parameter	
CommunicationCluster.baudrate	
Mapping Rule	**Mapping Type**



△

Set the reference to the container of the CanControllerBaudRate parameter that has been configured for SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00435]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController
BSW Parameter	**BSW Type**
CanFdPaddingValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the value which is used to pad unspecified data in CAN FD frames > 8 bytes for transmission. This is necessary due to the discrete possible values of the DLC if > 8 bytes. If the length of a PDU which was requested to be sent does not match the allowed DLC values, the remaining bytes up to the next possible value shall be padded with this value.	
Template Description	
CanControllerFdConfiguration.paddingValue: Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD. **CanControllerFdConfigurationRequirements.paddingValue:** Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.	
M2 Parameter	
CanControllerFdConfiguration.paddingValue, CanControllerFdConfigurationRequirements.paddingValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00485]

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController
BSW Parameter	**BSW Type**
CanWakeupSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
CAN driver support for wakeup over CAN Bus.	
Template Description	
Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
CommunicationController.wakeUpByControllerSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00330]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject
BSW Parameter	**BSW Type**
CanIdType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies whether the IdValue is of type standard identifier, extended identifier or mixed mode. ImplementationType: Can_IdType	
Template Description	
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
M2 Parameter	
CanFrameTriggering.canAddressingMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Can_00065]

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanIdType
BSW Parameter	**BSW Type**
EXTENDED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
All the CANIDs are of type extended only (29 bit).	
Template Description	
Extended 29-bit-identifiers are used (CAN 2.0B)	
M2 Parameter	
CanAddressingModeType.extended	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanIdType
BSW Parameter	**BSW Type**
MIXED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The type of CANIDs can be both Standard or Extended.	
Template Description	
The type of CANIDs can be both Standard or Extended.	
M2 Parameter	
CanAddressingModeType.mixed	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanIdType	
BSW Parameter		**BSW Type**
STANDARD		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
All the CANIDs are of type standard only (11bit).		
Template Description		
Standard 11-bit-identifiers are used (CAN 2.0A)		
M2 Parameter		
`CanAddressingModeType.standard`		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanObjectPayloadLength		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Specifies the maximum L-PDU payload length in bytes the hardware object can store. If the parameter is not provided, Can driver configuration generators have to assume the maximum length of the underlying CAN derivate, e.g. 8 bytes for CAN, 64 bytes for CAN-FD.		
Template Description		
The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.		
M2 Parameter		
`CanFrameTriggering.canAddressingMode`		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Can_00495]

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength	
BSW Parameter		**BSW Type**
CAN_OBJECT_PL_12		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Payload length of 12 Bytes		
Template Description		
Standard 11-bit-identifiers are used (CAN 2.0A)		

▽

△

M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_16	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 16 Bytes	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_20	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 20 Bytes	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_24	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 24 Bytes	
Template Description	

▽

△

Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_32	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 32 Bytes	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_48	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 48 Bytes	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_64	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 64 Bytes	

▽

△

Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Can	Can/CanConfigSet/CanHardwareObject/CanObjectPayloadLength
BSW Parameter	**BSW Type**
CAN_OBJECT_PL_8	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Payload length of 8 Bytes	
Template Description	
Standard 11-bit-identifiers are used (CAN 2.0A)	
M2 Parameter	
CanAddressingModeType.standard	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

C.2 CanIf

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfRxPduCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
CAN Identifier of Receive CAN L-PDUs used by the CAN Interface. Exa: Software Filtering. This parameter is used if exactly one Can Identifier is assigned to the Pdu. If a range is assigned then the CanIfRxPduCanIdRange parameter shall be used. Range: ExtId (bit 31), CanFd (bit 30), and 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier	
Template Description	
CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network. **CanFrameTriggering.canFrameRxBehavior:** Defines which CAN protocol shall be expected for frame reception. **CanFrameTriggering.canAddressingMode:** The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.	
M2 Parameter	

▽

△

CanFrameTriggering.identifier, CanFrameTriggering.canFrameRxBehavior, CanFrameTriggering.canAddressingMode	
Mapping Rule	**Mapping Type**
Bit 31 = (CanFrameTriggering.canAddressingMode == extended ? 1 : 0); Bit 30 = (CanFrame Triggering.canFrameRxBehavior == any | canFd ? 1 : 0); Bits 0 .. 29 = CanFrame Triggering.identifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00598]

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfRxPduCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter defines a CAN Identifier range in an alternative way to CanIfRxPduCanIdRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier. Range: ExtId (bit 31), CanFd (bit 30), and 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.	
Template Description	
CanFrameTriggering.rxMask: Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range. **CanFrameTriggering.canFrameRxBehavior:** Defines which CAN protocol shall be expected for frame reception.	
M2 Parameter	
CanFrameTriggering.rxMask, CanFrameTriggering.canFrameRxBehavior	
Mapping Rule	**Mapping Type**
Bit 30 = (CanFrameTriggering.canFrameRxBehavior == any ? 0 : 1); Bits 0 .. 29, 31 = CanFrame Triggering.rxMask	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00822]

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfRxPduCanIdRange	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Optional container that allows to map a range of CAN Ids to one PduId.	
Template Description	
Optional definition of a CanId range to reduce the effort of specifying every possible FrameTriggering within the defined Id range during reception. All frames received within a range are mapped to the same Pdu that is passed to a upper layer module (e.g. Nm, CDD, PduR).	
M2 Parameter	
RxIdentifierRange	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00743]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange
BSW Parameter	**BSW Type**
CanIfRxPduCanIdRangeLowerCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.	
Template Description	
This attribute can be used together with the upperCanId attribute to define a range of CanIds.	
M2 Parameter	
RxIdentifierRange.lowerCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00745]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange
BSW Parameter	**BSW Type**
CanIfRxPduCanIdRangeUpperCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.	
Template Description	
This attribute can be used together with the lowerCanId attribute to define a range of CanIds.	
M2 Parameter	
RxIdentifierRange.upperCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00744]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfRxPduDataLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Data length of the received CAN L-PDUs used by the CAN Interface. This information is used for Data Length Check. Additionally it might specify the valid bits in case of the discrete DLC for CAN FD L-PDUs > 8 bytes. The data area size of a CAN L-PDU can have a range from 0 to 2048 bytes.	
Template Description	
The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay). The frameLength of zero bytes is allowed. Please consider also TPS_SYST_02255.	
M2 Parameter	
Frame.frameLength	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00599]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg
BSW Parameter	**BSW Type**
CanIfTxPduCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
CAN Identifier of transmit CAN L-PDUs used by the CAN Driver for CAN L-PDU transmission. Range: ExtId (bit 31), CanFd (bit 30), and 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier The CAN Identifier may be omitted for dynamic transmit L-PDUs and CAN XL PDUs.	
Template Description	
CanFrameTriggering.identifier: This attribute is used to define the identifier this frame shall use on the CAN network. **CanFrameTriggering.canAddressingMode:** The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B. **CanFrameTriggering.canFrameTxBehavior:** Defines which CAN protocol shall be used for frame transmission.	
M2 Parameter	
`CanFrameTriggering.identifier`, `CanFrameTriggering.canAddressingMode`, `CanFrameTriggering.canFrameTxBehavior`	
Mapping Rule	**Mapping Type**
Bit 31 = (CanFrameTriggering.canAddressingMode == extended ? 1 : 0); Bit 30 = (CanFrame Triggering.canFrameTxBehavior == canFd ? 1 : 0); Bits 0 .. 29 = CanFrameTriggering.identifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00592]

BSW Module	BSW Context
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg
BSW Parameter	**BSW Type**
CanIfTxPduCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter may be used to keep parts of the CAN Identifier of dynamic transmit L-PDUs static. Range: ExtId (bit 31), CanFd (bit 30), and 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.	
Template Description	
Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.	
M2 Parameter	
`CanFrameTriggering.txMask`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanIf_00823]

C.3 CanNm

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmBusLoadReductionEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling busload reduction support.	
Template Description	
Enables busload reduction support	
M2 Parameter	
CanNmClusterCoupling.nmBusloadReductionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00040]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmBusSynchronizationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only.	
Template Description	
Enables bus synchronization support.	
M2 Parameter	
NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00006]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmChannelConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the channel specific configuration parameter of the CanNm.	
Template Description	
Can specific NmCluster attributes	
M2 Parameter	
CanNmCluster	
Mapping Rule	**Mapping Type**
Create container for each existing CanNmCluster.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00017]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmAllNmMessagesKeepAwake	ECUC-BOOLEAN-PARAM-DEF

BSW Description

Specifies if CanNm drops irrelevant NM PDUs.

false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling true: Every NM PDU triggers the standard RX indication handling

Template Description

Specifies if Nm drops irrelevant NM PDUs.

false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake

true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake

M2 Parameter

`CanNmNode.allNmMessagesKeepAwake`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00068]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmBusLoadReductionActive	ECUC-BOOLEAN-PARAM-DEF

BSW Description

This parameter defines if bus load reduction for the respective NM channel is active or not.

Template Description

It determines if bus load reduction for the respective CanNm channel is active or not.

M2 Parameter

`CanNmCluster.nmBusloadReductionActive`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00042]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmCarWakeUpBitPosition	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Specifies the Bit position of the CWU within the NM PDU.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
CanNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNm CarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00075]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmCarWakeUpBytePosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the Byte position of the CWU within the NM PDU.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
CanNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNm CarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NM PDU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00076]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmCarWakeUpFilterEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNode Id is considered as CWU request. FALSE - CWU filtering is not supported TRUE - CWU filtering is supported	
Template Description	
If this attribute is set to true the CareWakeUp filtering is supported.	
M2 Parameter	
CanNmNode.nmCarWakeUpFilterEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00077]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmCarWakeUpFilterNodeId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNodeId is considered as CWU request.	
Template Description	
Source node identifier for CarWakeUp filtering.	
M2 Parameter	
CanNmCluster.nmCarWakeUpFilterNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00078]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmCarWakeUpRxEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of CarWakeUp bit evaluation in received NM PDUs. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported	
Template Description	
If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.	
M2 Parameter	
CanNmNode.nmCarWakeUpRxEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00074]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmDynamicPncToChannelMappingEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	



△

Mapping Rule	Mapping Type
1:1 Mapping. If M2 Parameter not defined then do not create CanNmDynamicPncToChannel MappingEnabled	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00093]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmImmediateNmCycleTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the immediate NM PDU cycle time in seconds which is used for CanNmImmediateNmTransmissions NM PDU transmissions.	
Template Description	
Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This parameter is only valid if CanNmImmediateNmTransmissions is greater one.	
M2 Parameter	
CanNmCluster.nmImmediateNmCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00057]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmImmediateNmTransmissions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immeditate NM PDUs is defined by CanNmImmediateNmCycleTime.	
Template Description	
Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.	
M2 Parameter	
CanNmCluster.nmImmediateNmTransmissions	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00056]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**

▽

△

CanNmMsgCycleOffset	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
Template Description	
Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
M2 Parameter	
CanNmNode.nmMsgCycleOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00029]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgCycleTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Period of a NM PDU in seconds. It determines the periodic rate in the "periodic transmission mode with bus load reduction" and is the basis for transmit scheduling in the "periodic transmission mode without bus load reduction".	
Template Description	
Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.	
M2 Parameter	
CanNmCluster.nmMsgCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00028]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgReducedTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.	
Template Description	
Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.	
M2 Parameter	
CanNmNode.nmMsgReducedTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00043]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgTimeoutTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
When using Partial Network and this timeout is defined then CanNm monitors that a NM-PDU is transmitted successfully within this Transmission Timeout Time and provides an error notification otherwise.	
Template Description	
Timeout of an NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.	
M2 Parameter	
CanNmCluster.nmMessageTimeoutTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00030]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmNodeDetectionEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the node detection feature.	
Template Description	
Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter	
NmCluster.nmNodeDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00088]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmNodeId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Node identifier of local node.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
NmNode.nmNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00031]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmNodeIdEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the source node identifier.	
Template Description	
Enables the source node identifier.	
M2 Parameter	
NmCluster.nmNodeIdEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00090]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmPduCbvPosition	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the position of the control bit vector within the NM PDU. The value of the parameter represents the location of the Control Bit Vector in the NM PDU (CanNmPduByte0 means byte 0, CanNmPduByte1 means byte 1, CanNmPduOff means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType	
Template Description	
Defines the position of the control bit vector within the NmPdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.	
M2 Parameter	
CanNmCluster.nmCbvPosition	
Mapping Rule	**Mapping Type**
Derive byte position from nmCbvPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00026]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmPduNidPosition	ECUC-ENUMERATION-PARAM-DEF
BSW Description	

▽

△

Defines the position of the source node identifier within the NM PDU. The value of the parameter represents the location of the source node identifier in the NM PDU (CANNM_PDU_BYTE_0 means byte 0, CANNM_PDU_BYTE_1 means byte 1, CANNM_PDU_OFF means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType	
Template Description	
Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.	
M2 Parameter	
CanNmCluster.nmNidPosition	
Mapping Rule	**Mapping Type**
Derive byte position from nmNidPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00025]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmPnEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
NmCluster.nmPncParticipation	
Mapping Rule	**Mapping Type**
If NmCluster.nmPncParticipation has the value "true" or is not defined then CanNmPnEnabled shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00066]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmPnHandleMultipleNetworkRequests	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if CanNm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).	
Template Description	
Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).	
M2 Parameter	
CanNmCluster.nmPnHandleMultipleNetworkRequests	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanNm_00073]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRemoteSleepIndTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep.	
Template Description	
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.	
M2 Parameter	
CanNmCluster.nmRemoteSleepIndicationTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00023]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRepeatMessageTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.	
Template Description	
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
M2 Parameter	
CanNmCluster.nmRepeatMessageTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00022]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRepeatMsgIndEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable/disable the notification that a RepeatMessageRequest bit has been received.	
Template Description	

▽

△

Switch for enabling the Repeat Message Bit Indication.	
M2 Parameter	
NmCluster.nmRepeatMsgIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00089]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is used to configure the Rx PDU properties that are used for the CanNm Channel.	
Template Description	
M2 Parameter	
NmNode.rxNmPdu	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that is received on the regarded Nm cluster	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00038]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmTimeoutTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
If NM is in Ready Sleep State it denotes the time in seconds how long after the last NM PDU transmission or reception state transition into the Prepare Bus-Sleep Mode is initiated. If NM is in Repeat Message or Normal Operation state and no NM PDU can be transmitted or received within this time, a run-time error is raised.	
Template Description	
Network Timeout for NmPdus in seconds It denotes the time how long the CanNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.	
M2 Parameter	
CanNmCluster.nmNetworkTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00020]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This container contains the CanNmTxConfirmationPduId and the CanNmTxPduRef.

Template Description

M2 Parameter

`NmNode.txNmPdu`

Mapping Rule	Mapping Type
Create container for each NmPdu that is transmitted on the regarded Nmcluster	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00036]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmUserDataTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This optional container is used to configure the UserNm PDU. This container is only available if CanNmComUserDataSupport is enabled.

Template Description

An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.

M2 Parameter

`ISignalToIPduMapping`

Mapping Rule	Mapping Type
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00045]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig

BSW Parameter	BSW Type
CanNmWaitBusSleepTime	ECUC-FLOAT-PARAM-DEF

BSW Description

Timeout for bus calm down phase. It denotes the time in seconds how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.

Template Description

Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.

M2 Parameter

`CanNmCluster.nmWaitBusSleepTime`

Mapping Rule	Mapping Type



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00021]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmComControlEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
NmEcu.nmComControlEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00013]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmComUserDataSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.	
Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_00044]

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmDynamicPncToChannelMappingSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	

▽

△

Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
If at least one dynamicPncToChannelMappingEnabled attribute is defined and if at least one CommunicationConnector of the EcuInstance has dynamicPncToChannelMappingEnabled set to true, then CanNmDynamicPncToChannelMappingSupport shall be set to true. Otherwise CanNm DynamicPncToChannelMappingSupport shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00094]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmImmediateRestartEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the immediate transmission of a NM PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
Template Description	
Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
M2 Parameter	
CanNmClusterCoupling.nmImmediateRestartEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00009]

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmPassiveModeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling support of the Passive Mode.	
Template Description	
Enables support of the Passive Mode. The passive mode is configurable per channel.	
M2 Parameter	
NmNode.nmPassiveModeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping. nmNode.nmPassiveModeEnabled shall always have the same value in all Nm Clusters with the same bus protocol in the scope of one EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanNm_ 00010]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPduRxIndicationEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication.		
Template Description		
Switch for enabling the PDU Rx Indication.		
M2 Parameter		
NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanNm_00011]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmRemoteSleepIndEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only.		
Template Description		
Switch for enabling remote sleep indication support.		
M2 Parameter		
NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanNm_00055]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmStateChangeIndEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Pre-processor switch for enabling the CAN NM state change notification.		
Template Description		
Enables the CAN Network Management state change notification.		
M2 Parameter		
NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanNm_00012]

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmUserDataEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Pre-processor switch for enabling user data support.		
Template Description		
Switch for enabling user data support.		
M2 Parameter		
NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanNm_00004]

C.4 CanSM

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter		**BSW Type**
CanSMBorCounterL1ToL2		ECUC-INTEGER-PARAM-DEF
BSW Description		
This threshold defines the count of bus-offs after which the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).		
Template Description		
This threshold defines the count of bus-offs after which the bus-off recovery switches from level 1 (short recovery time) to level 2 (long recovery time).		
M2 Parameter		
CanClusterBusOffRecovery.borCounterL1ToL2		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanSM_00131]

BSW Module	BSW Context	
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork	
BSW Parameter		**BSW Type**
CanSMBorTimeL1		ECUC-FLOAT-PARAM-DEF
BSW Description		
This time parameter defines in seconds the duration of the bus-off recovery time in level 1 (short recovery time).		
Template Description		
This attribute defines the duration of the bus-off recovery time in level 1 (short recovery time) in seconds.		

▽

△

M2 Parameter	
CanClusterBusOffRecovery.borTimeL1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanSM_00128]

BSW Module	BSW Context
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork
BSW Parameter	**BSW Type**
CanSMBorTimeL2	ECUC-FLOAT-PARAM-DEF
BSW Description	
This time parameter defines in seconds the duration of the bus-off recovery time in level 2 (long recovery time).	
Template Description	
This attribute defines the duration of the bus-off recovery time in level 2 (long recovery time) in seconds.	
M2 Parameter	
CanClusterBusOffRecovery.borTimeL2	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanSM_00129]

BSW Module	BSW Context
CanSM	CanSM/CanSMConfiguration/CanSMManagerNetwork
BSW Parameter	**BSW Type**
CanSMBorTimeTxEnsured	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines in seconds the duration of the bus-off event check. This check assesses, if the recovery has been successful after the recovery reenables the transmit path. If a new bus-off occurs during this time period, the CanSM assesses this bus-off as sequential bus-off without successful recovery. Because a bus-off only can be detected, when PDUs are transmitted, the time has to be great enough to ensure that PDUs are transmitted again (e. g. time period of the fastest cyclic transmitted PDU of the COM module, ComTxModeTimePeriod).	
Template Description	
This attribute defines the duration of the bus-off event check in seconds.	
M2 Parameter	
CanClusterBusOffRecovery.borTimeTxEnsured	
Mapping Rule	**Mapping Type**
If borTimeTxEnsured is defined set this parameter to true otherwise to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanSM_00130]

BSW Module	BSW Context	
CanSM	CanSM/CanSMGeneral	
BSW Parameter		**BSW Type**
CanSMMainFunctionTimePeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
This parameter defines the cycle time of the function CanSM_MainFunction in seconds		
Template Description		
This attribute defines the cycle time of the function CanSM_MainFunction in seconds.		
M2 Parameter		
CanClusterBusOffRecovery.mainFunctionPeriod		
Mapping Rule		**Mapping Type**
The value that is defined in the System Extract defines the upperbound of the cycle time. The integrator may choose a smaller value.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanSM_00312]

C.5 CanTSyn

BSW Module	BSW Context	
CanTSyn	CanTSyn	
BSW Parameter		**BSW Type**
CanTSynGlobalTimeDomain		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This represents the existence of a global time domain on CAN. The CanTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the CanTSyn exists it is assumed that at least one global time domain exists.		
Template Description		
This represents the ability to define a global time domain.		
M2 Parameter		
GlobalTimeDomain		
Mapping Rule		**Mapping Type**
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the EcucInstance for which the ECU configuration is created.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTSyn_00004]

BSW Module	BSW Context	
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain	
BSW Parameter		**BSW Type**
CanTSynGlobalTimeDomainId		ECUC-INTEGER-PARAM-DEF
BSW Description		

▽

△

The global time domain ID.	
Template Description	
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.	
M2 Parameter	
GlobalTimeDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00005]

BSW Module	**BSW Context**
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
CanTSynGlobalTimeFupDataIDList	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
CanGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00025]

BSW Module	**BSW Context**
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeFupDataIDList
BSW Parameter	**BSW Type**
CanTSynGlobalTimeFupDataIDListElement	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Element of the DataIDList for FUP messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
CanGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	**Mapping Type**
Value shall be derived from element of the ordered fupDataIDList.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00031]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
CanTSynGlobalTimeMaster	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a Time Master for a Time Domain (refer to parent container). If CanTSynGlobalTimeMaster container exists, the local ECU acts as a Time Master for the Time Domain.	
Template Description	
This represents the generic concept of a global time master.	
M2 Parameter	
GlobalTimeMaster	
Mapping Rule	**Mapping Type**
The existence of the CanTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00007]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
CanTSynCyclicMsgResumeTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds	
Template Description	
Defines the minimum time between an "immediate" message and the next periodic message.	
M2 Parameter	
GlobalTimeMaster.immediateResumeTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00044]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
CanTSynGlobalTimeDebounceTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents the configuration of a TX debounce time for SYNC and FUP messages compared to a message before with the same PDU. Unit: seconds	
Template Description	
Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	
GlobalTimeDomain.debounceTime	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00045]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeMasterPdu
BSW Parameter	**BSW Type**
CanTSynGlobalTimePduRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated Global TimeSlaves.	
M2 Parameter	
GlobalTimeDomain.pduTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00027]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
CanTSynGlobalTimeTxCrcSecured	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This represents the configuration of whether or not CRC is supported.	
Template Description	
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.	
M2 Parameter	
GlobalTimeCanMaster.crcSecured	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00015]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTxCrcSecured
BSW Parameter	**BSW Type**
CRC_NOT_SUPPORTED	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
This represents a configuration where CRC is not supported.	
Template Description	
This indicates that CRC is not supported	
M2 Parameter	
GlobalTimeCrcSupportEnum.crcNotSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTxCrcSecured
BSW Parameter	**BSW Type**
CRC_SUPPORTED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
This represents a configuration where CRC is supported.	
Template Description	
This indicates that CRC is supported	
M2 Parameter	
GlobalTimeCrcSupportEnum.crcSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
CanTSynGlobalTimeTxIcvGeneration	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container collects configuration that shall be used for ICV generation.	
Template Description	
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.	
M2 Parameter	
GlobalTimeMaster.icvSecured	
Mapping Rule	**Mapping Type**
Create this container if GlobalTimeMaster.icvSecured is defined as the value GlobalTimeIcvSupportEnum.icvSupported.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00060]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTxIcv Generation
BSW Parameter	**BSW Type**
CanTSynIcvGenerationFvIdRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the FV taken to generate the ICV generation.	
Template Description	
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeMaster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00061]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster/CanTSynGlobalTimeTxIcv Generation
BSW Parameter	**BSW Type**
CanTSynIcvGenerationJobRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the CSM job to fetch the CSM job ID.	
Template Description	
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.	
M2 Parameter	
GlobalTimeMaster.icvSecured	
Mapping Rule	**Mapping Type**
If GlobalTimeMaster.icvSecured is set to the value icvSupported, then the referenced CsmJob needs to be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00064]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
CanTSynGlobalTimeTxPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents configuration of the TX period. Unit: seconds	
Template Description	
This represents the period. Unit: seconds	
M2 Parameter	
GlobalTimeMaster.syncPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTSyn_00017]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
CanTSynGlobalTimeNetworkSegmentId	ECUC-INTEGER-PARAM-DEF
BSW Description	
This represents the numerical identifier of the network on system level scope where this Global Time has been communicated on.	
Template Description	
This attribute represents the numerical identifier of a PhysicalChannel on system level scope.	
M2 Parameter	
NetworkSegmentIdentification.networkSegmentId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00052]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
CanTSynGlobalTimeSlave	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a Time Slave for a Time Domain (refer to parent container). If CanTSynGlobalTimeSlave container exists, the local ECU acts as a Time Slave for the Time Domain.	
Template Description	
This represents the generic concept of a global time slave.	
M2 Parameter	
GlobalTimeSlave	
Mapping Rule	**Mapping Type**
The existence of the CanTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00012]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
CanTSynGlobalTimeFollowUpTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Rx timeout for the follow-up message. This is only relevant for selected bus systems Unit:seconds	

▽

△

Template Description	
Rx timeout for the follow-up message.	
M2 Parameter	
GlobalTimeSlave.followUpTimeoutValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00006]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
CanTSynGlobalTimeRxIcvVerification	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container collects configuration required for ICV verification.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
Create this container if GlobalTimeSlave.icvVerification is defined as the value GlobalTimeIcv VerificationEnum. icvVerified or GlobalTimeIcvVerificationEnum.icvOptional.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00076]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeRxIcv Verification
BSW Parameter	**BSW Type**
CanTSynIcvVerificationAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the number of ICV verification attempts that are to be carried out when the verification of the ICV failed for a given FUP message. If zero is set, then only one ICV verification attempt is done.	
Template Description	
This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given message. If zero is set than only one authentication attempt is done.	
M2 Parameter	
SecOcSecureComProps.authenticationVerifyAttempts	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00082]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeRxIcv Verification
BSW Parameter	**BSW Type**
CanTSynIcvVerificationFvIdRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the FV taken to generate the ICV generation.	
Template Description	
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeSlave.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00077]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeRxIcv Verification
BSW Parameter	**BSW Type**
CanTSynIcvVerificationJobRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the CSM job to fetch the CSM job ID.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
If GlobalTimeSlave.icvVerification is set to the value icvVerified or icvOptional, then the referenced CsmJob needs to be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00080]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
CanTSynGlobalTimeSequenceCounterJumpWidth	ECUC-INTEGER-PARAM-DEF
BSW Description	
The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC messages.	
Template Description	
Specifies the maximum allowed gap of the sequence counter between two SYNC messages.	
M2 Parameter	
GlobalTimeCanSlave.sequenceCounterJumpWidth	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00011]

BSW Module	**BSW Context**
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynGlobalTimeSlavePdu
BSW Parameter	**BSW Type**
CanTSynGlobalTimePduRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated Global TimeSlaves.	
M2 Parameter	
GlobalTimeDomain.pduTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00027]

BSW Module	**BSW Context**
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
CanTSynRxCrcValidated	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of whether or not validation of the CRC is supported.	
Template Description	
Definition of whether or not validation of the CRC is supported.	
M2 Parameter	
GlobalTimeCanSlave.crcValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00021]

BSW Module	**BSW Context**
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_IGNORED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts Time Synchronization messages, which are CRC secured (without actually validating the CRC) and those, which are not CRC secured. That means, the Timesync module ignores the CRC.	

▽

△

Template Description	
The CRC is supposed to be ignored	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcIgnored	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_NOT_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are not CRC secured. All other Time Synchronization messages are ignored.	
Template Description	
The CRC is not supposed to be present. If CRC is present the message is ignored.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcNotValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_OPTIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages which are not CRC secured and Time Synchronization messages which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Template Description	
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcOptional	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSlave/CanTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
The Timesync module accepts only Time Synchronization messages, which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Template Description	
This CRC is supposed to be validated.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
CanTSynGlobalTimeSyncDataIDList	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for SYNC messages to calculate CRC.	
M2 Parameter	
CanGlobalTimeDomainProps.syncDataIDList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00024]

BSW Module	BSW Context
CanTSyn	CanTSyn/CanTSynGlobalTimeDomain/CanTSynGlobalTimeSyncDataIDList
BSW Parameter	**BSW Type**
CanTSynGlobalTimeSyncDataIDListElement	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for SYNC messages to calculate CRC.	
M2 Parameter	
CanGlobalTimeDomainProps.syncDataIDList	
Mapping Rule	**Mapping Type**
Value shall be derived from element of the ordered syncDataIDList.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTSyn_00028]

C.6 CanTp

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig	
BSW Parameter		**BSW Type**
CanTpChannel		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration parameters of the CanTp channel.		
Template Description		
Configuration parameters of the CanTp channel.		
M2 Parameter		
CanTpChannel		
Mapping Rule		**Mapping Type**
Create Container ifor each CanTpChannel that exist in ECU Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTp_00288]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		**BSW Type**
CanTpRxNSdu		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
The following parameters needs to be configured for each CAN N-SDU that the CanTp module receives via the CanTp Channel. This N-SDU produces meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16 and ADDRESS_EXTENSION_8.		
Template Description		
M2 Parameter		
CanTpConnection.tpSdu		
Mapping Rule		**Mapping Type**
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is received.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTp_00137]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpBs		ECUC-INTEGER-PARAM-DEF
BSW Description		
Sets the number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs.For further details on this parameter value see ISO 15765-2 specification.		
Template Description		

▽

△

The maximum number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification. Note: For reasons of buffer length, the CAN Transport Layer can adapt the BS value within the limit of this maximum BS	
M2 Parameter	
CanTpConnection.maxBlockSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00276]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpNAe	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "mixed".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00284]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNAe
BSW Parameter	**BSW Type**
CanTpNAe	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the transport protocol address extension value.	
Template Description	
If the mixed addressing format is used, this parameter contains the transport protocol address extension value.	
M2 Parameter	
CanTpAddress.tpAddressExtensionValue	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddressExtension.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00285]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpNSa	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with Addressing Format CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "extended".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00253]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNSa
BSW Parameter	**BSW Type**
CanTpNSa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the transport protocol source address value.	
Template Description	
An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter	
CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00254]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpNTa	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	

▽

△

M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "extended".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00139]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNTa
BSW Parameter	**BSW Type**
CanTpNTa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the transport protocol target address value.	
Template Description	
An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter	
CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00255]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpNar	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the N_Ar timeout. N_Ar is the time for transmission of a CAN frame (any N_PDU) on the receiver side.	
Template Description	
This attribute states the timeout between the PDU transmit request of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the receiver side (for FC or AF). Specified in seconds.	
M2 Parameter	
CanTpNode.timeoutAr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00277]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**

▽

△

CanTpNbr	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.	
Template Description	
Value in seconds of the performance requirement for (N_Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.	
M2 Parameter	
CanTpConnection.timeoutBr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00245]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpNcr	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.	
Template Description	
This parameter defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.	
M2 Parameter	
CanTpConnection.timeoutCr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00279]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpRxAddressingFormat	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Declares which communication addressing mode is supported for this RxNSdu. Definition of Enumeration values: CanTp Standard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_CanTp_00281]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu

BSW Parameter	BSW Type
CanTpRxNPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU consumes a meta data item of type CAN_ID_32.

Template Description

M2 Parameter

CanTpConnection.dataPdu

Mapping Rule	Mapping Type
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00256]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu

BSW Parameter	BSW Type
CanTpRxPaddingActivation	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Defines if the received frame uses padding or not. This parameter is restricted to 8 bytes N-PDUs.

Definition of enumeration values:

CanTpOn: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always >= 8 bytes in case of CAN 2.0)

CanTpOff: The N-PDU received does not use padding for SF, FC and the last CF. (N-PDU length is dynamic - any valid DLC value). Note: The mandatory mapping to the next higher valid DLC value for N-PDUs with a length > 8 bytes is not affected by this parameter.

Template Description

This specifies whether or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.

true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes)

false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)

M2 Parameter

CanTpConnection.paddingActivation

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00249]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter	**BSW Type**	
CanTpRxTaType	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Declares the communication type of this Rx N-SDU.		
Template Description		
Network Target Address type.		
M2 Parameter		
CanTpConnection.taType		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTp_ 00250]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxTaType	
BSW Parameter	**BSW Type**	
CANTP_FUNCTIONAL	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Functional request type		
Template Description		
Functional request type		
M2 Parameter		
NetworkTargetAddressType.functional		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxTaType	
BSW Parameter	**BSW Type**	
CANTP_PHYSICAL	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Physical request type		
Template Description		
Physical request type		
M2 Parameter		
NetworkTargetAddressType.physical		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpRxWftMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter indicates how many Flow Control wait N-PDUs can be consecutively transmitted by the receiver. It is local to the node and is not transmitted inside the FC protocol data unit. CanTpRxWftMax is used to avoid sender nodes being potentially hooked-up in case of a temporarily reception inability on the part of the receiver nodes, whereby the sender could be waiting continuously.	
Template Description	
This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.	
M2 Parameter	
CanTpNode.maxFcWait	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00251]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpSTmin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs. For further details on this parameter value see ISO 15765-2 specification.	
Template Description	
Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.	
M2 Parameter	
CanTpNode.stMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00252]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpTxFcNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
CanTpConnection.flowControlPdu	

▽

△

Mapping Rule	Mapping Type
Create container if the CanTpConnection contains a reference to a FlowControlNPdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00259]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel
BSW Parameter	**BSW Type**
CanTpTxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The following parameters needs to be configured for each CAN N-SDU that the CanTp module transmits via the CanTp Channel. This N-SDU consumes meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16 and ADDRESS_EXTENSION_8.	
Template Description	
M2 Parameter	
CanTpConnection.tpSdu	
Mapping Rule	**Mapping Type**
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is transmitted.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00138]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNAe	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "mixed".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00284]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNAe
BSW Parameter	**BSW Type**
CanTpNAe	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
This parameter contains the transport protocol address extension value.	
Template Description	
If the mixed addressing format is used, this parameter contains the transport protocol address extension value.	
M2 Parameter	
CanTpAddress.tpAddressExtensionValue	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddressExtension.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00285]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNSa	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with Addressing Format CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "extended".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00253]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNSa
BSW Parameter	**BSW Type**
CanTpNSa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the transport protocol source address value.	
Template Description	
An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter	
CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00254]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNTa	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	
Mapping Rule	**Mapping Type**
Create container if addressingFormat is set to "extended".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00139]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNTa
BSW Parameter	**BSW Type**
CanTpNTa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the transport protocol target address value.	
Template Description	
An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter	
CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the TpNode that contains the TpAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00255]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNas	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in second of the N_As timeout. N_As is the time for transmission of a CAN frame (any N_PDU) on the part of the sender.	
Template Description	

▽

△

This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.	
M2 Parameter	
CanTpNode.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00263]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNbs	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the N_Bs timeout. N_Bs is the time of transmission until reception of the next Flow Control N_PDU.	
Template Description	
This parameter defines the timeout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.	
M2 Parameter	
CanTpConnection.timeoutBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00264]

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNcs	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the performance requirements relating to N_Cs. CanTpNcs is the time in which CanTp is allowed to request from PduR the Tx data of a Consecutive Frame N_PDU.	
Template Description	
The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.	
M2 Parameter	
CanTpConnection.timeoutCs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00265]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpRxFcNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU consumes a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
CanTpConnection.flowControlPdu	
Mapping Rule	**Mapping Type**
Create container if the CanTpConnection contains a reference to a FlowControlNPdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00271]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTc	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Switch for enabling Transmit Cancellation.	
Template Description	
With this switch Tx Cancellation can be turned on or off. Please note that the Rx Cancellation is always enabled.	
M2 Parameter	
CanTpConnection.cancellation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00282]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxAddressingFormat	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Declares which communication addressing format is supported for this TxNSdu. Definition of Enumeration values: CanTp Standard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.	
Template Description	
Declares which communication addressing mode is supported.	
M2 Parameter	
CanTpConnection.addressingFormat	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00262]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
CanTpConnection.dataPdu	
Mapping Rule	**Mapping Type**
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00274]

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxPaddingActivation	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines if the transmit frame use padding or not. This parameter is restricted to 8 byte N-PDUs. Definition of Enumeration values: CanTpOn The transmit N-PDU uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes in case of CAN 2.0) CanTpOff The transmit N-PDU does not use padding for SF, CF and the last CF. (N-PDU length is dynamic - any valid DLC value). Note: The mandatory mapping to the next higher valid DLC value for N-PDUs with a length > 8 bytes is not affected by this parameter.	
Template Description	
This specifies whether or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload. true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes) false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)	
M2 Parameter	
CanTpConnection.paddingActivation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTp_00269]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpTxTaType		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Declares the communication type of this TxNsdu. Enumeration values: CanTpPhysical. Used for 1:1 communication. CanTpFunctional. Used for 1:n communication.		
Template Description		
Network Target Address type.		
M2 Parameter		
CanTpConnection.taType		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTp_00270]

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig	
BSW Parameter		**BSW Type**
CanTpMainFunctionPeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
Allow to configure the time for the MainFunction (as float in seconds). The CanTpMainFunctionPeriod should be assigned a value which is optimal regarding all of the timers configured for CanTp in TX and RX data transfer i.e. the differences from the configured timing should be as small as possible. Please note: This period shall be the same as call cycle time of the periodic task were CanTp Main function is called.		
Template Description		
The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.		
M2 Parameter		
CanTpEcu.cycleTimeMainFunction		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_CanTp_00240]

C.7 CanTrcv

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvBaudRate		ECUC-INTEGER-PARAM-DEF
BSW Description		
Indicates the data transfer rate in kbps.		

▽

△

Template Description	
Channels speed in bits/s.	
M2 Parameter	
CommunicationCluster.baudrate	
Mapping Rule	**Mapping Type**
CanTrcvBaudRate = SystemTemplate baudrate is in bps, so divide it by 1000 to get kbps	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00169]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnCanIdIsExtended	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Indicates whether extended or standard ID is used. TRUE = Extended Can identifier is used. FALSE = Standard Can identifier is used	
Template Description	
Defines whether pncWakeupCanId and pncWakeupCanIdMask shall be interpreted as extended or standard CAN ID.	
M2 Parameter	
CanCommunicationConnector.pncWakeupCanIdExtended	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00164]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnFrameCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
CAN ID of the Wake-up Frame (WUF).	
Template Description	
CAN Identifier used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
CanCommunicationConnector.pncWakeupCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00163]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnFrameCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID Mask for the selective activation of the transceiver. It is used to enableFrame Wake-up (WUF) on a group of IDs.	
Template Description	
Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
CanCommunicationConnector.pncWakeupCanIdMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00162]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnFrameDataMaskSpec	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).	
Template Description	
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
CanCommunicationConnector.pncWakeupDataMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00165]

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec
BSW Parameter	**BSW Type**
CanTrcvPnFrameDataMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the n byte (Byte0 = LSB) of the data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).	
Template Description	
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
CanCommunicationConnector.pncWakeupDataMask	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00166]

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec
BSW Parameter	**BSW Type**
CanTrcvPnFrameDataMaskIndex	ECUC-INTEGER-PARAM-DEF
BSW Description	
holds the position n in frame of the mask-part	
Template Description	
Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.	
M2 Parameter	
CanCommunicationConnector.pncWakeupDataMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00167]

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnFrameDlc	ECUC-INTEGER-PARAM-DEF
BSW Description	
Data Length of the Wake-up Frame (WUF).	
Template Description	
Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.	
M2 Parameter	
CanCommunicationConnector.pncWakeupDlc	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00168]

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	**BSW Type**
CanTrcvWakeupByBusUsed	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Is wake up by bus supported? If CAN transceiver hardware does not support wake up by bus value is always FALSE. If CAN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not. TRUE = Is used. FALSE = Is not used.	
Template Description	
Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
CommunicationController.wakeUpByControllerSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_CanTrcv_00148]

C.8 Cdd

BSW Module	**BSW Context**
Cdd	Cdd/CddGlobalTimeContribution/CddGlobalTimeDomain/CddGlobalTimeMaster
BSW Parameter	**BSW Type**
CddGlobalTimeTxPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents configuration of the TX period. Unit: seconds	
Template Description	
This represents the period. Unit: seconds	
M2 Parameter	
GlobalTimeMaster.syncPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Cdd_00074]

C.9 Com

BSW Module	**BSW Context**
Com	Com/ComConfig
BSW Parameter	**BSW Type**
ComGwMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Each instance of this container defines one mapping of the integrated Signal Gateway.	

▽

△

Template Description	
Arranges those signals (or SignalGroups) that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a source and a target referencing to a ISignalTriggering.	
M2 Parameter	
ISignalMapping	
Mapping Rule	**Mapping Type**
In the System Extract an explicit ISignalMapping or an implicit ISignalMapping may be defined. Explicit Mapping: Create Container for each ISignalMapping.sourceSignal where the referenced ISignalTriggering refers to an ISignal. Implicit Mapping: Ifthe ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignal Group), then create Container for each ISignal referenced by the ISignalGroup where the short Name of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00544]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping
BSW Parameter	**BSW Type**
ComGwDestination	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
Each instance of this choice container allows to define one routing destination either by reference to an already configured COM signal / group signal or by a destination description container.	
Template Description	
M2 Parameter	
ISignalMapping.targetSignal	
Mapping Rule	**Mapping Type**
Explicit Mapping: Create Container for each targetSignal reference that is defined in the ISignal Mapping. Implicit Mapping: If the ISignalMapping.targetSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignal Group), then create Container for each ISignal referenced by the ISignalGroup where the short Name of the target ISignal matches the shortName of a source ISignal of the ISignal Mapping.sourceSignal ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00546]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination
BSW Parameter	**BSW Type**
ComGwDestinationDescription	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Description of a gateway destination. This container allows defining a gateway destination without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	

▽

△

M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Informations can be derived from ISignalToIPduMapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00549]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order	
Template Description	
This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packing ByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.	
M2 Parameter	
ISignalToIPduMapping.startPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00259]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.	
Template Description	
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.	
M2 Parameter	
DataFilter	
Mapping Rule	**Mapping Type**
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionModeCondition element contains a reference to this signal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00339]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterAlgorithm	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The range of values is specified in the Clause 3.2.3 (Reception Filtering) of [3].	
Template Description	
This attribute specifies the type of the filter.	
M2 Parameter	
DataFilter.dataFilterType	
Mapping Rule	**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00146]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Mask for old and new value.	
M2 Parameter	
DataFilter.mask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00235]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to specify the upper boundary	
M2 Parameter	
DataFilter.max	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00317]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterMin	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to specify the lower boundary	
M2 Parameter	
DataFilter.min	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00318]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3]. Range = 0..(ComFilterPeriod-1)	
Template Description	
Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter	
DataFilter.offset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00313]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterPeriod	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.	
Template Description	
Specifies number of messages to occur before the message is passed again	
M2 Parameter	
DataFilter.period	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00312]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription/ComFilter
BSW Parameter	**BSW Type**
ComFilterX	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to compare with	
M2 Parameter	
DataFilter.x	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00147]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComGwIPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to an I-PDU of a Signal Gateway source or destination description.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00550]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the signal's network representation.	
Template Description	
This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.	
M2 Parameter	
ISignalToIPduMapping.packingByteOrder	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00157]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComSignalInitValue	ECUC-STRING-PARAM-DEF
BSW Description	
Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0. In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.	
Template Description	
Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.	
M2 Parameter	
`ISignal.initValue`, `NonqueuedSenderComSpec.initValue`	
Mapping Rule	**Mapping Type**
It is possible to aggregate an initValue at the level of a ComSpec in the SWC Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the init Value is defined in the System Template.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00170]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription
BSW Parameter	**BSW Type**
ComTransferProperty	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.	
Template Description	
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.	
M2 Parameter	
`ISignalToIPduMapping.transferProperty`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00232]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestinationDescription

BSW Parameter	BSW Type
ComUpdateBitPosition	ECUC-INTEGER-PARAM-DEF

BSW Description

Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side.

Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.

Template Description

The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored.

Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByte Last. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change.

This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

M2 Parameter

`ISignalToIPduMapping.updateIndicationBitPosition`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00257]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination

BSW Parameter	BSW Type
ComGwSignal	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This container allows specifying a gateway source or destination respectively with a reference to a ComSignal or a Com GroupSignal.

Template Description

M2 Parameter

`ISignalTriggering.iSignal`

Mapping Rule	Mapping Type
Explicit Mapping: Create Container if ISignal is referenced from ISignalMapping.sourceSignal or ISignalMapping.targetSignal via ISignalTriggering. Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignal Group), then create Container for each ISignal referenced by the ISignalGroup where the short Name of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup. If the ISignalMapping.targetSignal refers to an ISignal Triggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignal Group where the shortName of the target ISignal matches the shortName of a source ISignal of the ISignalMapping.sourceSignal ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00551]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwSignal
BSW Parameter	**BSW Type**
ComGwSignalRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
Reference to an object of a gateway relation. Either to a ComSignal or a ComGroupSignal.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Refers to the to be routed ComSignal or ComGroupSignal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00547]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping
BSW Parameter	**BSW Type**
ComGwSource	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
This choice container allows the definition of the gateway source signal either by reference to an already configured COM signal / group signal or by a source description container.	
Template Description	
M2 Parameter	
ISignalMapping.sourceSignal	
Mapping Rule	**Mapping Type**
Explicit Mapping: Create Container for sourceSignal reference that is defined in the ISignal Mapping. Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignal Group), then create Container for each ISignal referenced by the ISignalGroup where the short Name of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00545]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource
BSW Parameter	**BSW Type**
ComGwSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container allows specifying a gateway source or destination respectively with a reference to a ComSignal or a Com GroupSignal.	
Template Description	
M2 Parameter	



△

ISignalTriggering.iSignal	
Mapping Rule	**Mapping Type**
Explicit Mapping: Create Container if ISignal is referenced from ISignalMapping.sourceSignal or ISignalMapping.targetSignal via ISignalTriggering. Implicit Mapping: If the ISignalMapping.sourceSignal refers to an ISignalTriggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignal Group), then create Container for each ISignal referenced by the ISignalGroup where the short Name of the source ISignal matches the shortName of a destination ISignal of the ISignal Mapping.targetSignal ISignalGroup. If the ISignalMapping.targetSignal refers to an ISignal Triggering where the ISignalTriggering refers to an ISignalGroup (and no explicit mapping is defined for the ISignalGroup), then create Container for each ISignal referenced by the ISignal Group where the shortName of the target ISignal matches the shortName of a source ISignal of the ISignalMapping.sourceSignal ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00551]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSignal
BSW Parameter	**BSW Type**
ComGwSignalRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
Reference to an object of a gateway relation. Either to a ComSignal or a ComGroupSignal.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Refers to the to be routed ComSignal or ComGroupSignal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00547]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwSource
BSW Parameter	**BSW Type**
ComGwSourceDescription	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Description of a gateway source. This container allows defining a gateway source without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Informations can be derived from ISignalToIPduMapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00548]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription

BSW Parameter	BSW Type
ComBitPosition	ECUC-INTEGER-PARAM-DEF

BSW Description

Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order

Template Description

This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packing ByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array.

If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.

M2 Parameter

`ISignalToIPduMapping.startPosition`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00259]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription

BSW Parameter	BSW Type
ComBitSize	ECUC-INTEGER-PARAM-DEF

BSW Description

Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by Com SignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.

Template Description

Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals.

The ISignal length of zero bits is allowed.

M2 Parameter

`ISignal.length`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00158]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription

BSW Parameter	BSW Type
ComGwIPduRef	ECUC-REFERENCE-DEF

BSW Description

Reference to an I-PDU of a Signal Gateway source or destination description.

▽

△

Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00550]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	**BSW Type**
ComSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the signal's network representation.	
Template Description	
This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.	
M2 Parameter	
ISignalToIPduMapping.packingByteOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00157]

BSW Module	BSW Context
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	**BSW Type**
ComSignalLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored. The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.	
Template Description	
Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.	
M2 Parameter	
ISignal.length	
Mapping Rule	**Mapping Type**

▽

△

ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00437]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	**BSW Type**
ComSignalType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.	
Template Description	
With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the Port Interface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.	
M2 Parameter	
ISignal.dataTypePolicy	
Mapping Rule	**Mapping Type**
The mapping depends from the setting in the ISignal.dataTypePolicy: - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::Core Communication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetwork Representation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBase Type SWComponentTemplate::Communication::ReceiverComSpec.compositeNetwork Representation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationData Type.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_ 02079 and access SwDataDefProps.swBaseType. Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => Com SignalType should not be configured. - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if System Signal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00127]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription
BSW Parameter	**BSW Type**
ComUpdateBitPosition	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.
Template Description
The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByte Last. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.
M2 Parameter
ISignalToIPduMapping.updateIndicationBitPosition

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00257]

BSW Module	BSW Context
Com	Com/ComConfig

BSW Parameter	BSW Type
ComIPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.	
Template Description	
Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed.	
M2 Parameter	
ISignalIPdu	

Mapping Rule	Mapping Type
create container for each SignalIPdu that is transmitted by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00340]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu

BSW Parameter	BSW Type
ComIPduDirection	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.	
Template Description	

▽

△

Communication Direction of the Connector Port (input or output Port).	
M2 Parameter	
CommConnectorPort.communicationDirection	
Mapping Rule	**Mapping Type**
Find IPduTriggering of the regarded SignalIPdu. The IPduTriggering contains a reference to an IPduPort that is aggegated by the regarded ECU. If the communicationDirection of the Comm ConnectorPort is "in" than the IPdu is received.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00493]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**
ComIPduGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the I-PDU groups this I-PDU belongs to.	
Template Description	
M2 Parameter	
ISignalIPduGroup.iSignalIPdu	
Mapping Rule	**Mapping Type**
Find IPduGroup that points to this SignalIPdu and create the reference. Also consider that, in addition to direct references, an IPduGroup may reference a SignalIPdu indirectly via ISignalIPdu Group.containedISignalIPduGroup references.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00206]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**
ComIPduSignalGroupRef	ECUC-REFERENCE-DEF
BSW Description	
References to all signal groups contained in this I-Pdu	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Find ISignal in the ISignalIPdu that refers to a ISignalGroup and create reference to this Group	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00519]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**

▽

△

ComIPduSignalProcessing	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
For the definition of the two modes Immediate and Deferred.	
Template Description	
Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.	
M2 Parameter	
`IPduPort.iPduSignalProcessing`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00119]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**
ComIPduSignalRef	ECUC-REFERENCE-DEF
BSW Description	
References to all signals contained in this I-PDU.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
`ISignalToIPduMapping`	
Mapping Rule	**Mapping Type**
Find ISignal in the IPdu which refers to a SystemSignal and create reference to this Signal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00518]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**
ComIPduType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.	
Template Description	
Contains all configuration elements for AUTOSAR TP.	
M2 Parameter	
`TpConfig`	
Mapping Rule	**Mapping Type**
If this PduTriggering is referenced by a TpConnection then set this EnumerationLiteral to TP.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00761]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu
BSW Parameter	**BSW Type**
ComTxIPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.	
Template Description	
Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed.	
M2 Parameter	
ISignalIPdu	
Mapping Rule	**Mapping Type**
create container if an ISignalIPdu is transmitted by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00496]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	**BSW Type**
ComMinimumDelayTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the Minimum Delay Time (MDT) between successive transmissions of this I-PDU in seconds. The MDT is independent of the possible different transmission modes. There is only one minimum delay time parameter for one I-PDU. The minimum delay timer is not reset by changing the transmission mode. Hence, it is not allowed to violate the minimum delay time by transmission mode changes. It is not possible to monitor the minimum delay time for I-PDUs that are requested using the Com_TriggerTransmit API.	
Template Description	
Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.	
M2 Parameter	
IPduTiming.minimumDelay	
Mapping Rule	**Mapping Type**
Find IPduTiming for the transmitted IPdu and use the specified value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00181]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	**BSW Type**
ComTxIPduUnusedAreasDefault	ECUC-INTEGER-PARAM-DEF
BSW Description	
The AUTOSAR COM module fills not used areas of an I-PDU with this byte pattern. This attribute is mandatory to avoid undefined behaviour. This byte-pattern will be repeated throughout the I-PDU before any init-values or update-bits were set.	
Template Description	
AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.	
M2 Parameter	



△

ISignalIPdu.unusedBitPattern	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00017]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	**BSW Type**
ComTxModeFalse	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to false.	
Template Description	
M2 Parameter	
TransmissionModeDeclaration.transmissionModeFalseTiming	
Mapping Rule	**Mapping Type**
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00454]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse
BSW Parameter	**BSW Type**
ComTxMode	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionMode Timing element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)	
M2 Parameter	
TransmissionModeTiming	
Mapping Rule	**Mapping Type**
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00351]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The transmission mode None shall not have any further sub-attributes in the ComTxMode object.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionMode Timing element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)	
M2 Parameter	
TransmissionModeTiming	
Mapping Rule	**Mapping Type**
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlled Timing. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00137]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeNumberOfRepetitions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.	
Template Description	
Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter	
EventControlledTiming.numberOfRepetitions	
Mapping Rule	**Mapping Type**
ComTxModeNumberOfRepetitions = EventControlledTiming.numberOfRepetitions	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00281]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeRepetitionPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.	
Template Description	
M2 Parameter	
EventControlledTiming.repetitionPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00282]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimeOffset	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupStart and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected. In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.	
Template Description	
M2 Parameter	
CyclicTiming.timeOffset	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from IPdu Timing.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00180]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the repetition period in seconds of the periodic transmission requests in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
Template Description	
M2 Parameter	
CyclicTiming.timePeriod	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from IPdu Timing.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00178]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	**BSW Type**
ComTxModeTrue	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to true.	
Template Description	
M2 Parameter	
TransmissionModeDeclaration.transmissionModeTrueTiming	
Mapping Rule	**Mapping Type**
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00455]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue
BSW Parameter	**BSW Type**
ComTxMode	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.	
Template Description	
If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionMode Timing element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)	
M2 Parameter	
TransmissionModeTiming	
Mapping Rule	**Mapping Type**
Create Container if a timing specification is defined for this IPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00351]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The transmission mode None shall not have any further sub-attributes in the ComTxMode object.	
Template Description	

▽

△

If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionMode Timing element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: • Periodic (Cyclic Timing) • Direct /n-times (EventControlledTiming) • Mixed (Cyclic and EventControlledTiming are assigned) • None (no timing is assigned)	
M2 Parameter	
TransmissionModeTiming	
Mapping Rule	**Mapping Type**
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlled Timing. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00137]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeNumberOfRepetitions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.	
Template Description	
Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter	
EventControlledTiming.numberOfRepetitions	
Mapping Rule	**Mapping Type**
ComTxModeNumberOfRepetitions = EventControlledTiming.numberOfRepetitions	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00281]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeRepetitionPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than or equal to 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.	
Template Description	
M2 Parameter	
EventControlledTiming.repetitionPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00282]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimeOffset	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupStart and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected. In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.	
Template Description	
M2 Parameter	
CyclicTiming.timeOffset	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from IPdu Timing.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00180]

BSW Module	BSW Context
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the repetition period in seconds of the periodic transmission requests in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
Template Description	
M2 Parameter	
CyclicTiming.timePeriod	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from IPdu Timing.TransmissionModeDeclaration.TransmissionModeTiming element	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00178]

BSW Module	BSW Context
Com	Com/ComConfig
BSW Parameter	**BSW Type**
ComIPduGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's I-PDU groups.	

▽

△

Template Description	
The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups.	
M2 Parameter	
ISignalIPduGroup	
Mapping Rule	**Mapping Type**
Create container for each CoreCommunication::ISignalIPduGroup that is contained in the ECU Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00341]

BSW Module	BSW Context
Com	Com/ComConfig
BSW Parameter	**BSW Type**
ComSignal	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's signals.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
A ComSignal container shall be created for an ISignal that is contained in an ISignalIPdu which the Com module is sending. The creation of a ComSignal container may be omitted for an ISignal that is contained in an ISignal IPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal. Please note that more details for the upstream mapping of ComSignals are described in [TPS_SYST_01066] and [TPS_SYST_01067].	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00344]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order	
Template Description	
This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packing ByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.	

▽

△

M2 Parameter	
ISignalToIPduMapping.startPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00259]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComBitSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by Com SignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.	
Template Description	
Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.	
M2 Parameter	
ISignal.length	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00158]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComDemEventParameterRefs	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameters DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
A ComSignal container shall be created for an ISignal that is contained in an ISignalIPdu which the Com module is sending. The creation of a ComSignal container may be omitted for an ISignal that is contained in an ISignal IPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_10046]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComFilter		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.		
Template Description		
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.		
M2 Parameter		
DataFilter		
Mapping Rule		**Mapping Type**
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionModeCondition element contains a reference to this signal.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00339]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterAlgorithm		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
The range of values is specified in the Clause 3.2.3 (Reception Filtering) of [3].		
Template Description		
This attribute specifies the type of the filter.		
M2 Parameter		
DataFilter.dataFilterType		
Mapping Rule		**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00146]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMask		ECUC-INTEGER-PARAM-DEF
BSW Description		
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].		
Template Description		
Mask for old and new value.		
M2 Parameter		
DataFilter.mask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Com_00235]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMax		ECUC-INTEGER-PARAM-DEF
BSW Description		
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].		
Template Description		
Value to specify the upper boundary		
M2 Parameter		
DataFilter.max		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00317]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMin		ECUC-INTEGER-PARAM-DEF
BSW Description		
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].		
Template Description		
Value to specify the lower boundary		
M2 Parameter		
DataFilter.min		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00318]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterOffset		ECUC-INTEGER-PARAM-DEF
BSW Description		
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3]. Range = 0..(ComFilterPeriod-1)		
Template Description		
Specifies the initial number of messages to occur before the first message is passed		
M2 Parameter		
DataFilter.offset		

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00313]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterPeriod	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.	
Template Description	
Specifies number of messages to occur before the message is passed again	
M2 Parameter	
DataFilter.period	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00312]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterX	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to compare with	
M2 Parameter	
DataFilter.x	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00147]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComFirstTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.	
Template Description	

▽

△

• ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. • ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.
M2 Parameter
ISignalPort.firstTimeout

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00183]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal

BSW Parameter	**BSW Type**
ComInitialValueOnly	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the users (e.g. RTE, SwCluC). Thus the Com implementation does not need to expect any API calls for this signal (group).	
Template Description	
Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or Data Filters for ISignals need to be specified several ISignalPorts may be created.	
M2 Parameter	
ISignalPort	

Mapping Rule	**Mapping Type**
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00811]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal

BSW Parameter	**BSW Type**
ComRxDataTimeoutAction	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.	
Template Description	
This attribute controls the behavior with respect to the handling of timeouts.	
M2 Parameter	
NonqueuedReceiverComSpec.handleTimeoutType	

Mapping Rule	**Mapping Type**

▽

△

If a full DataMapping exists for the SystemSignal and there is a single receiver on this ECU then this information shall be configured in accordance with the configured NonqueuedReceiverCom Spec. If a full DataMapping exists for the SystemSignal and there are multiple receivers on this ECU then this information is available in the NonqueuedReceiverComSpecs. In this case the attribute Com RxDataTimeoutAction of the related ComSignal/ComSignalGroup shall be configured to NONE to ensure that the RTE always has access to the last received value. Please note that the SWS_RTE defines an algorithm to implement the applicable timeout action.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00412]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComSignalDataInvalidValue	ECUC-STRING-PARAM-DEF
BSW Description	
Defines the data invalid value of the signal. In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.	
Template Description	
InvalidationPolicy: Specifies whether the component can actively invalidate a particular `dataElement`. If no `invalidationPolicy` points to a `dataElement` this is considered to yield the identical result as if the `handleInvalid` attribute was set to `dontInvalidate`. **SwDataDefProps.invalidValue:** Optional value to express invalidity of the actual data element.	
M2 Parameter	
`InvalidationPolicy`, `SwDataDefProps.invalidValue`	
Mapping Rule	**Mapping Type**
ComSignalDataInvalidValue is only derived 1:1 from the SwDataDefProps.invalidValue if the InvalidationPolicy equals keep or replace. In all other cases of InvalidationPolicy the ComSignal DataInvalidValue shall not be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00391]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the signal's network representation.	
Template Description	

▽

△

This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.
M2 Parameter
ISignalToIPduMapping.packingByteOrder

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00157]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal

BSW Parameter	**BSW Type**
ComSignalInitValue	ECUC-STRING-PARAM-DEF

BSW Description
Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0. In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.
Template Description
Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.
M2 Parameter
ISignal.initValue, NonqueuedSenderComSpec.initValue

Mapping Rule	**Mapping Type**
It is possible to aggregate an initValue at the level of a ComSpec in the SWC Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the init Value is defined in the System Template.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00170]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal

BSW Parameter	**BSW Type**
ComSignalLength	ECUC-INTEGER-PARAM-DEF

BSW Description
Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored. The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.

▽

△

Template Description	
Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.	
M2 Parameter	
ISignal.length	
Mapping Rule	**Mapping Type**
ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00437]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComSignalType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.	
Template Description	
With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the Port Interface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.	
M2 Parameter	
ISignal.dataTypePolicy	
Mapping Rule	**Mapping Type**
The mapping depends from the setting in the ISignal.dataTypePolicy: - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::Core Communication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetwork Representation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBase Type SWComponentTemplate::Communication::ReceiverComSpec.compositeNetwork Representation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationData Type.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_ 02079 and access SwDataDefProps.swBaseType. Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => Com SignalType should not be configured. - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if System Signal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00127]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal

BSW Parameter	BSW Type
ComSystemTemplateSystemSignalRef	ECUC-FOREIGN-REFERENCE-DEF

BSW Description

Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.

Template Description

An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.

M2 Parameter

`ISignalToIPduMapping`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00002]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal

BSW Parameter	BSW Type
ComTimeout	ECUC-FLOAT-PARAM-DEF

BSW Description

Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.

Template Description

ISignalPort.timeout:

- ISignalPort with communicationDirection = in:

Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverCom Spec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module.

- ISignalPort with communicationDirection = out:

Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the SenderCom Spec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured SenderComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module.

This attribute can be used in the following cases:

- legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the DataMapping is missing.
- bus monitoring use cases in which the DataMapping is ignored.

TransmissionAcknowledgementRequest.timeout:
Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

NonqueuedReceiverComSpec.aliveTimeout:
Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description.

If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.

M2 Parameter

▽

△

ISignalPort.timeout, TransmissionAcknowledgementRequest.timeout, NonqueuedReceiverComSpec.aliveTimeout	
Mapping Rule	**Mapping Type**
TX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec.transmissionAcknowledge.timeout that specifies the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been transmitted according to the specified timing description. In this case the timeout value in SenderComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. RX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec.aliveTimeout. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. Please note that the SWS_RTE defines an algorithm to finally set the applicable timeout value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00263]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComTimeoutSubstitutionValue	ECUC-STRING-PARAM-DEF

BSW Description

The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE.

- In case of UINT8_N, the default value is a string of length ComSignalLength with all bytes set to 0x00.
- In case of UINT8_DYN, the initial size shall be 0.
- In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64, the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification.
- In case the ComSignalType is FLOAT32, FLOAT64, the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification.
- In case the ComSignalType is BOOLEAN, the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification.
- In case the ComSignal is a UINT8_N, UINT8_DYN, the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN, the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.

Template Description

Defines and enables the ComTimeoutSubstituition for this ISignal.

M2 Parameter

ISignal.timeoutSubstitutionValue

Mapping Rule	**Mapping Type**
The mapping of ComTimeoutSubstitutionValue depends on the setting in the ISignal.dataType Policy: - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::Core Communication::ISignal.timeoutSubstitutionValue - ISignal.dataTypePolicy = networkRepresentationFromComSpec: SWComponent Template::Communication::NonequeuedReceiverComSpec.timeoutSubstitutionValue - ISignal.dataTypePolicy = transformingISignal this is not supported.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_10006]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComTransferProperty		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
Template Description		
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.		
M2 Parameter		
ISignalToIPduMapping.transferProperty		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00232]

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComUpdateBitPosition		ECUC-INTEGER-PARAM-DEF
BSW Description		
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.		
Template Description		
The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByte Last. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.		
M2 Parameter		
ISignalToIPduMapping.updateIndicationBitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Com_00257]

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComSignalGroup		ECUC-PARAM-CONF-CONTAINER-DEF

▽

△

BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's signal groups.	
Template Description	
SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)	
M2 Parameter	
ISignalGroup	
Mapping Rule	**Mapping Type**
A ComSignalGroup container shall be created for an ISignalGroup that is contained in an ISignal IPdu which the Com module is sending. The creation of a ComSignalGroup container may be omitted for an ISignalGroup that is contained in an ISignalIPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal Group.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00345]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComDemEventParameterRefs	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_SetEventStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameters DemEventId symbolic value. The standardized errors are provided in this container and can be extended by vendor-specific error references.	
Template Description	
An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
A ComSignal container shall be created for an ISignal that is contained in an ISignalIPdu which the Com module is sending. The creation of a ComSignal container may be omitted for an ISignal that is contained in an ISignal IPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_10046]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComFirstTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.	
Template Description	

▽

△

• ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. • ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.
M2 Parameter
ISignalPort.firstTimeout

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00183]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup

BSW Parameter	**BSW Type**
ComGroupSignal	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.
Template Description
Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different Signal IPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup.
M2 Parameter
ISignal

Mapping Rule	**Mapping Type**
A ComGroupSignal container shall be created for an ISignal contained in an ISignalGroup that is contained in an ISignalIPdu which the Com module is sending. The creation of a ComGroupSignal container may be omitted for an ISignal contained in an ISignal Group that is contained in an ISignalIPdu which the Com module is receiving if no Rx ISignalPort exists for the ISignal of the ISignalGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00520]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal

BSW Parameter	**BSW Type**
ComBitPosition	ECUC-INTEGER-PARAM-DEF

BSW Description
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order
Template Description

▽

△

This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packing ByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.	
M2 Parameter	
ISignalToIPduMapping.startPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00259]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**
ComBitSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by Com SignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.	
Template Description	
Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.	
M2 Parameter	
ISignal.length	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00158]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**
ComFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.	
Template Description	
Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.	
M2 Parameter	
DataFilter	
Mapping Rule	**Mapping Type**

▽

△

Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionModeCondition element contains a reference to this signal.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00339]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterAlgorithm	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The range of values is specified in the Clause 3.2.3 (Reception Filtering) of [3].	
Template Description	
This attribute specifies the type of the filter.	
M2 Parameter	
DataFilter.dataFilterType	
Mapping Rule	**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00146]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Mask for old and new value.	
M2 Parameter	
DataFilter.mask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00235]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to specify the upper boundary	

▽

△

M2 Parameter	
DataFilter.max	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00317]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterMin	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to specify the lower boundary	
M2 Parameter	
DataFilter.min	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00318]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3]. Range = 0..(ComFilterPeriod-1)	
Template Description	
Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter	
DataFilter.offset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00313]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterPeriod	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.	

▽

△

Template Description	
Specifies number of messages to occur before the message is passed again	
M2 Parameter	
DataFilter.period	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00312]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter
BSW Parameter	**BSW Type**
ComFilterX	ECUC-INTEGER-PARAM-DEF
BSW Description	
The name of this attribute corresponds to the parameter name in the Clause 3.2.3 (Reception filtering) of [3].	
Template Description	
Value to compare with	
M2 Parameter	
DataFilter.x	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00147]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**
ComGroupSignalTransferProperty	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Optionally defines whether this group signal shall contribute to the TRIGGERED_ON_CHANGE transfer property of the signal group. If at least one group signal of a signal group has the "ComGroupSignalTransferProperty" configured all other group signals of that signal group shall have the attribute configured as well.	
Template Description	
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping.transferProperty	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00560]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**

▽

△

ComSignalDataInvalidValue	ECUC-STRING-PARAM-DEF
BSW Description	
Defines the data invalid value of the signal. In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.	
Template Description	
InvalidationPolicy: Specifies whether the component can actively invalidate a particular `dataElement`. If no `invalidationPolicy` points to a `dataElement` this is considered to yield the identical result as if the `handleInvalid` attribute was set to `dontInvalidate`. **SwDataDefProps.invalidValue:** Optional value to express invalidity of the actual data element.	
M2 Parameter	
`InvalidationPolicy`, `SwDataDefProps.invalidValue`	
Mapping Rule	**Mapping Type**
ComSignalDataInvalidValue is only derived 1:1 from the SwDataDefProps.invalidValue if the InvalidationPolicy equals keep or replace. In all other cases of InvalidationPolicy the ComSignal DataInvalidValue shall not be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00391]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**
ComSignalEndianness	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the endianness of the signal's network representation.	
Template Description	
This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.	
M2 Parameter	
`ISignalToIPduMapping.packingByteOrder`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00157]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal

BSW Parameter	BSW Type
ComSignalInitValue	ECUC-STRING-PARAM-DEF

BSW Description

Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0.

In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.

Template Description

Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.

M2 Parameter

`ISignal.initValue`, `NonqueuedSenderComSpec.initValue`

Mapping Rule	Mapping Type
It is possible to aggregate an initValue at the level of a ComSpec in the SWC Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the init Value is defined in the System Template.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00170]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal

BSW Parameter	BSW Type
ComSignalLength	ECUC-INTEGER-PARAM-DEF

BSW Description

Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored.

The supported maximum length is restricted by the used transportation system. For non TP-PDUs the maximum size of a PDU, and therefore also of any included signal, is limited by the concrete bus characteristic. For example, the limit is 8 bytes for CAN and LIN, 64 bytes for CAN FD and 254 for FlexRay.

Template Description

Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals.

The ISignal length of zero bits is allowed.

M2 Parameter

`ISignal.length`

Mapping Rule	Mapping Type
ComSignalLength = ISignal.length / 8 (i.e. value of baseTypeSize)	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00437]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal

BSW Parameter	BSW Type
ComSignalType	ECUC-ENUMERATION-PARAM-DEF

BSW Description

The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.

Template Description

With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks.

If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the Port Interface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.

M2 Parameter

`ISignal.dataTypePolicy`

Mapping Rule	Mapping Type
The mapping depends from the setting in the ISignal.dataTypePolicy: - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::Core Communication::ISignal.networkRepresentationProps.swBaseType - ISignal.dataTypePolicy = networkRepresentationFromComSpec: if defined on ComSpec: SWComponentTemplate::Communication::SenderComSpec.networkRepresentation.swBaseType SWComponentTemplate::Communication::SenderComSpec.compositeNetwork Representation.networkRepresentation.swBaseType SWComponentTemplate::Communication::ReceiverComSpec.networkRepresentation.swBase Type SWComponentTemplate::Communication::ReceiverComSpec.compositeNetwork Representation.networkRepresentation.swBaseType if not defined on ComSpec: CommonStructure::ImplementationDataTypes::ImplementationData Type.swDataDefProps.swBaseType Find the ImplementationDataType according to TPS_SYST_ 02079 and access SwDataDefProps.swBaseType. Consequence: If two SenderReceiverToSignalMappings that point to the same SystemSignal result in incompatible BaseTypes (see constr_1220 in SoftwareComponentTemplate) => Com SignalType should not be configured. - ISignal.dataTypePolicy = portInterfaceDefinition --> option has atpStatus "removed", in consequence no mapping is available." - ISignal.dataTypePolicy = transformingISignal Hardcoded to UINT8_N or UINT8_DYN. Datatype can be derived from SystemSignal.dynamicLength: UINT8_N should be used if System Signal.dynamicLength = false UINT8_DYN should be used if SystemSignal.dynamicLength = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00127]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal

BSW Parameter	BSW Type
ComSystemTemplateSystemSignalRef	ECUC-FOREIGN-REFERENCE-DEF

BSW Description

Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.

Template Description

An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.

▽

△

M2 Parameter	
ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00002]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal
BSW Parameter	**BSW Type**
ComTimeoutSubstitutionValue	ECUC-STRING-PARAM-DEF
BSW Description	

The signal substitution value will be used in case of a timeout and ComRxDataTimeoutAction is set to SUBSTITUTE.

- In case of UINT8_N, the default value is a string of length ComSignalLength with all bytes set to 0x00.
- In case of UINT8_DYN, the initial size shall be 0.
- In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64, the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification.
- In case the ComSignalType is FLOAT32, FLOAT64, the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification.
- In case the ComSignalType is BOOLEAN, the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification.
- In case the ComSignal is a UINT8_N, UINT8_DYN, the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN, the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpreted as 0-sized dynamic signal.

Template Description	
Defines and enables the ComTimeoutSubstituition for this ISignal.	
M2 Parameter	
ISignal.timeoutSubstitutionValue	
Mapping Rule	**Mapping Type**
The mapping of ComTimeoutSubstitutionValue depends on the setting in the ISignal.dataType Policy: - ISignal.dataTypePolicy = override or legacy: SystemTemplate::Fibex::FibexCore::Core Communication::ISignal.timeoutSubstitutionValue - ISignal.dataTypePolicy = networkRepresentationFromComSpec: SWComponent Template::Communication::NonequeuedReceiverComSpec.timeoutSubstitutionValue - ISignal.dataTypePolicy = transformingISignal this is not supported.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_10006]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComInitialValueOnly	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the users (e.g. RTE, SwCluC). Thus the Com implementation does not need to expect any API calls for this signal (group).

Template Description

▽

△

Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or Data Filters for ISignals need to be specified several ISignalPorts may be created.	
M2 Parameter	
ISignalPort	
Mapping Rule	**Mapping Type**
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00811]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComRxDataTimeoutAction	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.	
Template Description	
This attribute controls the behavior with respect to the handling of timeouts.	
M2 Parameter	
NonqueuedReceiverComSpec.handleTimeoutType	
Mapping Rule	**Mapping Type**
If a full DataMapping exists for the SystemSignal and there is a single receiver on this ECU then this information shall be configured in accordance with the configured NonqueuedReceiverCom Spec. If a full DataMapping exists for the SystemSignal and there are multiple receivers on this ECU then this information is available in the NonqueuedReceiverComSpecs. In this case the attribute Com RxDataTimeoutAction of the related ComSignal/ComSignalGroup shall be configured to NONE to ensure that the RTE always has access to the last received value. Please note that the SWS_RTE defines an algorithm to implement the applicable timeout action.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00412]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComSignalGroupArrayAccess	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether the uint8-array based access shall be used for this ComSignalGroup.	
Template Description	
M2 Parameter	
ISignalGroup.comBasedSignalGroupTransformation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_10003]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup

BSW Parameter	BSW Type
ComSystemTemplateSignalGroupRef	ECUC-FOREIGN-REFERENCE-DEF

BSW Description

Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this Com SignalGroup represents.

Template Description

An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.

M2 Parameter

`ISignalToIPduMapping`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00001]

BSW Module	BSW Context
Com	Com/ComConfig/ComSignalGroup

BSW Parameter	BSW Type
ComTimeout	ECUC-FLOAT-PARAM-DEF

BSW Description

Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.

Template Description

ISignalPort.timeout:

- ISignalPort with communicationDirection = in:

Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverCom Spec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module.

- ISignalPort with communicationDirection = out:

Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the SenderCom Spec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured SenderComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module.

This attribute can be used in the following cases:

- legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the DataMapping is missing.
- bus monitoring use cases in which the DataMapping is ignored.

TransmissionAcknowledgementRequest.timeout:
Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

NonqueuedReceiverComSpec.aliveTimeout:
Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description.

If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.

M2 Parameter

▽

△

ISignalPort.timeout, TransmissionAcknowledgementRequest.timeout, NonqueuedReceiverComSpec.aliveTimeout	
Mapping Rule	**Mapping Type**
TX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec.transmissionAcknowledge.timeout that specifies the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been transmitted according to the specified timing description. In this case the timeout value in SenderComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. RX Signals: If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec.aliveTimeout. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template defined on the ISignalPort. Please note that the SWS_RTE defines an algorithm to finally set the applicable timeout value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00263]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComTransferProperty	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.	
Template Description	
Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.	
M2 Parameter	
ISignalToIPduMapping.transferProperty	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00232]

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComUpdateBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN, 0..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.	
Template Description	

▽

△

The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByte Last. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.
M2 Parameter
ISignalToIPduMapping.updateIndicationBitPosition

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00257]

BSW Module	**BSW Context**
Com	Com/ComGeneral

BSW Parameter	**BSW Type**
ComEnableMDTForCyclicTransmission	ECUC-BOOLEAN-PARAM-DEF

BSW Description
Enables globally for the whole Com module the minimum delay time monitoring for cyclic and repeated transmissions (Com TxModeMode=PERIODIC or ComTxModeMode=MIXED for the cyclic transmissions, ComTxModeNumberOfRepetitions > 0 for repeated transmissions).
Template Description
Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions > 0).
M2 Parameter
EcuInstance.comEnableMDTForCyclicTransmission

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Com_00788]

C.10 ComM

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet

BSW Parameter	**BSW Type**
ComMChannel	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container contains the configuration (parameters) of the bus channel(s). The channel parameters shall be harmonized within the whole communication stack.
Template Description

▽

△

M2 Parameter	
PhysicalChannel.commConnector	
Mapping Rule	**Mapping Type**
• Can, Lin, Fr: For each CommunicationCluster the EcuInstance is connected to, one ComMChannel container is created. • For Ethernet: For each EthernetPhysicalChannel the EcuInstance is connected to, one ComMChannel container is created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00565]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**
ComMBusType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Identifies the bus type of the channel.	
Template Description	
The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels.	
M2 Parameter	
CommunicationCluster	
Mapping Rule	**Mapping Type**
Depends of the used CommunicationCluster subclass: abstractCanCluster --> COMM_BUS_TYPE_CAN FlexRayCluster --> COMM_BUS_TYPE_FR EthernetCluster --> COMM_BUS_TYPE_ETH LinCluster --> COMM_BUS_TYPE_LIN UserDefinedCluster --> COMM_BUS_TYPE_CDD	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00567]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**
ComMDynamicPncToChannelMappingEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**

▽

△

1:1 Mapping. If M2 Parameter not defined then do not create ComMDynamicPncToChannel MappingEnabled	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_ 00896]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**
ComMManageReference	ECUC-REFERENCE-DEF
BSW Description	
Represents the reference between a ComMChannel with role managing channel and a ComMChannel with role managed channel.	
Template Description	
Reference between a channel with role managing channel and a channel with role managed channel.	
M2 Parameter	
PhysicalChannel.managedPhysicalChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_ 00893]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement
BSW Parameter	**BSW Type**
ComMNmLightTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left. The range shall be greater than 0.0 and less or equal to 255.0.	
Template Description	
CommunicationCluster.comMLightTimeout: Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine. **EthernetPhysicalChannel.comMLightTimeout:** Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left in the ComM channel state machine.	
M2 Parameter	
CommunicationCluster.comMLightTimeout, EthernetPhysicalChannel.comMLightTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_ 00606]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement
BSW Parameter	**BSW Type**
ComMNmVariant	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the functionality of the networkmanagement. Shall be harmonized with NM configuration.	
Template Description	
Defines the functionality of Network Management.	
M2 Parameter	
CommunicationConnector.nmVariant	
Mapping Rule	**Mapping Type**
SLAVE_PASSIVE -> if CommunicationConnector.nmVariant is set to slavePassive SLAVE-ACTIVE -> if CommunicationConnector.nmVariant is set to slaveActive FULL -> if CommunicationConnector.nmVariant is set to full PASSIVE -> if CommunicationConnector.nmVariant is set to passive LIGHT -> if CommunicationConnector.nmVariant is set to light NONE -> if CommunicationConnector.nmVariant is set to none	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00568]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement
BSW Parameter	**BSW Type**
ComMPncNmRequest	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If this parameter equals true, then Nm shall be requested again by calling Nm_NetworkRequest under either the following conditions: • every time a FULL Communication is requested due to a change in the PNC state machine to COMM_PNC_REQUESTED • if a shutdown for a PNC coincides with a PNC request of the same PNC	
Template Description	
Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.	
M2 Parameter	
EcuInstance.pncNmRequest	
Mapping Rule	**Mapping Type**
If EcuInstance.pncNmRequest is set to true, then ComM shall set ComMPncNmRequest to TRUE for every ComMChannel which has ComMNmVariant set to FULL	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00886]

BSW Module	BSW Context
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**

▽

△

ComMPncGatewayType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Identifies the Partial Network Gateway behaviour of a ComMChannel.	
Template Description	
Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".	
M2 Parameter	
CommunicationConnector.pncGatewayType	
Mapping Rule	**Mapping Type**
1:1 mapping none or not defined --> do not create ECUC Parameter	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00842]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**
ComMUserPerChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains a list of identifiers that are needed to refer to a user in the system which is linked to a channel.	
Template Description	
ComManagementMapping.comManagementGroup: IPduGroup participating in a Mode Management PortGroup. **ComManagementMapping.physicalChannel:** This reference maps the Mode Management PortGroup partial network to communication channels.	
M2 Parameter	
ComManagementMapping.comManagementGroup, ComManagementMapping.physicalChannel	
Mapping Rule	**Mapping Type**
The ComMUser that need to be referenced shall be derived from a superset of PhysicalChannels that are reachable by ComManagementMapping.comManagementGroup and ComManagement Mapping.physicalChannel. From the comManagementGroup reference all ISignalIPduGroups can be retrieved to which the ComManagementMapping refers to. From the ISignalIPduGroup all ISignalIPdus shall be collected that are contained in the ISignalIPduGroup or one of the sub ISignalIPduGroups. The search for all PduTriggerings associated with these ISignalIPdus provides a set of PhysicalChannels since the PduTriggerings are directly aggregated by a PhysicalChannel. In addition to the PhysicalChannels that are retrieved from the ComManagementMapping.comManagementGroup the directly referenced ComManagementMapping.physicalChannel shall be added. Further mappings may be required from an ECU integration point of view.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00657]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet/ComMChannel/ComMUserPerChannel
BSW Parameter	**BSW Type**
ComMUserChannel	ECUC-REFERENCE-DEF
BSW Description	

▽

△

Reference to the ComMUser that corresponds to this channel user. ImplementationType: COMM_UserHandleType	
Template Description	
Mode Management PortGroup to be mapped onto a communication channel. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems.	
M2 Parameter	
ComManagementMapping.comManagementPortGroup	
Mapping Rule	**Mapping Type**
The ComMUser reference shall be derived from the ComMgrUserNeeds which are referenced by the ComManagementMapping. Further mappings may be required from an ECU integration point of view.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00658]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet/ComMChannel
BSW Parameter	**BSW Type**
ComMWakeupSleepRequestEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Used for communication channels where the corresponding hardware support wake-up and/or sleep request capability on the network, e.g. OA TC10 compatible PHYs for Ethernet.	
Template Description	
EcuInstance.wakeUpOverBusSupported: Driver support for wakeup over Bus. **EcuInstance.ethSwitchPortGroupDerivation:** Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. **EthernetCommunicationController:** Ethernet specific communication port attributes.	
M2 Parameter	
EcuInstance.wakeUpOverBusSupported, EcuInstance.ethSwitchPortGroupDerivation, EthernetCommunicationController	
Mapping Rule	**Mapping Type**
If EcuInstance.wakeUpOverBusSupported is defined and set to TRUE and the aggregated EthernetCommunicatonController aggregate an CouplingPort with physicalLayerType set to "100 Base-T1" and "1000Base-T1",respectively, or the aggregated CouplingPort is partof a Coupling Element with couplingType set to "switch" and EcuInstance.ethSwitchPortGroupDeriviation is not defined or set to FALSE and the affected CouplingPorts have a physicalLayerType set to "100 Base-T1" and "1000Base-T1", respectively, than the corresponding ComM channel shall set Com MWakeupSleepRequestEnabled to TRUE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00898]

BSW Module	**BSW Context**
ComM	ComM/ComMConfigSet
BSW Parameter	**BSW Type**
ComMPnc	ECUC-PARAM-CONF-CONTAINER-DEF

▽

△

BSW Description	
This container contains the configuration of the partial network cluster (PNC).	
Template Description	
Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.	
M2 Parameter	
PncMapping	
Mapping Rule	**Mapping Type**
Create ComMPnc container for each PncMapping element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00843]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMChannelPerPnc		ECUC-REFERENCE-DEF
BSW Description		
Reference to the ComMChannel that is required for this PNC. ImplementationType: NetworkHandleType		
Template Description		
PncMapping.pncGroup: IPduGroup participating in a Partial Network Cluster. This reference is optional in case an ecu extract has only indirect pnc access, i.e. ecu is not directly connected to a network which supports partial network. **PncMapping.physicalChannel:** This reference maps the partial network to a communication channel. **PncMapping.pncPdurGroup:** This reference maps the Partial Network Cluster to a set of PdurIpduGroups.		
M2 Parameter		
PncMapping.pncGroup, PncMapping.physicalChannel, PncMapping.pncPdurGroup		
Mapping Rule		**Mapping Type**
The ComMChannels that need to be referenced shall be derived from the PhysicalChannels that are either reachable by PncMapping.pncGroup, PncMapping.pncPdurGroup, or by Pnc Mapping.physicalChannel. From the pncGroup reference all ISignalIPduGroups can be retrieved to which the PncMapping refers. From the ISignalIPduGroup all ISignalIPdus shall be collected that are contained in the ISignalIPduGroup or one of the sub ISignalIPduGroups. The search for all PduTriggerings associated with these ISignalIPdus provides a set of PhysicalChannels since the PduTriggerings are directly aggregated by a PhysicalChannel. From the pncPdurGroup reference all PdurIPduGroups can be retrieved to which the PncMapping refers to. From the PdurIPduGroup all PduTriggerings associated with these PdurIPduGroups provides a set of PhysicalChannels since the PduTriggerings are directly aggregated by a Physical Channel. In addition to the PhysicalChannels that are retrieved from the PncMapping.pncGroup and Pnc Mapping.pncPdurGroup the directly referenced PncMapping.physicalChannels shall be added. Please note that the PncMapping.physicalChannel reference was introduced in Release 4.4.0 and for backward compatibility reasons nobody is forced to configure this new reference. Therefore the old approach via the PncMapping.pncGroup and PncMapping.pncPdurGroup shall still be respected.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_ComM_00880]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMPncEthIfSwitchPortGroupRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the PortGroups that correspond to this PNC. Note: This is only for documentation.		
Template Description		
Reference to the partial networks this CouplingPort participates in.		
M2 Parameter		
CouplingPort.pncMapping		
Mapping Rule		**Mapping Type**
The references are derived from the reference CouplingPort to PncMapping.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_ComM_ 00891]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMPncId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Partial network cluster identification number.		
Template Description		
Identifer of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu.		
M2 Parameter		
PncMapping.pncIdentifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_ComM_ 00874]

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMPncWakeupSleepRequestEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Used for PNCs where a requested PNC shall report an active communication request towards the BswM. The BswM forward the active communication request to the lower layer communication channels where the used hardware support wake-up and/ or sleep request capability on the network, e.g. OA TC10 compatible PHYs for Ethernet. This is used e.g. for Ethernet Switch port group switching.		
Template Description		

▽

△

EcuInstance.wakeUpOverBusSupported: Driver support for wakeup over Bus. **EcuInstance.ethSwitchPortGroupDerivation:** Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. **EthernetCommunicationController:** Ethernet specific communication port attributes. **PncMapping:** Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.
M2 Parameter
EcuInstance.wakeUpOverBusSupported, EcuInstance.ethSwitchPortGroupDerivation, EthernetCommunicationController, PncMapping

Mapping Rule	**Mapping Type**
If EcuInstance.wakeUpOverBusSupported is defined and set to TRUE, the EcuInstance.ethSwitch PortGroupDeriviation is set to TRUE and the aggregated EthernetCommunicatonController aggregate a CouplingElement with couplingType set to “switch” and the aggregated CouplingPorts has set the with physicalLayerType to “100Base-T1” and “1000Base-T1”, respectively, then all derived ComMPnc shall set ComMPncWakeupSleepRequestEnabled to TRUE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_ 00899]

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral
BSW Parameter	**BSW Type**
ComMDynamicPncToChannelMappingSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	

Mapping Rule	**Mapping Type**
If at least one dynamicPncToChannelMappingEnabled attribute is defined and if at least one CommunicationConnector of the EcuInstance has dynamicPncToChannelMappingEnabled set to true, then ComMDynamicPncToChannelMappingSupport shall be set to true. Otherwise Com MDynamicPncToChannelMappingSupport shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_ 00895]

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral
BSW Parameter	**BSW Type**
ComMPncGatewayEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Enables or disables support of Partial Network Gateway. False: Partial Networking Gateway is disabled True: Partial Networking Gateway is enabled
Template Description
Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".
M2 Parameter
CommunicationConnector.pncGatewayType

Mapping Rule	**Mapping Type**
If at least one pncGatewayType attribute is defined, then ComMPncGatewayEnabled shall be set to true, if at least one CommunicationConnector of the EcuInstance has the pncGatewayType set to either active or passive. If all pncGatewayType attributes are set to none or are not defined, the value shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00887]

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral

BSW Parameter	**BSW Type**
ComMPncPrepareSleepTimer	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds the PNC state machine shall wait in COMM_PNC_PREPARE_SLEEP.	
Template Description	
Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.	
M2 Parameter	
EcuInstance.pncPrepareSleepTimer	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_ComM_00841]

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral

BSW Parameter	**BSW Type**
ComMPncSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of partial networking. False: Partial Networking is disabled True: Partial Networking is enabled	
Template Description	
Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.	
M2 Parameter	
PncMapping	

Mapping Rule	**Mapping Type**
If at least one Pnc is configured this parameter shall be set to true. Otherwise false.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_ComM_00839]

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMSynchronousWakeUp		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Wake up of one channel shall lead to a wake up of all channels if true. true: Enabled false: Disabled		
Template Description		
If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.		
M2 Parameter		
EcuInstance.channelSynchronousWakeup		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_ComM_00695]

C.11 Csm

BSW Module	BSW Context	
Csm	Csm/CsmJobs/CsmJob	
BSW Parameter		**BSW Type**
CsmJobKeyRef		ECUC-REFERENCE-DEF
BSW Description		
This parameter refers to the key which shall be used for the CsmPrimitive. It's possible to use a CsmKey for different jobs		
Template Description		
This meta-class has the ability to represent a crypto key.		
M2 Parameter		
CryptoServiceKey		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Csm_00126]

BSW Module	BSW Context	
Csm	Csm/CsmJobs/CsmJob	
BSW Parameter		**BSW Type**
CsmJobPrimitiveRef		ECUC-REFERENCE-DEF

▽

△

BSW Description	
This parameter refers to the used CsmPrimitive. Different jobs may refer to one CsmPrimitive. The referred CsmPrimitive provides detailed information on the actual cryptographic routine.	
Template Description	
This reference identifies the applicable crypto primitive for the authentication.	
M2 Parameter	
SecOcCryptoServiceMapping.authentication	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Csm_00122]

BSW Module	BSW Context
Csm	Csm/CsmKeys
BSW Parameter	**BSW Type**
CsmKey	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container for configuration of a CSM key. The container name serves as a symbolic name for the identifier of a key configuration.	
Template Description	
This meta-class has the ability to represent a crypto key.	
M2 Parameter	
CryptoServiceKey	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Csm_00014]

C.12 Dcm

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection
BSW Parameter	**BSW Type**
DcmDslMainConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration for a main connection of a diagnostic protocol. Additionally it may contain references to ROE and Periodic connections if the protocol type or protocol transmission type needs them.	
Template Description	
DiagnosticConncection that is used to describe the relationship between several TP connections.	
M2 Parameter	
DiagnosticConnection	
Mapping Rule	**Mapping Type**

▽

△

A DcmDslMainConnection subcontainer is created for a DiagnosticConnection that refers via functionRequest, physicalRequest or response to one or more TpConnectionIdent elements that are in turn contained in TpConnection elements via which the EcuInstancethat is included in the EcuExtract sends or receives PDUs.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_00706]

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection

BSW Parameter	**BSW Type**
DcmDslProtocolRxTesterSourceAddr	ECUC-INTEGER-PARAM-DEF

BSW Description
Tester source address uniquely describes a client and will be used e.g within the jump to Bootloader interfaces. This parameter is not required for generic connections (DcmPdus with MetaDataLength >= 1).
Template Description
TpConnection Base Class.
M2 Parameter
`TpConnection`

Mapping Rule	**Mapping Type**
DcmDslProtocolRxTesterSourceAddr shall be derived from the TpConnection that is referenced from the DiagnosticConnection from which the DcmDslMainConnection container was derived. The TpConnection points in the role transmitter and receiver to TpNodes that in turn point to a Tp Address. The DcmDslProtocolRxTesterSourceAddr value shall be set to the diagnostic address (Tp Address) of the remote node. This means that if the TpConnection.transmitter does not represent the configured EcuInstance then the DcmDslProtocolRxTesterSourceAddr value can be derived from the TpNode that is referenced in the transmitter role and in turn references the TpAddress. If the TpConnection.transmitter represents the configured EcuInstance then the DcmDslProtocolRx TesterSourceAddr value can only be retrived if the TpConnection sends its Pdus to a physical address. In this case the remote node is the TpNode referenced in the TpConnection.receiver role. The tester address can be derived from the TpAddress that is referenced by the TpNode if only one receiver is configured. In case that several TpNode elements are referenced in the Tp Conncetion.receiver role this parameter can not be derived from the System Description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01115]

BSW Module	**BSW Context**
Dcm	Dcm/DcmGeneral/DcmSecureCoding

BSW Parameter	**BSW Type**
DcmSecureCodingValidationRoutine	ECUC-REFERENCE-DEF

BSW Description
Refers to a routine control that triggers the secure coding validation and data update process.
Template Description
Refer to a routine control that is used to authenticate and persist the secure coding data.
M2 Parameter
`DiagnosticSecureCodingMapping.validation`

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dcm_01248]

C.13 Dds

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsAppDataRxPduCollection
BSW Parameter	**BSW Type**
DdsAppDataRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The upper layer PDU used to send received data from DDS to Application.	
Template Description	
DdsCpISignalToDdsTopicMapping: Mapping of an ISignal to a DdsTopic. **DdsServiceInstanceEventCp:** This element represents an event as part of the DDS Service Instance. **DdsServiceInstanceOperationCp:** This element represents an operation as part of the DDS Service Instance.	
M2 Parameter	
DdsCpISignalToDdsTopicMapping, DdsServiceInstanceEventCp, DdsServiceInstanceOperationCp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00135]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsAppDataTxPduCollection
BSW Parameter	**BSW Type**
DdsAppDataTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The upper layer PDU used to transmit data from Application to DDS itself.	
Template Description	
DdsCpISignalToDdsTopicMapping: Mapping of an ISignal to a DdsTopic. **DdsServiceInstanceEventCp:** This element represents an event as part of the DDS Service Instance. **DdsServiceInstanceOperationCp:** This element represents an operation as part of the DDS Service Instance.	
M2 Parameter	
DdsCpISignalToDdsTopicMapping, DdsServiceInstanceEventCp, DdsServiceInstanceOperationCp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00132]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainDpMulticastRtpsPduCollection/DdsRtpsDpMulticastDataRxPdu
BSW Parameter	**BSW Type**
DdsDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

The ID of the Domain to which this DDS node belongs. It unambiguously identifies the DDS Domain to which the Domain Participant belongs. Note: Only entities that belong to the same DDS Domain can communicate with each other.	
Template Description	
Definition of the DDS Domain Id.	
M2 Parameter	
DdsCpDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00138]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainDpMulticastRtpsPduCollection/DdsRtpsDpMulticastDataTxPdu
BSW Parameter	**BSW Type**
DdsDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ID of the Domain to which this DDS node belongs. It unambiguously identifies the DDS Domain to which the Domain Participant belongs. Note: Only entities that belong to the same DDS Domain can communicate with each other.	
Template Description	
Definition of the DDS Domain Id.	
M2 Parameter	
DdsCpDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00138]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainMulticastRtpsPduCollection/DdsRtpsMulticastDataRxPdu
BSW Parameter	**BSW Type**
DdsDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ID of the Domain to which this DDS node belongs. It unambiguously identifies the DDS Domain to which the Domain Participant belongs. Note: Only entities that belong to the same DDS Domain can communicate with each other.	
Template Description	
Definition of the DDS Domain Id.	
M2 Parameter	
DdsCpDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00138]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainMulticastRtpsPduCollection/DdsRtpsMulticastDataTxPdu
BSW Parameter	**BSW Type**
DdsDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ID of the Domain to which this DDS node belongs. It unambiguously identifies the DDS Domain to which the Domain Participant belongs. Note: Only entities that belong to the same DDS Domain can communicate with each other.	
Template Description	
Definition of the DDS Domain Id.	
M2 Parameter	
DdsCpDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00138]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection
BSW Parameter	**BSW Type**
DdsDomainParticipant	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of one single Domain Participant hosted within the current node. One node can contain more than one Domain Participant.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00012]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsConsumedServiceInstance	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of consumed service instance at Ecuc level in a concrete implementation on top of DDS.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dds_00219]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance

BSW Parameter	BSW Type
DdsDiscoveryType	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Supported discovery schemes for DDS Service Instances.

Template Description
See "HISTORY" chapter of DDS.

M2 Parameter
DdsHistory.historyKind

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00197]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance

BSW Parameter	BSW Type
DdsServiceEventReceiver	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container represents an Event receiver relying on DdsDataWriter linked to a VariableDataPrototype in a SenderReceiver interface.

Template Description
Definition of a DDS Domain.

M2 Parameter
DdsCpDomain

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00221]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance

BSW Parameter	BSW Type
DdsServiceFieldsFieldClient	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This object collects the configuration needed to handle all the Fields actions on client side.

Template Description
Definition of a DDS Domain.

▽

△

M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00222]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance/DdsServiceFieldsFieldClient
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldClientGetter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects the configuration needed to handle all the Fields Getter action on client side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00224]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance/DdsServiceFieldsFieldClient
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldClientSetter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects the configuration needed to handle all the Fields Setter action on client side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00225]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance/DdsServiceFieldsFieldClient
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldNotifierReceiver	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This object collects the configuration needed to handle all the Fields Notify action on client side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00223]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance
BSW Parameter	**BSW Type**
DdsServiceMethodsFireAndForgetMethod	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects all the configuration needed to handle Methods relying on the Fire And Forget communication paradigms. Role (e.g server v.s side) is defined in the parent object.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00217]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsConsumedService Instance
BSW Parameter	**BSW Type**
DdsServiceMethodsRequestReplyMethod	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects all the configuration needed to handle Methods relying on the Remote Procedure Call communication paradigms. Role (e.g server v.s side) is defined in the parent object.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00215]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsDataReceiver	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container useful to usability: it defines a DDS DataReader linked to a Sender/ReceiverInterface.	
Template Description	
Mapping of an ISignal to a DdsTopic.	
M2 Parameter	
DdsCpISignalToDdsTopicMapping	
Mapping Rule	**Mapping Type**
Collection of DdsDataReaders used for signal based communication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00159]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsDataSender	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container useful to usability: it defines a DDS DataWriter linked to a Sender/ReceiverInterface.	
Template Description	
Mapping of an ISignal to a DdsTopic.	
M2 Parameter	
DdsCpISignalToDdsTopicMapping	
Mapping Rule	**Mapping Type**
Collection of DdsDataWriters used for signal based communication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00157]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The ID of the Domain to which this DDS node belongs. It unambiguously identifies the DDS Domain to which the Domain Participant belongs. Note: Only entities that belong to the same DDS Domain can communicate with each other.	
Template Description	
Definition of the DDS Domain Id.	
M2 Parameter	
DdsCpDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00138]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant	
BSW Parameter	**BSW Type**	
DdsDomainParticipantQoS	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container represents the configuration of QoS supported by the Dds DomainParticipant.		
Template Description		
Definition of a DDS QOS Profile.		
M2 Parameter		
DdsCpQosProfile		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00013]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant	
BSW Parameter	**BSW Type**	
DdsParticipantLeaseDuration	ECUC-FLOAT-PARAM-DEF	
BSW Description		
The periodic rate to send SPDP announcement message. Time given in seconds.		
Template Description		
See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds.		
M2 Parameter		
DdsDurabilityService.durabilityServiceCleanupDelay		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00228]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant	
BSW Parameter	**BSW Type**	
DdsProvidedServiceInstance	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container represents the configuration of a provided service instance at Ecuc level in a concrete implementation on top of DDS.		
Template Description		
Definition of a DDS Domain.		
M2 Parameter		
DdsCpDomain		
Mapping Rule		**Mapping Type**
1:1 mapping		full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dds_00193]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance

BSW Parameter	BSW Type
DdsDiscoveryType	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Supported discovery schemes for DDS Service Instances.

Template Description
See "HISTORY" chapter of DDS.

M2 Parameter
DdsHistory.historyKind

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00197]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance

BSW Parameter	BSW Type
DdsServiceEventSender	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container represents an Event sender relying on DdsDataWriter linked to a VariableDataPrototype in a SenderReceiver Interface.

Template Description
Definition of a DDS Domain.

M2 Parameter
DdsCpDomain

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00205]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance

BSW Parameter	BSW Type
DdsServiceFieldsFieldServer	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This object collects the configuration needed to handle the Field actions on server side.

Template Description
Definition of a DDS Domain.

▽

△

M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00208]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance/DdsServiceFieldsFieldServer
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldNotifierSender	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects configuration needed to handle all the Fields Notify action on server side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00209]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance/DdsServiceFieldsFieldServer
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldServerGetter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects the configuration needed to handle all the Fields Getter action on server side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00210]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance/DdsServiceFieldsFieldServer
BSW Parameter	**BSW Type**
DdsServiceFieldsFieldServerSetter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This object collects the configuration needed to handle all the Fields Setter action on server side.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00214]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance
BSW Parameter	**BSW Type**
DdsServiceMethodsFireAndForgetMethod	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects all the configuration needed to handle Methods relying on the Fire And Forget communication paradigms. Role (e.g server v.s side) is defined in the parent object.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00217]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsProvidedService Instance
BSW Parameter	**BSW Type**
DdsServiceMethodsRequestReplyMethod	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This object collects all the configuration needed to handle Methods relying on the Remote Procedure Call communication paradigms. Role (e.g server v.s side) is defined in the parent object.	
Template Description	
Definition of a DDS Domain.	
M2 Parameter	
DdsCpDomain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00215]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsPublisher	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of one Publisher.	
Template Description	
DdsCpDomain: Definition of a DDS Domain. **DdsCpPartition:** Definition of a DDS Partition.	
M2 Parameter	
DdsCpDomain, DdsCpPartition	
Mapping Rule	**Mapping Type**
DdsPublisher are dependent on DdsCpDomain and DdsCpPartition, but independent of DdsCp Topic.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00016]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher
BSW Parameter	**BSW Type**
DdsDataWriter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of one data writer. One publisher can refer to one or more writer, but a writer can belong to one single publisher.	
Template Description	
DdsCpTopic: Definition of a DDS Topic. **DdsCpDomain:** Definition of a DDS Domain. **DdsCpPartition:** Definition of a DDS Partition.	
M2 Parameter	
DdsCpTopic, DdsCpDomain, DdsCpPartition	
Mapping Rule	**Mapping Type**
DdsDataWriter are dependent on DdsCpDomain, DdsCpPartition, and DdsCpTopic.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00023]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsData Writer
BSW Parameter	**BSW Type**
DdsDataWriterQoS	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of QoS Profiles related to the current DdsDataWriter.	

▽

△

Template Description	
Definition of a DDS QOS Profile.	
M2 Parameter	
DdsCpQosProfile	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00028]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDeadline	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DEADLINE QoS policy.	
Template Description	
Describes the DDS DEADLINE QoS policy.	
M2 Parameter	
DdsDeadline	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00039]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDeadline
BSW Parameter	**BSW Type**
DdsDeadlinePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DEADLINE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DEADLINE" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsDeadline.deadlinePeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00040]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDestinationOrder	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DESTINATION_ORDER QoS policy.	
Template Description	
Describes the DDS DESTINATION_ORDER QoS policy.	
M2 Parameter	
DdsDestinationOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00057]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDestinationOrder
BSW Parameter	**BSW Type**
DdsDestinationOrderKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DESTINATION_ORDER" chapter of DDS [32].	
Template Description	
See "DESTINATION_ORDER" chapter of DDS.	
M2 Parameter	
DdsDestinationOrder.destinationOrderKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00058]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDurability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DURABILITY QoS policy.	
Template Description	
Describes the DDS DURABILITY QoS policy.	
M2 Parameter	
DdsDurability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00034]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurability
BSW Parameter	**BSW Type**
DdsDurabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DURABILITY" chapter of DDS [32].	
Template Description	
See "DURABILITY" chapter in DDS.	
M2 Parameter	
DdsDurability.durabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00035]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDurabilityService	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DURABILITY_SERVICE QoS policy.	
Template Description	
Describes the DDS DURABILITY_SERVICE QoS policy.	
M2 Parameter	
DdsDurabilityService	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00036]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceCleanupDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds.	
M2 Parameter	
DdsDurabilityService.durabilityServiceCleanupDelay	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00037]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceHistoryDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00119]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00038]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	

▽

△

See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00121]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00120]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00122]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**

▽

△

DdsHistory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] HISTORY QoS policy.	
Template Description	
Describes the DDS HISTORY QoS policy.	
M2 Parameter	
DdsHistory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00059]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00060]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryOrderDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyOrderDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00063]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsLatencyBudget	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LATENCY_BUDGET QoS policy.	
Template Description	
Describes the DDS LATENCY_BUDGET QoS policy.	
M2 Parameter	
DdsLatencyBudget	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00041]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsLatencyBudget
BSW Parameter	**BSW Type**
DdsLatencyBudgetDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LATENCY_BUDGET" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LATENCY_BUDGET" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLatencyBudget.latencyBudgetDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00042]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsLifespan	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIFESPAN QoS policy.	
Template Description	
Describes the DDS LIFESPAN QoS policy.	
M2 Parameter	
DdsLifespan	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00055]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsLifespan
BSW Parameter	**BSW Type**
DdsLifespanDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIFESPAN" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIFESPAN" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLifespan.lifespanDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00056]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsLiveliness	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIVELINESS QoS policy.	
Template Description	
Describes the DDS LIVELINESS QoS policy.	
M2 Parameter	
DdsLiveliness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00047]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivelinessLeaseDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

See "LIVELINESS" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIVELINESS" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLiveliness.livelinessLeaseDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00049]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivenessKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32].	
Template Description	
See "LIVELINESS" chapter of DDS.	
M2 Parameter	
DdsLiveliness.livenessKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00048]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsOwnership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP QoS policy.	
Template Description	
Describes the DDS OWNERSHIP QoS policy.	
M2 Parameter	
DdsOwnership	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00043]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsOwnership
BSW Parameter	**BSW Type**
DdsOwnershipKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "OWNERSHIP" chapter of DDS [32].	
Template Description	
See "OWNERSHIP" chapter of DDS.	
M2 Parameter	
DdsOwnership.ownershipKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00044]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsOwnershipStrength	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP_STRENGTH QoS policy.	
Template Description	
Describes the DDS OWNERSHIP_STRENGTH QoS policy.	
M2 Parameter	
DdsOwnershipStrength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00045]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsOwnershipStrength
BSW Parameter	**BSW Type**
DdsOwnershipStrengthValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "OWNERSHIP_STRENGTH" chapter of DDS [32].	
Template Description	
See "OWNERSHIP_STRENGTH" chapter of DDS.	
M2 Parameter	
DdsOwnershipStrength.ownershipStrength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00046]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsReliability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RELIABILITY QoS policy.	
Template Description	
Describes the DDS RELIABILITY QoS policy.	
M2 Parameter	
DdsReliability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00050]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32].	
Template Description	
See "RELIABILITY" chapter of DDS.	
M2 Parameter	
DdsReliability.reliabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00051]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityMaxBlockingTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "RELIABILITY" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsReliability.reliabilityMaxBlockingTime	
Mapping Rule	**Mapping Type**

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00052]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsResourceLimits	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RESOURCE_LIMITS QoS policy.	
Template Description	
Describes the DDS RESOURCE_LIMITS QoS policy.	
M2 Parameter	
DdsResourceLimits	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00061]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00064]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	

▽

△

M2 Parameter	
DdsResourceLimits.maxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00062]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00065]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsTransportPriority	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] TRANSPORT_PRIORITY QoS policy.	
Template Description	
Describes the DDS TRANSPORT_PRIORITY QoS policy.	
M2 Parameter	
DdsTransportPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00053]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsPublisher/DdsDataWriter/DdsDataWriterQoS/DdsTransportPriority
BSW Parameter	**BSW Type**
DdsTransportPriorityValue	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

See "TRANSPORT_PRIORITY" chapter of DDS [32].	
Template Description	
See "TRANSPORT_PRIORITY" chapter of DDS.	
M2 Parameter	
DdsTransportPriority.transportPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00054]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsRemoteDomainParticipantCollection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Collection of Remote Domain Participants.	
Template Description	
DdsCpDomain: Definition of a DDS Domain. **DdsCpTopic:** Definition of a DDS Topic. **DdsCpPartition:** Definition of a DDS Partition.	
M2 Parameter	
DdsCpDomain, DdsCpTopic, DdsCpPartition	
Mapping Rule	**Mapping Type**
Remote domain participant definitions are needed only for communication over the same DdsCp Topic and over the same DdsCpDomain/DdsCpPartition.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00182]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader
BSW Parameter	**BSW Type**
DdsDataReaderQoS	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of QoS Profiles related to the current DdsDataReader.	
Template Description	
Definition of a DDS QOS Profile.	
M2 Parameter	
DdsCpQosProfile	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00079]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsDeadline	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DEADLINE QoS policy.	
Template Description	
Describes the DDS DEADLINE QoS policy.	
M2 Parameter	
DdsDeadline	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00039]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsDeadline
BSW Parameter	**BSW Type**
DdsDeadlinePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DEADLINE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DEADLINE" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsDeadline.deadlinePeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00040]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsDestinationOrder	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DESTINATION_ORDER QoS policy.	
Template Description	
Describes the DDS DESTINATION_ORDER QoS policy.	
M2 Parameter	
DdsDestinationOrder	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00057]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/DdsDestinationOrder
BSW Parameter	**BSW Type**
DdsDestinationOrderKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DESTINATION_ORDER" chapter of DDS [32].	
Template Description	
See "DESTINATION_ORDER" chapter of DDS.	
M2 Parameter	
DdsDestinationOrder.destinationOrderKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00058]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsHistory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] HISTORY QoS policy.	
Template Description	
Describes the DDS HISTORY QoS policy.	
M2 Parameter	
DdsHistory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00059]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	

▽

△

See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00060]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryOrderDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyOrderDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00063]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsLatencyBudget	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LATENCY_BUDGET QoS policy.	
Template Description	
Describes the DDS LATENCY_BUDGET QoS policy.	
M2 Parameter	
DdsLatencyBudget	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00041]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsLatencyBudget
BSW Parameter	**BSW Type**
DdsLatencyBudgetDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LATENCY_BUDGET" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LATENCY_BUDGET" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLatencyBudget.latencyBudgetDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00042]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsLiveliness	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIVELINESS QoS policy.	
Template Description	
Describes the DDS LIVELINESS QoS policy.	
M2 Parameter	
DdsLiveliness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00047]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivelinessLeaseDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32]. Time given in seconds.	
Template Description	



△

See "LIVELINESS" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLiveliness.livelinessLeaseDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00049]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivenessKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32].	
Template Description	
See "LIVELINESS" chapter of DDS.	
M2 Parameter	
DdsLiveliness.livenessKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00048]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsOwnership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP QoS policy.	
Template Description	
Describes the DDS OWNERSHIP QoS policy.	
M2 Parameter	
DdsOwnership	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00043]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsOwnership
BSW Parameter	**BSW Type**
DdsOwnershipKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "OWNERSHIP" chapter of DDS [32].	
Template Description	
See "OWNERSHIP" chapter of DDS.	
M2 Parameter	
DdsOwnership.ownershipKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00044]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsReliability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RELIABILITY QoS policy.	
Template Description	
Describes the DDS RELIABILITY QoS policy.	
M2 Parameter	
DdsReliability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00050]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32].	
Template Description	
See "RELIABILITY" chapter of DDS.	
M2 Parameter	
DdsReliability.reliabilityKind	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00051]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityMaxBlockingTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "RELIABILITY" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsReliability.reliabilityMaxBlockingTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00052]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsResourceLimits	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RESOURCE_LIMITS QoS policy.	
Template Description	
Describes the DDS RESOURCE_LIMITS QoS policy.	
M2 Parameter	
DdsResourceLimits	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00061]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxInstances	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00064]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00062]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataReader/DdsDataReaderQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00065]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter
BSW Parameter	**BSW Type**
DdsDataWriterQoS	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of QoS Profiles related to the current DdsDataWriter.	
Template Description	
Definition of a DDS QOS Profile.	
M2 Parameter	
DdsCpQosProfile	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00028]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDeadline	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DEADLINE QoS policy.	
Template Description	
Describes the DDS DEADLINE QoS policy.	
M2 Parameter	
DdsDeadline	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00039]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDeadline
BSW Parameter	**BSW Type**
DdsDeadlinePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DEADLINE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DEADLINE" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsDeadline.deadlinePeriod	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00040]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDestinationOrder	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DESTINATION_ORDER QoS policy.	
Template Description	
Describes the DDS DESTINATION_ORDER QoS policy.	
M2 Parameter	
DdsDestinationOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00057]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDestinationOrder
BSW Parameter	**BSW Type**
DdsDestinationOrderKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DESTINATION_ORDER" chapter of DDS [32].	
Template Description	
See "DESTINATION_ORDER" chapter of DDS.	
M2 Parameter	
DdsDestinationOrder.destinationOrderKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00058]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDurability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DURABILITY QoS policy.	

▽

△

Template Description	
Describes the DDS DURABILITY QoS policy.	
M2 Parameter	
DdsDurability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00034]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDurability
BSW Parameter	**BSW Type**
DdsDurabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DURABILITY" chapter of DDS [32].	
Template Description	
See "DURABILITY" chapter in DDS.	
M2 Parameter	
DdsDurability.durabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00035]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsDurabilityService	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DURABILITY_SERVICE QoS policy.	
Template Description	
Describes the DDS DURABILITY_SERVICE QoS policy.	
M2 Parameter	
DdsDurabilityService	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00036]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceCleanupDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds.	
M2 Parameter	
DdsDurabilityService.durabilityServiceCleanupDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00037]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceHistoryDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00119]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomainParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	

▽

△

M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00038]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00121]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00120]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsDurabilityService

▽

△

BSW Parameter	BSW Type
DdsDurabilityServiceMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00122]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsHistory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] HISTORY QoS policy.	
Template Description	
Describes the DDS HISTORY QoS policy.	
M2 Parameter	
DdsHistory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00059]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00060]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsHistory

BSW Parameter	BSW Type
DdsHistoryOrderDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyOrderDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00063]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS

BSW Parameter	BSW Type
DdsLatencyBudget	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LATENCY_BUDGET QoS policy.	
Template Description	
Describes the DDS LATENCY_BUDGET QoS policy.	
M2 Parameter	
DdsLatencyBudget	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00041]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsLatencyBudget

BSW Parameter	BSW Type
DdsLatencyBudgetDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LATENCY_BUDGET" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LATENCY_BUDGET" chapter of DDS. Time given in seconds.	
M2 Parameter	



△

DdsLatencyBudget.latencyBudgetDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00042]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsLifespan	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIFESPAN QoS policy.	
Template Description	
Describes the DDS LIFESPAN QoS policy.	
M2 Parameter	
DdsLifespan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00055]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsLifespan
BSW Parameter	**BSW Type**
DdsLifespanDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIFESPAN" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIFESPAN" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLifespan.lifespanDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00056]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**

▽

△

DdsLiveliness	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIVELINESS QoS policy.	
Template Description	
Describes the DDS LIVELINESS QoS policy.	
M2 Parameter	
DdsLiveliness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00047]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivelinessLeaseDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIVELINESS" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLiveliness.livelinessLeaseDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00049]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivenessKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32].	
Template Description	
See "LIVELINESS" chapter of DDS.	
M2 Parameter	
DdsLiveliness.livenessKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00048]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsOwnership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP QoS policy.	
Template Description	
Describes the DDS OWNERSHIP QoS policy.	
M2 Parameter	
DdsOwnership	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00043]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsOwnership
BSW Parameter	**BSW Type**
DdsOwnershipKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "OWNERSHIP" chapter of DDS [32].	
Template Description	
See "OWNERSHIP" chapter of DDS.	
M2 Parameter	
DdsOwnership.ownershipKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00044]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsOwnershipStrength	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP_STRENGTH QoS policy.	
Template Description	
Describes the DDS OWNERSHIP_STRENGTH QoS policy.	
M2 Parameter	
DdsOwnershipStrength	
Mapping Rule	**Mapping Type**
1:1 mapping	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dds_00045]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsOwnershipStrength	
BSW Parameter		**BSW Type**
DdsOwnershipStrengthValue		ECUC-INTEGER-PARAM-DEF
BSW Description		
See "OWNERSHIP_STRENGTH" chapter of DDS [32].		
Template Description		
See "OWNERSHIP_STRENGTH" chapter of DDS.		
M2 Parameter		
DdsOwnershipStrength.ownershipStrength		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00046]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS	
BSW Parameter		**BSW Type**
DdsReliability		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes the DDS [32] RELIABILITY QoS policy.		
Template Description		
Describes the DDS RELIABILITY QoS policy.		
M2 Parameter		
DdsReliability		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00050]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsReliability	
BSW Parameter		**BSW Type**
DdsReliabilityKind		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
See "RELIABILITY" chapter of DDS [32].		
Template Description		

▽

△

See "RELIABILITY" chapter of DDS.	
M2 Parameter	
DdsReliability.reliabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00051]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityMaxBlockingTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "RELIABILITY" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsReliability.reliabilityMaxBlockingTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00052]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsResourceLimits	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RESOURCE_LIMITS QoS policy.	
Template Description	
Describes the DDS RESOURCE_LIMITS QoS policy.	
M2 Parameter	
DdsResourceLimits	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00061]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00064]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00062]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamplesPerInstance	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00065]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS
BSW Parameter	**BSW Type**
DdsTransportPriority	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] TRANSPORT_PRIORITY QoS policy.	
Template Description	
Describes the DDS TRANSPORT_PRIORITY QoS policy.	
M2 Parameter	
DdsTransportPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00053]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsRemoteDomain ParticipantCollection/DdsRemoteDomainParticipant/DdsRemoteDataWriter/DdsDataWriterQoS/ DdsTransportPriority
BSW Parameter	**BSW Type**
DdsTransportPriorityValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "TRANSPORT_PRIORITY" chapter of DDS [32].	
Template Description	
See "TRANSPORT_PRIORITY" chapter of DDS.	
M2 Parameter	
DdsTransportPriority.transportPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00054]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsSpdpResendPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

How long a Participant should be considered alive every time an announcement is received from the Participant. If a Participant fails to send another announcement within this time period, the Participant can be considered gone. Time given in seconds.	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds.	
M2 Parameter	
DdsDurabilityService.durabilityServiceCleanupDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00229]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant
BSW Parameter	**BSW Type**
DdsSubscriber	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of a Subscriber.	
Template Description	
DdsCpPartition: Definition of a DDS Partition. **DdsCpDomain:** Definition of a DDS Domain.	
M2 Parameter	
DdsCpPartition, DdsCpDomain	
Mapping Rule	**Mapping Type**
DdsPublisher are dependent on DdsCpDomain and DdsCpPartition, but independent of DdsCp Topic.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00017]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber
BSW Parameter	**BSW Type**
DdsDataReader	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of one data reader. One subscriber can refer to one or more readers, but a reader can belong to one single subscriber.	
Template Description	
DdsCpTopic: Definition of a DDS Topic. **DdsCpDomain:** Definition of a DDS Domain. **DdsCpPartition:** Definition of a DDS Partition.	
M2 Parameter	

▽

△

DdsCpTopic, DdsCpDomain, DdsCpPartition	
Mapping Rule	**Mapping Type**
DdsDataReader are dependent on DdsCpDomain, DdsCpPartition, and DdsCpTopic.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00075]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader
BSW Parameter	**BSW Type**
DdsDataReaderQoS	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of QoS Profiles related to the current DdsDataReader.	
Template Description	
Definition of a DDS QOS Profile.	
M2 Parameter	
DdsCpQosProfile	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00079]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsDeadline	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DEADLINE QoS policy.	
Template Description	
Describes the DDS DEADLINE QoS policy.	
M2 Parameter	
DdsDeadline	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00039]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsDeadline
BSW Parameter	**BSW Type**
DdsDeadlinePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

See "DEADLINE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DEADLINE" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsDeadline.deadlinePeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00040]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsDestinationOrder	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DESTINATION_ORDER QoS policy.	
Template Description	
Describes the DDS DESTINATION_ORDER QoS policy.	
M2 Parameter	
DdsDestinationOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00057]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsDestinationOrder
BSW Parameter	**BSW Type**
DdsDestinationOrderKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DESTINATION_ORDER" chapter of DDS [32].	
Template Description	
See "DESTINATION_ORDER" chapter of DDS.	
M2 Parameter	
DdsDestinationOrder.destinationOrderKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00058]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsHistory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] HISTORY QoS policy.	
Template Description	
Describes the DDS HISTORY QoS policy.	
M2 Parameter	
DdsHistory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00059]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00060]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryOrderDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyOrderDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00063]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsLatencyBudget	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LATENCY_BUDGET QoS policy.	
Template Description	
Describes the DDS LATENCY_BUDGET QoS policy.	
M2 Parameter	
DdsLatencyBudget	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00041]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsLatencyBudget
BSW Parameter	**BSW Type**
DdsLatencyBudgetDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LATENCY_BUDGET" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LATENCY_BUDGET" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLatencyBudget.latencyBudgetDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00042]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsLiveliness	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIVELINESS QoS policy.	
Template Description	
Describes the DDS LIVELINESS QoS policy.	
M2 Parameter	
DdsLiveliness	
Mapping Rule	**Mapping Type**

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00047]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivelinessLeaseDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIVELINESS" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLiveliness.livelinessLeaseDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00049]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivenessKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32].	
Template Description	
See "LIVELINESS" chapter of DDS.	
M2 Parameter	
DdsLiveliness.livenessKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00048]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsOwnership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP QoS policy.	

▽

△

Template Description	
Describes the DDS OWNERSHIP QoS policy.	
M2 Parameter	
DdsOwnership	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00043]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsOwnership
BSW Parameter	**BSW Type**
DdsOwnershipKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "OWNERSHIP" chapter of DDS [32].	
Template Description	
See "OWNERSHIP" chapter of DDS.	
M2 Parameter	
DdsOwnership.ownershipKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00044]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsReliability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RELIABILITY QoS policy.	
Template Description	
Describes the DDS RELIABILITY QoS policy.	
M2 Parameter	
DdsReliability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00050]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsReliability

▽

△

BSW Parameter	BSW Type
DdsReliabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32].	
Template Description	
See "RELIABILITY" chapter of DDS.	
M2 Parameter	
DdsReliability.reliabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00051]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityMaxBlockingTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "RELIABILITY" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsReliability.reliabilityMaxBlockingTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00052]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS
BSW Parameter	**BSW Type**
DdsResourceLimits	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RESOURCE_LIMITS QoS policy.	
Template Description	
Describes the DDS RESOURCE_LIMITS QoS policy.	
M2 Parameter	
DdsResourceLimits	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00061]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00064]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00062]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsSubscriber/DdsDataReader/DdsDataReaderQoS/DdsResourceLimits
BSW Parameter	**BSW Type**
DdsResouceLimitsMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "RESOURCE_LIMITS" chapter of DDS [32].	
Template Description	
See "RESOURCE_LIMITS" chapter of DDS.	
M2 Parameter	
DdsResourceLimits.maxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00065]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant	
BSW Parameter	**BSW Type**	
DdsTopic	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container represents the configuration of one Topic.		
Template Description		
Definition of a DDS Topic.		
M2 Parameter		
DdsCpTopic		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00018]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic	
BSW Parameter	**BSW Type**	
DdsTopicName	ECUC-STRING-PARAM-DEF	
BSW Description		
Identifies name of the Topic. Communication between publishers and subscribers is based on the topic name.		
Template Description		
Definition of the DDS Topic Name.		
M2 Parameter		
DdsCpTopic.topicName		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00103]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic	
BSW Parameter	**BSW Type**	
DdsTopicQoS	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container contains the configuration of the QoS supported by the DdsTopic		
Template Description		
Definition of a DDS QOS Profile.		
M2 Parameter		
DdsCpQosProfile		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00102]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsDeadline	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DEADLINE QoS policy.	
Template Description	
Describes the DDS DEADLINE QoS policy.	
M2 Parameter	
DdsDeadline	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00039]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDeadline
BSW Parameter	**BSW Type**
DdsDeadlinePeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "DEADLINE" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "DEADLINE" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsDeadline.deadlinePeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00040]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsDestinationOrder	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] DESTINATION_ORDER QoS policy.	
Template Description	
Describes the DDS DESTINATION_ORDER QoS policy.	
M2 Parameter	
DdsDestinationOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dds_00057]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDestinationOrder

BSW Parameter	BSW Type
DdsDestinationOrderKind	ECUC-ENUMERATION-PARAM-DEF

BSW Description
See "DESTINATION_ORDER" chapter of DDS [32].

Template Description
See "DESTINATION_ORDER" chapter of DDS.

M2 Parameter
DdsDestinationOrder.destinationOrderKind

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00058]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS

BSW Parameter	BSW Type
DdsDurability	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Describes the DDS [32] DURABILITY QoS policy.

Template Description
Describes the DDS DURABILITY QoS policy.

M2 Parameter
DdsDurability

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00034]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDurability

BSW Parameter	BSW Type
DdsDurabilityKind	ECUC-ENUMERATION-PARAM-DEF

BSW Description
See "DURABILITY" chapter of DDS [32].

Template Description
See "DURABILITY" chapter in DDS.

M2 Parameter

▽

△

DdsDurability.durabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00035]

BSW Module	**BSW Context**	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS	
BSW Parameter		**BSW Type**
DdsDurabilityService		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes the DDS [32] DURABILITY_SERVICE QoS policy.		
Template Description		
Describes the DDS DURABILITY_SERVICE QoS policy.		
M2 Parameter		
DdsDurabilityService		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00036]

BSW Module	**BSW Context**	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/ DdsDurabilityService	
BSW Parameter		**BSW Type**
DdsDurabilityServiceCleanupDelay		ECUC-FLOAT-PARAM-DEF
BSW Description		
See "DURABILITY_SERVICE" chapter of DDS [32]. Time given in seconds.		
Template Description		
See "DURABILITY_SERVICE" chapter in DDS. Time given in seconds.		
M2 Parameter		
DdsDurabilityService.durabilityServiceCleanupDelay		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00037]

BSW Module	**BSW Context**	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/ DdsDurabilityService	
BSW Parameter		**BSW Type**
DdsDurabilityServiceHistoryDepth		ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00119]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceHistoryKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00038]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxInstances	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxInstances	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00121]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxSamples	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamples	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00120]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsDurabilityService
BSW Parameter	**BSW Type**
DdsDurabilityServiceMaxSamplesPerInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "DURABILITY_SERVICE" chapter of DDS [32].	
Template Description	
See "DURABILITY_SERVICE" chapter in DDS.	
M2 Parameter	
DdsDurabilityService.durabilityServiceMaxSamplesPerInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00122]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsHistory	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] HISTORY QoS policy.	
Template Description	
Describes the DDS HISTORY QoS policy.	
M2 Parameter	
DdsHistory	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00059]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00060]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsHistory
BSW Parameter	**BSW Type**
DdsHistoryOrderDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
See "HISTORY" chapter of DDS [32].	
Template Description	
See "HISTORY" chapter of DDS.	
M2 Parameter	
DdsHistory.historyOrderDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00063]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsLatencyBudget	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LATENCY_BUDGET QoS policy.	
Template Description	
Describes the DDS LATENCY_BUDGET QoS policy.	
M2 Parameter	
DdsLatencyBudget	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00041]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsLatencyBudget
BSW Parameter	**BSW Type**
DdsLatencyBudgetDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LATENCY_BUDGET" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LATENCY_BUDGET" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLatencyBudget.latencyBudgetDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00042]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsLifespan	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIFESPAN QoS policy.	
Template Description	
Describes the DDS LIFESPAN QoS policy.	
M2 Parameter	
DdsLifespan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00055]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsLifespan
BSW Parameter	**BSW Type**
DdsLifespanDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIFESPAN" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIFESPAN" chapter of DDS. Time given in seconds.	
M2 Parameter	

▽

△

DdsLifespan.lifespanDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00056]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsLiveliness	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] LIVELINESS QoS policy.	
Template Description	
Describes the DDS LIVELINESS QoS policy.	
M2 Parameter	
DdsLiveliness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00047]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivelinessLeaseDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "LIVELINESS" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "LIVELINESS" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsLiveliness.livelinessLeaseDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00049]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsLiveliness
BSW Parameter	**BSW Type**
DdsLivenessKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	

▽

△

See "LIVELINESS" chapter of DDS [32].	
Template Description	
See "LIVELINESS" chapter of DDS.	
M2 Parameter	
DdsLiveliness.livenessKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00048]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**
DdsOwnership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] OWNERSHIP QoS policy.	
Template Description	
Describes the DDS OWNERSHIP QoS policy.	
M2 Parameter	
DdsOwnership	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00043]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/ DdsOwnership
BSW Parameter	**BSW Type**
DdsOwnershipKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "OWNERSHIP" chapter of DDS [32].	
Template Description	
See "OWNERSHIP" chapter of DDS.	
M2 Parameter	
DdsOwnership.ownershipKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00044]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS
BSW Parameter	**BSW Type**

▽

△

DdsReliability	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the DDS [32] RELIABILITY QoS policy.	
Template Description	
Describes the DDS RELIABILITY QoS policy.	
M2 Parameter	
DdsReliability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00050]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityKind	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32].	
Template Description	
See "RELIABILITY" chapter of DDS.	
M2 Parameter	
DdsReliability.reliabilityKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00051]

BSW Module	**BSW Context**
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsReliability
BSW Parameter	**BSW Type**
DdsReliabilityMaxBlockingTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
See "RELIABILITY" chapter of DDS [32]. Time given in seconds.	
Template Description	
See "RELIABILITY" chapter of DDS. Time given in seconds.	
M2 Parameter	
DdsReliability.reliabilityMaxBlockingTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00052]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS	
BSW Parameter	**BSW Type**	
DdsResourceLimits	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
Describes the DDS [32] RESOURCE_LIMITS QoS policy.		
Template Description		
Describes the DDS RESOURCE_LIMITS QoS policy.		
M2 Parameter		
DdsResourceLimits		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00061]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/ DdsResourceLimits	
BSW Parameter	**BSW Type**	
DdsResouceLimitsMaxInstances	ECUC-INTEGER-PARAM-DEF	
BSW Description		
See "RESOURCE_LIMITS" chapter of DDS [32].		
Template Description		
See "RESOURCE_LIMITS" chapter of DDS.		
M2 Parameter		
DdsResourceLimits.maxInstances		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00064]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/ DdsResourceLimits	
BSW Parameter	**BSW Type**	
DdsResouceLimitsMaxSamples	ECUC-INTEGER-PARAM-DEF	
BSW Description		
See "RESOURCE_LIMITS" chapter of DDS [32].		
Template Description		
See "RESOURCE_LIMITS" chapter of DDS.		
M2 Parameter		
DdsResourceLimits.maxSamples		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00062]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsResourceLimits	
BSW Parameter		**BSW Type**
DdsResouceLimitsMaxSamplesPerInstance		ECUC-INTEGER-PARAM-DEF
BSW Description		
See "RESOURCE_LIMITS" chapter of DDS [32].		
Template Description		
See "RESOURCE_LIMITS" chapter of DDS.		
M2 Parameter		
DdsResourceLimits.maxSamplesPerInstance		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00065]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS	
BSW Parameter		**BSW Type**
DdsTopicData		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes the DDS [32] TOPIC_DATA QoS policy.		
Template Description		
Describes the DDS TOPIC_DATA QoS policy.		
M2 Parameter		
DdsTopicData		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00106]

BSW Module	BSW Context	
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS	
BSW Parameter		**BSW Type**
DdsTransportPriority		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes the DDS [32] TRANSPORT_PRIORITY QoS policy.		
Template Description		
Describes the DDS TRANSPORT_PRIORITY QoS policy.		
M2 Parameter		
DdsTransportPriority		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dds_00053]

BSW Module	BSW Context
Dds	Dds/DdsConfig/DdsDomainParticipantCollection/DdsDomainParticipant/DdsTopic/DdsTopicQoS/DdsTransportPriority

BSW Parameter	BSW Type
DdsTransportPriorityValue	ECUC-INTEGER-PARAM-DEF

BSW Description
See "TRANSPORT_PRIORITY" chapter of DDS [32].

Template Description
See "TRANSPORT_PRIORITY" chapter of DDS.

M2 Parameter
DdsTransportPriority.transportPriority

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dds_00054]

C.14 Dlt

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput

BSW Parameter	BSW Type
DltLogChannel	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Contains settings for log/trace message output

Template Description
This element contains the settings for the log/trace message output for a tuple of ApplicationId and ContextId (verbose mode) or a SessionId (non-verbose mode).

M2 Parameter
DltLogChannel

Mapping Rule	Mapping Type
Create container for each DltLogChannel that is available in the Ecu description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00876]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel

BSW Parameter	BSW Type
DltLogChannelId	ECUC-STRING-PARAM-DEF

BSW Description
This is the 4 ASCII character long name of the log channel as used in the Dlt control messages as parameter name Dlt_interface

Template Description
This attribute identifies the Channel for usage within the Log And Trace protocol.

▽

△

M2 Parameter	
DltLogChannel.logChannelId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00877]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel
BSW Parameter	**BSW Type**
DltLogChannelSegmentationSupported	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Segmentation will be used if a DLT message is larger than Pdu length.	
Template Description	
If enabled, segmentation will be used if a DLT message is larger than Pdu.length referenced via DltLogChannel.txPdu Triggering.	
M2 Parameter	
DltLogChannel.segmentationSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00916]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel
BSW Parameter	**BSW Type**
DltLogChannelThreshold	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
LogLevel Threshold	
Template Description	
This attribute allows to set a log level Threshold for Log Level filtering.	
M2 Parameter	
DltLogChannel.logTraceDefaultLogThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00878]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel
BSW Parameter	**BSW Type**
DltLogTraceStatusFlag	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Parameter to turn on/off tracing on this LogChannel completely.	

▽

△

Template Description	
This attributes defines the default trace status.	
M2 Parameter	
DltLogChannel.defaultTraceState	
Mapping Rule	**Mapping Type**
Set to true if DefaultTraceStateEnabled. Set to false if DefaultTraceStateDisabled.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00879]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannel
BSW Parameter	**BSW Type**
DltTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters of the AUTOSAR Dlt module's Tx Pdus.	
Template Description	
Reference to DltPdu that is transmitted by the DltLogChannel.	
M2 Parameter	
DltLogChannel.txPduTriggering	
Mapping Rule	**Mapping Type**
This container shall be created for the PduTriggering that is referenced from the DltLogChannel. The referenced PduTriggering shall point to a GeneralPurposeIPdu with the category DLT.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00907]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput
BSW Parameter	**BSW Type**
DltLogChannelAssignment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains a preconfiguration of ApplicationId / ContextId pairs and their assigned log channel.	
Template Description	
Reference to the Swc that produces the log or trace message. Please note that this reference shall not be set in case that the Bsw module produces the associated log or trace messages.	
M2 Parameter	
DltLogChannel.applicationContext	
Mapping Rule	**Mapping Type**
This container shall be created for each applicationContext that is referenced from the DltLog Channel that is available in the Ecu description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00887]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannelAssignment
BSW Parameter	**BSW Type**

▽

△

DltLogChannelAssignmentSwcContextRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to an ApplicationId/ContextId pair that is assigned to a DltLogChannel.	
Template Description	
Reference to the Swc that produces the log or trace message. Please note that this reference shall not be set in case that the Bsw module produces the associated log or trace messages.	
M2 Parameter	
DltLogChannel.applicationContext	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00896]

BSW Module	**BSW Context**
Dlt	Dlt/DltConfigSet/DltLogOutput/DltLogChannelAssignment
BSW Parameter	**BSW Type**
DltLogChannelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a DltLogChannel that is assigned to an ApplicationId / ContextId pair.	
Template Description	
This element contains the settings for the log/trace message output for a tuple of ApplicationId and ContextId (verbose mode) or a SessionId (non-verbose mode).	
M2 Parameter	
DltLogChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00888]

BSW Module	**BSW Context**
Dlt	Dlt/DltConfigSet/DltProtocol
BSW Parameter	**BSW Type**
DltEcuId	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
This is a choice container to choose between a EcuId value or a callout to get the EcuId.	
Template Description	
This element represents an Ecu or Machine that produces logging and tracing information.	
M2 Parameter	
DltEcu	
Mapping Rule	**Mapping Type**
Create container if DltEcu is referenced by DltConfig. Please note that only the DltEcuIdValue Choice can be derived from System Description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00860]

BSW Module	BSW Context	
Dlt	Dlt/DltConfigSet/DltProtocol/DltEcuId/DltEcuIdValueChoice	
BSW Parameter		**BSW Type**
DltEcuIdValue		ECUC-STRING-PARAM-DEF
BSW Description		
If this choice is used the EcuId shall be taken from the configured string. This is the name of the ECU for use within the Dlt protocol. If you want to use a number representation type this as character.		
Template Description		
This attribute defines the name of the ECU for use within the Dlt protocol.		
M2 Parameter		
DltEcu.ecuId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dlt_00861]

BSW Module	BSW Context	
Dlt	Dlt/DltConfigSet/DltProtocol	
BSW Parameter		**BSW Type**
DltHeaderUseEcuId		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Corresponds to field WEID (With ECU ID). If set ECU ID shall be placed in the header, else not. If DltGeneralNvRAMSupport is enabled the value of the parameter defined here is also the initial value for the corresponding NvRam entry. If DltGeneralNvRAMSupport is not set, Link-Time or Post-Build configuration shall be used.		
Template Description		
Reference to the Ecu representation in the Log And Trace Template.		
M2 Parameter		
DltConfig.dltEcu		
Mapping Rule		**Mapping Type**
Set this parameter to true if the reference is available. Set to false if reference is missing.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dlt_00811]

BSW Module	BSW Context	
Dlt	Dlt/DltConfigSet/DltProtocol	
BSW Parameter		**BSW Type**
DltHeaderUseSessionID		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Corresponds to field WSID (with Session ID). If set the Session ID shall be placed in the header, else not. If DltGeneralNvRAMSupport is enabled the value of the parameter defined here is also the initial value for the corresponding NvRam entry. If DltGeneralNvRAMSupport is not set, Link-Time or Post-Build configuration shall be used.		
Template Description		
This attribute defines whether the sessionId is used or not.		
M2 Parameter		
DltConfig.sessionIdSupport		
Mapping Rule		**Mapping Type**
1:1 mapping		full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Dlt_00813]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltProtocol
BSW Parameter	**BSW Type**
DltHeaderUseTimestamp	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Corresponds to field WTMS (With Timestamp). If set the timestamp shall be placed in the header, else not. If DltGeneralNvRAMSupport is enabled the value of the parameter defined here is also the initial value for the corresponding NvRam entry. If DltGeneralNvRAMSupport is not set, Link-Time or Post-Build configuration shall be used.	
Template Description	
This attribute defines whether a timestamp shall be added to the Dlt messages or not.	
M2 Parameter	
DltConfig.timestampSupport	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00814]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet/DltProtocol
BSW Parameter	**BSW Type**
DltUseVerboseMode	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If this flag is set to TRUE, the payload shall be transmitted in verbose mode, else the payload shall be transmitted in none-verbose mode.	
Template Description	
This attribute defines whether this channel supports non-Verbose Dlt messages. If disabled only verbose mode messages shall be used.	
M2 Parameter	
DltLogChannel.nonVerboseMode	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00911]

BSW Module	BSW Context
Dlt	Dlt/DltConfigSet
BSW Parameter	**BSW Type**
DltRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the Pdu IDs to be used for Dlt control messages reception.	
Template Description	

▽

△

GeneralPurposeIPdu: This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **DltLogChannel.rxPduTriggering:** Reference to DltPdu that is received by the DltLogChannel
M2 Parameter
GeneralPurposeIPdu, DltLogChannel.rxPduTriggering

Mapping Rule	**Mapping Type**
This container shall be created if a GeneralPurposeIPdu with the category DLT is defined in the ECU_SYSTEM_DESCRIPTION and this GeneralPurposeIPdu is received by the regarded Ecu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00900]

BSW Module	**BSW Context**
Dlt	Dlt/DltSwc

BSW Parameter	**BSW Type**
DltSwcContext	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration of ApplicationId / ContextId pairs which are supported by this SWC.	
Template Description	
This meta-class represents the Context that groups Log and Trace messages that are generated by an application.	
M2 Parameter	
DltContext	

Mapping Rule	**Mapping Type**
Create container for each DltContext that is available in the ECU Description	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00854]

BSW Module	**BSW Context**
Dlt	Dlt/DltSwc/DltSwcContext

BSW Parameter	**BSW Type**
DltSwcApplicationId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Abbreviation for the SWC (4 characters)	
Template Description	
This attribute identifies the SW-C/BSW module in the log and trace message.	
M2 Parameter	
DltApplication.applicationId	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Dlt_00858]

BSW Module	BSW Context	
Dlt	Dlt/DltSwc/DltSwcContext	
BSW Parameter		**BSW Type**
DltSwcContextId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Abbreviation for the ContextId (4 characters)		
Template Description		
This attribute is used to group log and trace messages produced by an application to distinguish functionality.		
M2 Parameter		
DltContext.contextId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Dlt_00859]

C.15 DoIP

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		**BSW Type**
DoIPChannel		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of one DoIPChannel.		
Template Description		
A connection identifies the sender and the receiver of this particular communication. The DoIp module routes a tpSdu through this connection.		
M2 Parameter		
DoIpTpConnection		
Mapping Rule		**Mapping Type**
One DoIPChannel container is created for each DoIP channel of the configured EcuInstance. A Do IP channel is constituted by the set of all DoIpTpConnection elements via which the configured EcuInstance sends or receives SDUs (SDUs are referenced by the PduTriggering DoIpTp Connection.tpSdu refers to) and that share the same local diagnosis address and tester address.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_DoIP_00069]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel	
BSW Parameter		**BSW Type**
DoIPChannelSARef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the DoIPTester.		
Template Description		
Reference to the address of the sender of the tpSdu.		

▽

△

M2 Parameter	
DoIpTpConnection.doIpSourceAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00070]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel
BSW Parameter	**BSW Type**
DoIPChannelTARef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the target address.	
Template Description	
Reference to the address of the receiver of the tpSdu.	
M2 Parameter	
DoIpTpConnection.doIpTargetAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00071]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel
BSW Parameter	**BSW Type**
DoIPPduRRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the Rx Pdus to connect with the Rx Pdus of the PduR.	
Template Description	
M2 Parameter	
DoIpTpConnection.tpSdu	
Mapping Rule	**Mapping Type**
If the DoIP channel receives an SDU, one DoIPPduRRxPdu container is created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00055]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRRxPdu
BSW Parameter	**BSW Type**
DoIPPduRRxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	

▽

△

M2 Parameter	
DoIpTpConnection.tpSdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the SDU received by the DoIP channel	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00058]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel
BSW Parameter	**BSW Type**
DoIPPduRTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the Tx Pdus to connect with the Tx Pdus of the PduR. If the parameter is not configured the channel is for functional addressing.	
Template Description	

M2 Parameter	
DoIpTpConnection.tpSdu	
Mapping Rule	**Mapping Type**
If the DoIP channel sends an SDU, one DoIPPduRTxPdu container is created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00056]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPChannel/DoIPPduRTxPdu
BSW Parameter	**BSW Type**
DoIPPduRTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	

M2 Parameter	
DoIpTpConnection.tpSdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the SDU sent by the DoIP channel	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00059]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections
BSW Parameter	**BSW Type**
DoIPTargetAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a possible TargetAddress that is supported by DoIP.	
Template Description	

▽

△

The logical DoIP address.	
M2 Parameter	
DoIpLogicAddress	
Mapping Rule	**Mapping Type**
This container shall be created for each DoIpLogicAddress referenced by a DoIpTpConnection in the role doIpTargetAddress	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00053]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTargetAddress
BSW Parameter	**BSW Type**
DoIPTargetAddressValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Valid Target Address of a DoIP target address.	
Template Description	
The logical DoIP address.	
M2 Parameter	
DoIpLogicAddress.address	
Mapping Rule	**Mapping Type**
This value shall be derived from DoIpLogicAddress that is referenced by DoIpConnection in the role doIpTargetAddress	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00054]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections
BSW Parameter	**BSW Type**
DoIPTcpConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a TCP connection to the lower layer SoAd module.	
Template Description	
Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.	
M2 Parameter	
StaticSocketConnection	
Mapping Rule	**Mapping Type**
A DoIPTcpConnection is created for each SocketConnection for which the following conditions hold: - Protocol: Tcp, i.e. the SocketConnection refers to a local SocketAddress containing an ApplicationEndpoint which in turn contains a TcpTp configuration.- DoIP PDU transmission: The EcuInstance sends and/or receives via the SocketConnection PDUs that are of Type General PurposeIPdu and have their category set to "DoIP".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00045]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection
BSW Parameter	**BSW Type**
DoIPSoAdTcpRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a Rx PDU received via SoAd over TCP	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is received via the StaticSocketConnection that represents the DoIPTcpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00080]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpRxPdu
BSW Parameter	**BSW Type**
DoIPSoAdTcpRxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu receiving by the DoIPTcpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00083]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection
BSW Parameter	**BSW Type**
DoIPSoAdTcpTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a Tx PDU sent via SoAd over TCP	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SoConIPduIdentifier.pduTriggering	
Mapping Rule	**Mapping Type**
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is transmitted via the StaticSocketConnection that represents the DoIPTcpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00081]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPTcpConnection/DoIPSoAdTcpTxPdu
BSW Parameter	**BSW Type**
DoIPSoAdTcpTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the DoIPTcpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00084]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections
BSW Parameter	**BSW Type**
DoIPUdpConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This Container describes a Udp connection to the lower layer SoAd module.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
A DoIPUdpConnection is created for each StaticSocketConnection for which the following conditions hold: • Protocol: Udp, i.e. the StaticSocketConnection is aggregated by the local SocketAddress containing an ApplicationEndpoint which in turn contains a UdpTp configuration. • DoIP PDU transmission: The EcuInstance sends and/or receives via the SocketConnection PDUs that are of Type GeneralPurposeIPdu and have their category set to "DoIP".	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00052]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection
BSW Parameter	**BSW Type**
DoIPSoAdUdpRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a Rx PDU received via SoAd over UDP.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	



△

Mapping Rule	Mapping Type
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is received via the StaticSocketConnection that represents the DoIPUdpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00046]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpRxPdu
BSW Parameter	**BSW Type**
DoIPSoAdUdpRxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu received by the DoIPUdpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00049]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection
BSW Parameter	**BSW Type**
DoIPSoAdUdpTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes a Tx PDU sent via SoAd over UDP.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Create Container for each Pdu that is of Type GeneralPurposeIPdu and has category "DoIP" and is transmitted via the StaticSocketConnection that represents the DoIPUdpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00047]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpConnection/DoIPSoAdUdpTxPdu
BSW Parameter	**BSW Type**
DoIPSoAdUdpTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	

▽

△

Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the Do IPUdpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00050]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections
BSW Parameter	**BSW Type**
DoIPUdpVehicleAnnouncementConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the UDP multicast connections to the lower layer SoAd module.	
Template Description	
Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.	
M2 Parameter	
StaticSocketConnection	
Mapping Rule	**Mapping Type**
This container shall be created: - if a StaticSocketConnection contains a single SoConIpdu Identifier that references a PduTriggering of a GeneralPurposePdu with category set to DoIp - if the GeneralPurposePdu with category set to DoIp is sent by the regarded ECU. - if the SocketAddress containing this StaticSocketConnection contains an ApplicationEndpoint with a UdpTp configuration - if the SocketAddress containing this SocketConnection references contains an ip Address that either is a the limited broadcast address (i.e., 255.255.255.255) in case of IPv4 or the link-local scope multicast address (i.e., FF02::1) in case of IPv6.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00076]

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection
BSW Parameter	**BSW Type**
DoIPSoAdUdpVehicleAnnouncementTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the vehicle announcement TxPdu sent via the SoAd.	
Template Description	
Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.	
M2 Parameter	
StaticSocketConnection	
Mapping Rule	**Mapping Type**

▽

△

This container shall be created: - if a StaticSocketConnection contains a single SoConIpdu Identifier that references a PduTriggering of a GeneralPurposePdu with category set to DoIp - if the GeneralPurposePdu with category set to DoIp is sent by the regarded ECU. - if the SocketAddress containing this SocketConnection contains an ApplicationEndpoint with a UdpTp configuration - if the SocketAddress containing this SocketConnection contains an ipAddress that either is a the limited broadcast address (i.e., 255.255.255.255) in case of IPv4 or the link- local scope multicast address (i.e., FF02::1) in case of IPv6.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00077]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPConnections/DoIPUdpVehicleAnnouncementConnection/DoIPSoAdUdpVehicleAnnouncementTxPdu
BSW Parameter	**BSW Type**
DoIPSoAdUdpVehicleAnnouncementTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the "global" PDU structure to allow harmonization of handle IDs in the COM-Stack.	
Template Description	
Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
Reference to the EcuC container that corresponds to the GeneralPurposeIPdu sent by the Do IPUdpVehicleAnnouncement connection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00079]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface
BSW Parameter	**BSW Type**
DoIPRoutingActivation	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the routing activation possibilities by representing for each container a possible routing activation request message to the DoIP entity and the according references to the activated diagnostic messages.	
Template Description	
Collection of DoIpRoutingActivation possibilities defined in the DoIpInterface.	
M2 Parameter	
DoIpInterface.doIpRoutingActivation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00030]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPRoutingActivation
BSW Parameter	**BSW Type**

▽

△

DoIPTargetAddressRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to all DoIPTargetAddress which are activated on this Routing activation.	
Template Description	
Reference to DoIPTargetAddress which is activated on this DoIpRoutingActivation.	
M2 Parameter	
DoIpRoutingActivation.doIpTargetAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00034]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface
BSW Parameter	**BSW Type**
DoIPTester	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the properties of the possible connectable Tester for the DoIP entity.	
Template Description	
A connection identifies the sender and the receiver of this particular communication. The DoIp module routes a tpSdu through this connection.	
M2 Parameter	
DoIpTpConnection	
Mapping Rule	**Mapping Type**
One DoIPTester container is created for each valid and unique tester address value of any DoIpTp Connection of the configured EcuInstance. If the configured EcuInstance receives the PDU Triggering referenced via DoIpTpConnection.tpSdu, the tester address of a DoIpTpConnection is referenced via DoIpTpConnection.doIpSourceAddress. If the configured EcuInstance sends the PDU Triggering referenced via DoIpTpConnection.tpSdu, the tester address of a DoIpTpConnection is referenced via DoIpTpConnection.doIpTargetAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00031]

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPTester
BSW Parameter	**BSW Type**
DoIPRoutingActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a DoIPRoutingActivation describing the possible routing activations of the DoIPTester	
Template Description	
Reference to a DoIPRoutingActivation describing the possible routing activations of the DoIPTester.	
M2 Parameter	
DoIpLogicTesterAddressProps.doIpTesterRoutingActivation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_DoIP_00062]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface/DoIPTester	
BSW Parameter		**BSW Type**
DoIPTesterSA		ECUC-INTEGER-PARAM-DEF
BSW Description		
Source Address of the Tester sent via routing activation or diagnostic message.		
Template Description		
The logical DoIP address.		
M2 Parameter		
DoIpLogicAddress.address		
Mapping Rule		**Mapping Type**
If the configured EcuInstance receives the PDU Triggering referenced via DoIpTpConnection.tpSdu, the tester address of a DoIpTpConnection is referenced via DoIpLogicAddress.doIpSourceAddress. If the configured EcuInstance sends the PDU Triggering referenced via DoIpTpConnection.tpSdu, the tester address of a DoIpTpConnection is referenced via DoIpLogicAddress.doIpTargetAddress.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_DoIP_00043]

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPInterface	
BSW Parameter		**BSW Type**
DoIPUseMacAddressForIdentification		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Provided the information if a configured EID at vehicle identification response/vehicle announcement is used or the MAC address. TRUE: Use MAC Address instead of EID for Vehicle identification/announcement. FALSE: Use configured EID for vehicle identification/announcement.		
Template Description		
This attribute defines whether a configured EID at vehicle identification response/vehicle announcement is used or the MAC address. TRUE: Use MAC Address instead of EID for Vehicle identification/announcement. FALSE: Use configured EID for vehicle identification/announcement.		
M2 Parameter		
DoIpInterface.useMacAddressForIdentification		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_DoIP_00013]

C.16 EcuC

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/MetaDataType/MetaDataItem	
BSW Parameter		**BSW Type**
MetaDataItemType		ECUC-ENUMERATION-PARAM-DEF
BSW Description		

▽

△

This parameter defines the type of a meta data item.	
Template Description	
This aggregation contributes the specification of the concrete meta-data item type.	
M2 Parameter	
MetaDataItem.metaDataItemType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00076]

BSW Module	**BSW Context**
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu
BSW Parameter	**BSW Type**
DynamicLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].	
Template Description	
This attribute defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].	
M2 Parameter	
Pdu.hasDynamicLength	
Mapping Rule	**Mapping Type**
Attribute can be derived from Pdu.hasDynamicLength attribute that is only relevant for UserDefined Pdus, UserDefinedIPdus, J1939DcmIPdus.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00078]

BSW Module	**BSW Context**
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu
BSW Parameter	**BSW Type**
J1939Requestable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pdu can be triggered by the J1939 request message.	
Template Description	
CanFrameTriggering.j1939requestable: Frame can be triggered by the J1939 request message. **J1939TpPg.requestable:** Parameter Group can be triggered by the J1939 request message.	
M2 Parameter	
CanFrameTriggering.j1939requestable, J1939TpPg.requestable	
Mapping Rule	**Mapping Type**
CanFrameTriggering.j1939requestable: CanFrameTriggering references a Frame where the aggregated PduToFrameMapping references the given Pdu. J1939TpPg.requestable: J1939TpPg references the given Pdu in the role sdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00072]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu

BSW Parameter	BSW Type
MetaDataTypeRef	ECUC-REFERENCE-DEF

BSW Description

Reference to meta data that is transported in the Pdu through the AUTOSAR layers.

Template Description

VariableDataPrototype:
A `VariableDataPrototype` represents a formalized generic piece of information that is typically mutable by the application software layer. `VariableDataPrototype` is used in various contexts and the specific context gives the otherwise generic `VariableDataPrototype` a dedicated semantics.

SenderReceiverToSignalMapping:
Mapping of a sender receiver communication data element to a signal.

SystemSignal:
The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances.

ISignal:
Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different Signal IPdus to multiple receivers.

To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping.

ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping).

In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup.

ISignalIPdu:
Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer.

A maximum of one dynamic length signal per IPdu is allowed.

M2 Parameter

`VariableDataPrototype`, `SenderReceiverToSignalMapping`, `SystemSignal`, `ISignal`, `ISignalIPdu`

Mapping Rule	Mapping Type
A MetaDataTypeRef shall be derived for a given Pdu if a MetaDataItemSet exists that refers to a VariablePrototype that is also referenced from a SenderReceiverToSignalMapping that in turn references a SystemSignal that is referenced by a ISignal that is mapped to an ISignalIPdu that is derived to the mentioned Pdu in EcuC.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00077]

BSW Module	BSW Context
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu

BSW Parameter	BSW Type
PduLength	ECUC-INTEGER-PARAM-DEF

BSW Description

Length of the Pdu in bytes. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits.

Template Description

▽

△

Pdu.length: Pdu length in bytes. In case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits. The Pdu length of zero bytes is allowed. **IPduMapping.pduMaxLength:** Define the maximum length in bytes which limits the length of the Pdu during gateway operation if the runtime length of the received Pdu exceeds this limit.
M2 Parameter
`Pdu.length`, `IPduMapping.pduMaxLength`

Mapping Rule	**Mapping Type**
1:1 mapping of Pdu.length in case that IPduMapping is not used. In case IPduMapping is used: 1:1 (sourceIPdu:targetIPdu) routing: When the SysTPduToPduTriggeringRef PduTriggering is referenced by an IPduMapping in the role sourceIPdu or targetIPdu,respectively, and that IPdu Mapping has a pduMaxLength defined then IPduMapping.pduMaxLength shall be used as Pdu Length for the derived PduRSrcPdu and PduRDestPdu, respectively. Otherwise use Pdu.length. 1:N (sourceIPdu:targetIPdu) routing: If 1:N (sourceIPdu:targetIPdu) routing is modeled, then the maximum length of the available IPduMapping.pduMaxLength and Pdu.length shall be derived for the PduRSrcPdu. The derivation of the length of each PduRDestPdu shall follow the rule for 1:1 routing. N:1 (sourceIPdu:targetIPdu): If N:1 (sourceIPdu:targetIPdu) routing is modeled, then the maximum length of the available IPduMapping.pduMaxLength and Pdu.length shall be derived for the PduRDestPdu. The derivation of the length of each PduRSrcPdu shall follow the rule for 1:1 routing.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00003]

BSW Module	**BSW Context**
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu

BSW Parameter	**BSW Type**
SysTPduToFrameTriggeringRef	ECUC-FOREIGN-REFERENCE-DEF
BSW Description	
Reference to the FrameTriggering from the SystemTemplate which this Pdu belongs to. SysTPduToFrameTriggeringRef shall be used for UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToPduTriggeringRef exists.	
Template Description	
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.	
M2 Parameter	
`FrameTriggering`	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00052]

BSW Module	**BSW Context**
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu
BSW Parameter	**BSW Type**

▽

△

SysTPduToPduTriggeringRef	ECUC-FOREIGN-REFERENCE-DEF
BSW Description	
Reference to the PduTriggering from the SystemTemplate which this Pdu represents. SysTPduToPduTriggeringRef shall be used for all Pdus except UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. For these Pdus, SysTPduToFrameTriggeringRef shall be used.	
Template Description	
The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu. Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface. If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.	
M2 Parameter	
PduTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00054]

BSW Module	**BSW Context**
EcuC	EcuC/EcucHardware/EcucCoreDefinition
BSW Parameter	**BSW Type**
EcucCoreId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID of the core.	
Template Description	
Optional definition of Core Id.	
M2 Parameter	
EcuPartitionToCoreMapping.coreId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00058]

BSW Module	**BSW Context**
EcuC	EcuC/EcucPartitionCollection
BSW Parameter	**BSW Type**
EcucPartition	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of one Partition on this ECU. One Partition will be implemented using one Os-Application.	
Template Description	
Partitions are used as error containment regions. They permit the grouping of SWCs and resources and allow to describe recovery policies individually for each partition. Partitions can be terminated or restarted during run-time as a result of a detected error.	
M2 Parameter	
EcuPartition	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuC_00005]

C.17 EcuM

BSW Module	**BSW Context**
EcuM	EcuM/EcuMConfiguration/EcuMCommonConfiguration

BSW Parameter	**BSW Type**
EcuMWakeupSource	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

These containers describe the configured wakeup sources.

Template Description

CommunicationConnector.createEcuWakeupSource:
If this parameter is available and set to true then a channel wakeup source shall be created for the PhysicalChannel referencing this CommunicationConnector.

EcuInstance.channelSynchronousWakeup:
If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.

CommunicationConnector.explicitWakeupChannel:
Defines a restriction which PhysicalChannels shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies.

CommunicationConnector.explicitWakeupPnc:
Defines a restriction which PNCs shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies.

M2 Parameter

`CommunicationConnector.createEcuWakeupSource`, `EcuInstance.channelSynchronousWakeup`, `CommunicationConnector.explicitWakeupChannel`, `CommunicationConnector.explicitWakeupPnc`

Mapping Rule	**Mapping Type**
If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.channel SynchronousWakeup equals FALSE, and at least one CommunicationConnector.explicitWakeup Channel reference exists, then derive one EcuMWakeupSource for this CommunicationConnector where each explicitWakeupChannel is derived into a EcuMComMChannelRef to the respective explicitWakeupChannel. If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.channel SynchronousWakeup equals TRUE, then derive one EcuMWakeupSource for the whole Ecu Instance. This EcuMWakeupSource shall refer via EcuMComMChannelRef to any one ComM channel associated with one of the CommunicationConnectors. If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.pnc SynchronousWakeup equals TRUE, and no CommunicationConnector.explicitWakeupPnc reference exists, then derive one EcuMWakeupSource for the whole EcuInstance. This Ecu MWakeupSource shall refer with EcuMComMPNCRef to all PNCs associated with all CommunicationConnectors of this EcuInstance. If CommunicationConnector.createEcuWakeupSource equals TRUE and at least one CommunicationConnector.explicitWakeupPnc reference exists, then derive one EcuMWakeup Source for this CommunicationConnector where each explicitWakeupPnc is derived into a Ecu MComMPNCRef to the respective explicitWakeupPnc. Further EcuMWakeupSources may be added by the integrator.	partial
Mapping Status	**ECUC Parameter ID**

▽

△

valid	[ECUC_EcuM_00150]

BSW Module	BSW Context
EcuM	EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource

BSW Parameter	BSW Type
EcuMComMChannelRef	ECUC-REFERENCE-DEF

BSW Description

This parameter could reference multiple Networks (channels) defined in the Communication Manager. No reference indicates that the wakeup source is not a communication channel.

Template Description

CommunicationConnector.createEcuWakeupSource:
If this parameter is available and set to true then a channel wakeup source shall be created for the PhysicalChannel referencing this CommunicationConnector.

EcuInstance.channelSynchronousWakeup:
If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.

CommunicationConnector.explicitWakeupChannel:
Defines a restriction which PhysicalChannels shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies.

M2 Parameter

`CommunicationConnector.createEcuWakeupSource`, `EcuInstance.channelSynchronousWakeup`, `CommunicationConnector.explicitWakeupChannel`

Mapping Rule	Mapping Type
If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.channel SynchronousWakeup equals FALSE, and at least one CommunicationConnector.explicitWakeup Channel reference exists, then add to this EcuMWakeupSource for each explicitWakeupChannel a EcuMComMChannelRef to the ComMChannel which represents the explicitWakeupChannel. If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.channel SynchronousWakeup equals TRUE, then add to this EcuMWakeupSource one EcuMCom MChannelRef the an arbitrary ComMChannel (note that ComM will wakeup all ComMChannels due to ComMSynchronousWakeUp = TRUE).	partial

Mapping Status	ECUC Parameter ID
valid	[ECUC_EcuM_00101]

BSW Module	BSW Context
EcuM	EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMWakeupSource

BSW Parameter	BSW Type
EcuMComMPNCRef	ECUC-REFERENCE-DEF

BSW Description

This is a reference to a one or more PNC's defined in the Communication Manager.

No reference indicates that the wakeup source is not assigned to a partial network.

Template Description

▽

△

CommunicationConnector.createEcuWakeupSource: If this parameter is available and set to true then a channel wakeup source shall be created for the PhysicalChannel referencing this CommunicationConnector. **EcuInstance.pncSynchronousWakeup:** If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping. **CommunicationConnector.explicitWakeupPnc:** Defines a restriction which PNCs shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies.
M2 Parameter
CommunicationConnector.createEcuWakeupSource, EcuInstance.pncSynchronousWakeup, CommunicationConnector.explicitWakeupPnc

Mapping Rule	**Mapping Type**
If CommunicationConnector.createEcuWakeupSource equals TRUE and EcuInstance.pncSynchronousWakeup equals TRUE, and no CommunicationConnector.explicitWakeupPnc reference exists, then add to this EcuMWakeupSource for each PNC associated with all CommunicationConnectors of this EcuInstance a EcuMComMPNCRef to the ComMPnc which represents the PNC. If CommunicationConnector.createEcuWakeupSource equals TRUE and at least one CommunicationConnector.explicitWakeupPnc reference exists, then add to this EcuMWakeup Source for each explicitWakeupPnc a EcuMComMPNCRef to the ComMPnc which represents the explicitWakeupPnc.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EcuM_00228]

C.18 Eth

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress

BSW Parameter	**BSW Type**
EthCtrlConfigEgressLastSchedulerRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the scheduler which is the last in the egress structure.	
Template Description	
Defines which CouplingPortScheduler is the last in the egress port structure.	
M2 Parameter	
CouplingPortDetails.lastEgressScheduler	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00052]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressQueue/EthCtrlConfig EgressQueueTransmissionSelection/EthCtrlConfigEgressQueueTransmissionSelectionCBSConfig

BSW Parameter	**BSW Type**

▽

△

EthCtrlConfigEgressQueueCreditBasedShaperIdleSlope	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the idle slope of a queue at egress side.	
Template Description	
Defines the increase of credit in bits per second for the CBS shaper.	
M2 Parameter	
CouplingPortCreditBasedShaper.idleSlope	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Eth_00149]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressQueue/EthCtrlConfigEgressQueueTransmissionSelection/EthCtrlConfigEgressQueueTransmissionSelectionETS
BSW Parameter	**BSW Type**
EthCtrlETSConfigAvailableBandwidthInPercent	ECUC-INTEGER-PARAM-DEF
BSW Description	
Represents the configuration of an enhanced transmission selection algorithm for one egress queue, where the available bandwidth is configured in percent. The percent value represents the available bandwidth for emission opportunities to transmit Ethernet frames calculated in bits. The resolution is 1%.	
Template Description	
Defines the available bandwidth in percent of an enhanced transmission selection algorithm (ETS).	
M2 Parameter	
CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Eth_00139]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigEgressQueue/EthCtrlConfigEgressQueueTransmissionSelection/EthCtrlConfigEgressQueueTransmissionSelectionETS
BSW Parameter	**BSW Type**
EthCtrlETSConfigAvailableBandwidthInWeightValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Represents the configuration of an enhanced transmission selection algorithm of an egress queue, where the available bandwidth is configured as weight value. The weight value represents the number of emission opportunities to transmit Ethernet frames.	
Template Description	
Defines the available bandwidth as weight value of an enhanced transmission selection algorithm (ETS).	
M2 Parameter	
CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Eth_00140]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress
BSW Parameter	**BSW Type**
EthCtrlConfigScheduler	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents a Scheduler on the egress side.	
Template Description	
Defines a scheduler for the CouplingPort egress structure.	
M2 Parameter	
CouplingPortScheduler	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00053]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler
BSW Parameter	**BSW Type**
EthCtrlConfigSchedulerAlgorithm	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the scheduler algorithm.	
Template Description	
Defines the schedule algorithm to be used.	
M2 Parameter	
CouplingPortScheduler.portScheduler	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00141]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/EthCtrlConfig SchedulerAlgorithm
BSW Parameter	**BSW Type**
ETH_SCHEDULER_ENHANCED_TRAFFIC_SHAPER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Represents a scheduler used for enhanced traffic shaping (e.g. weighted round robin).	
Template Description	
Scheduler used for enhanced traffic shaping (e.g. weighted round robin)	
M2 Parameter	
EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/EthCtrlConfig SchedulerAlgorithm
BSW Parameter	**BSW Type**
ETH_SCHEDULER_STRICT_PRIORITY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Represents a strict priority scheduler.	
Template Description	
Schedule algorithm "strict priority"	
M2 Parameter	
EthernetCouplingPortSchedulerEnum.strictPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler
BSW Parameter	**BSW Type**
EthCtrlConfigSchedulerPredecessor	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines an ordered list of predecessors for this scheduler.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00054]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/EthCtrlConfig SchedulerPredecessor
BSW Parameter	**BSW Type**
EthCtrlConfigSchedulerPredecessorOrder	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the order of the scheduler predecessors.	
Template Description	

▽

△

Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
Defined by the order of CouplingPortScheduler.predecessor.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00055]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlConfigEgress/EthCtrlConfigScheduler/EthCtrlConfig SchedulerPredecessor
BSW Parameter	**BSW Type**
EthCtrlConfigSchedulerPredecessorRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
Choice reference to the scheduler predecessor.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00056]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	**BSW Type**
EthCtrlDuplexMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Controls the duplex mode of the Ethernet controller.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
HALF_DUPLEX for _10BASE_T1S, FULL_DUPLEX for all other values.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00155]

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlDuplexMode
BSW Parameter	**BSW Type**

▽

△

FULL_DUPLEX	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Allows simultaneous data transmission and reception.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
EthCtrlDuplexMode shall be set to FULL_DUPLEX for every value with the exception of _10BASE_T1S.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlDuplexMode
BSW Parameter	**BSW Type**
HALF_DUPLEX	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Allows data transmission in both directions, but only one direction at a time.	
Template Description	
Physical layer interface 10BASE-T1S (10Mbit/s, 2 pairs). Used for automotive.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.10BASE-T1S	
Mapping Rule	**Mapping Type**
if _10BASE_T1S is used then EthCtrlDuplexMode shall be set to HALF_DUPLEX.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	**BSW Type**
EthCtrlFramePreemptionEnable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Configures whether frame preemption for this Ethernet controller is enabled. If this parameter is set to TRUE and the Ethernet controller hardware supports frame preemption, then frame preemption for this Ethernet controller is enabled.	
Template Description	
Defines whether frames handled by this CouplingPort may be preemted.	
M2 Parameter	
CouplingPortDetails.framePreemptionSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Eth_00142]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig

BSW Parameter	BSW Type
EthCtrlMacLayerSpeed	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Defines the baud rate of the MAC layer.

Template Description

Specifies the mac layer type of the CouplingPort.

M2 Parameter

`CouplingPort.macLayerType`

Mapping Rule	Mapping Type
• 100BASE-TX => 100M, • 1000BASE-TX =>1G, • 100BASE-T1 => 100M, • 1000BASE-T1 => 1G, • 10BASE-T1S => 10M, • 2500BASE-T1 => 2500M, • 5000BASE-T1 => 5G, • 10000BASE-T1 => 10G, • iEEE802-11P => can not be mapped	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00063]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed

BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_100M	ECUC-ENUMERATION-LITERAL-DEF

BSW Description

100 MBit/s

Template Description

EthernetPhysicalLayerTypeEnum.100BASE-TX:
Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs.

EthernetPhysicalLayerTypeEnum.100BASE-T1:
Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100BASE-T1 is the IEEE Standardized version of BroadRReach.

M2 Parameter

`EthernetPhysicalLayerTypeEnum.100BASE-TX`, `EthernetPhysicalLayerTypeEnum.100BASE-T1`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed

BSW Parameter	BSW Type
ETH_MAC_LAYER_SPEED_10G	ECUC-ENUMERATION-LITERAL-DEF



△

BSW Description	
10 GBit/s	
Template Description	
Ethernet Standard (IEEE 802.3ch) to support 10Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.10000BASE_T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed
BSW Parameter	**BSW Type**
ETH_MAC_LAYER_SPEED_10M	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
10 MBit/s	
Template Description	
Physical layer interface 10BASE-T1S (10Mbit/s, 2 pairs). Used for automotive.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.10BASE-T1S	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed
BSW Parameter	**BSW Type**
ETH_MAC_LAYER_SPEED_1G	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
1 GBit/s	
Template Description	
EthernetPhysicalLayerTypeEnum.1000BASE-T1: Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable. **EthernetPhysicalLayerTypeEnum.1000BASE-T:** Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1, EthernetPhysicalLayerTypeEnum.1000BASE-T	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed	
BSW Parameter		**BSW Type**
ETH_MAC_LAYER_SPEED_2500M		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
2500 MBit/s (2.5 GBit/s)		
Template Description		
Ethernet Standard (IEEE 802.3ch) to support 2.5Gbit/s over a single twisted pair cable.		
M2 Parameter		
EthernetPhysicalLayerTypeEnum.2500BASE_T1		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerSpeed	
BSW Parameter		**BSW Type**
ETH_MAC_LAYER_SPEED_5G		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
5 GBit/s		
Template Description		
Ethernet Standard (IEEE 802.3ch) to support 5Gbit/s over a single twisted pair cable.		
M2 Parameter		
EthernetPhysicalLayerTypeEnum.5000BASE_T1		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		**BSW Type**
EthCtrlMacLayerType		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Defines the physical MAC/PHY Ethernet Interface type of the ethernet controller.		
Template Description		
Specifies the mac layer type of the CouplingPort.		
M2 Parameter		
CouplingPort.macLayerType		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Eth_00039]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType
BSW Parameter	**BSW Type**
ETH_MAC_LAYER_TYPE_XGMII	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
MAC layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
Template Description	
Mac layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)	
M2 Parameter	
EthernetMacLayerTypeEnum.xGMII	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType
BSW Parameter	**BSW Type**
ETH_MAC_LAYER_TYPE_XMII	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
MAC layer interface (data) bandwith class 10-100Mbit/s (e.g. MII, RMII, RvMII, SMII)	
Template Description	
Mac layer interface (data) bandwith class 100Mbit/s and 10Mbit/s (e.g. RMII, RvMII, SMII, RvMII)	
M2 Parameter	
EthernetMacLayerTypeEnum.xMII	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacLayerType
BSW Parameter	**BSW Type**
ETH_MAC_LAYER_TYPE_XXGMII	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
MAC layer interface (data) bandwith class 10Gbit/s	
Template Description	
Mac layer interface (data) bandwith class 10Gbit/s	
M2 Parameter	
EthernetMacLayerTypeEnum.xXGMII	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacSecConfig
BSW Parameter	**BSW Type**
EthCtrlMacSecBypassDestinationMacAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Bypassed destination MAC address (48-bit physical address). Ethernet frames, which carrying the configured destination MAC address, shall bypass the MACsec protection handling.	
Template Description	
The attribute is used to define MAC destination addresses that are bypassed by MACsec. The provided MAC addresses will not be MACsec protected. Please note, IEEE 802.1X-2020 chapter 11.1.1 (Table 11-1) define specific MAC destination addresses to be bypassed.	
M2 Parameter	
MacSecGlobalKayProps.bypassDestinationMacAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00152]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacSecConfig
BSW Parameter	**BSW Type**
EthCtrlMacSecBypassEtherType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed EtherType. Ethernet frames, which carrying the configured EtherType, shall bypass the MACsec protection handling.	
Template Description	
This attribute is used to define EtherTypes that are bypassed by MACsec. The providedEtherType will not be MACsec protected.	
M2 Parameter	
MacSecGlobalKayProps.bypassEtherType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00154]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig/EthCtrlMacSecConfig
BSW Parameter	**BSW Type**
EthCtrlMacSecBypassVlan	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed VLAN-ID. Ethernet frames, which carrying the configured VLAN-ID, shall bypass the MACsec protection handling. Please note: VLAN-ID 0 is interpreted as no VLAN, i.e. untagged Ethernet frames bypass the MACsec protection handling.	
Template Description	
This attribute is used to define VLAN-IDs that are bypassed by MACsec. The provided VLAN-IDs will not be MACsec protected. (VLAN-ID 0 is interpreted as no-VLAN --> Bypass untagged traffic)	
M2 Parameter	
MacSecGlobalKayProps.bypassVlan	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00153]

BSW Module	BSW Context
Eth	Eth/EthConfigSet/EthCtrlConfig
BSW Parameter	**BSW Type**
EthCtrlPhyAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order.	
Template Description	
Media Access Control address (MAC address) that uniquely identifies each EthernetCommunicationController in the network.	
M2 Parameter	
EthernetCommunicationController.macUnicastAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Eth_00020]

C.19 EthIf

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfController
BSW Parameter	**BSW Type**
EthIfCtrlMtu	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the maximum transmission unit (MTU) of the EthIfCtrl in [bytes]. Note: In case a VLAN tag is used for the EthIfCtrl, the frame length of the Ethernet frame will increase by 4 bytes.	
Template Description	
This attribute specifies the maximum transmission unit in bytes.	
M2 Parameter	
EthernetCommunicationConnector.maximumTransmissionUnit	
Mapping Rule	**Mapping Type**
Different MTU values may be defined for different VLANs. Therefore the maximumTransmission Unit is specified in the EthernetCommunicationConnector. The value that is defined in the CommunicationConnector shall be used as the value of this parameter.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthIf_00032]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfVlanId		ECUC-INTEGER-PARAM-DEF
BSW Description		
A virtual-LAN is identified by this attribute according to IEEE 802.1Q.		
Template Description		
A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.		
M2 Parameter		
VlanConfig.vlanIdentifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthIf_00029]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfFrameConfig	
BSW Parameter		**BSW Type**
EthIfFrameType		ECUC-INTEGER-PARAM-DEF
BSW Description		
Selects the Ethernet frame type.		
Template Description		
Ethernet specific attributes to the Frame.		
M2 Parameter		
AbstractEthernetFrame		
Mapping Rule		**Mapping Type**
If an AbstractEthernetFrame is defined in the System Extract then it may be possible to derive this parameter from this information.		partial
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_EthIf_00122]

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		**BSW Type**
EthIfSwitchPortGroup		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration of EthIfSwitchPortGroups. If EthIfSwitchPortGroups are controlled by PNC one EthIfSwitchPortGroup per PNC shall exist. The host port shall be part of all EthIfSwitchPortGroups. The up link port of a master switch and the up link port of the slave switch shall be part of all EthIfSwitchPortGroups that contain EthSwtPorts belonging to the slave switch.		
Template Description		

▽

△

EcuInstance.ethSwitchPortGroupDerivation: Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. **CouplingPort.pncMapping:** Reference to the partial networks this CouplingPort participates in.
M2 Parameter
EcuInstance.ethSwitchPortGroupDerivation, CouplingPort.pncMapping

Mapping Rule	**Mapping Type**
Derive EthIfSwitchPortGroup from M2 according to the following requirements only if Ecu Instance.ethSwitchPortGroupDerivation is defined and set to TRUE: • For each EthernetPhysicalChannel that has CouplingPorts connected (CouplingPort has a Vlan Membership refering to the EthernetPhysicalChannel) exactly one EthIfSwitchPortGroup shall be derived containing all connected CouplingPorts via the EthIfPortRef. Thus a EthSwtPort may be part in several EthIfSwitchPortGroups. • For each PNC that is referenced by at least one CouplingPort exactly one EthIfSwitchPortGroup shall be derived. The referenced CouplingPort shall be part of the EthIfSwitchPortGroup via the EthIfPortRef. Thus a EthSwtPort may be part in several EthIfSwitchPortGroups. • If the CouplingPorts have no reference to any PNC or all referenced PNCs have no relation to this EthernetPhysicalChannel then the derived EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_CONTROL, because this EthIfSwitchPortGroup is switched by EthSM. • If the CouplingPorts have at least one reference to any PNC that has a relation to this Ethernet PhysicalChannel then the derived EthIfSwitchPortGroupRefSemantics shall have the value ETHIF_SWITCH_PORT_GROUP_LINK_INFO, because this EthIfSwitchPortGroup is only used for link status accumulation towards EthSM.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthIf_00057]

BSW Module	**BSW Context**
EthIf	EthIf/EthIfConfigSet/EthIfTransceiver

BSW Parameter	**BSW Type**
EthIfQualifiedUnexpectedLinkDownTime	ECUC-FLOAT-PARAM-DEF

BSW Description
Specifies the time in seconds an unexpected link down is qualified. This parameter is only used for those Ethernet channels where the ECU act as a passive communication slave (referenced EthTrcv set EthTrcvActAsSlavePassiveEnabled = TRUE). The value shall be a multiple integral of EthIf_MainFunctionState.
Template Description
EcuInstance.wakeUpOverBusSupported: Driver support for wakeup over Bus. **EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime:** This attribute specifies time when an unexpected link down is evaluated as link down and indicated to the AUTOSAR communication stack.
M2 Parameter
EcuInstance.wakeUpOverBusSupported, EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime

Mapping Rule	**Mapping Type**
If EcuInstance.wakeUpOverBusSupported is defined and set to TRUE and the aggregated EthernetCommunicatonController has set EthernetCommunicationController.slaveActAsPassive Slave to TRUE and the EthernetController aggregate an CouplingPort with physicalLayerType set to “100Base-T1” and “1000Base-T1”, respectively, than the corresponding EthIfTransceiver shall set EthIfQualifiedUnexpectedLinkDownTime to EthernetCommunicatonController.slaveQualified UnexpectedLinkDownTime.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthIf_00078]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfPortStartupActiveTime		ECUC-FLOAT-PARAM-DEF
BSW Description		
Denote the time delay after the mode "ETH_MODE_ACTIVE" of all EthIfSwitchPorts are requested via EthIf_StartAllPorts. This is only used for ports in EthIfSwtPortGroups which are not referenced by any EthIfController.		
Template Description		
EthernetCluster.couplingPortStartupActiveTime: The attribute specifies the time in second a coupling port is switched on to enable the host ECU (ECU that maintains an Ethernet switch) to listen to the network for potential network management requests. **EcuInstance.ethSwitchPortGroupDerivation:** Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.		
M2 Parameter		
EthernetCluster.couplingPortStartupActiveTime, EcuInstance.ethSwitchPortGroupDerivation		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthIf_00055]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfSwitchOffPortTimeDelay		ECUC-FLOAT-PARAM-DEF
BSW Description		
Denote the time delay after the mode "ETH_MODE_DOWN" of a EthIfSwitchPortGroup will be executed. This is only used for EthIfSwtPortGroups which are not referenced by any EthIfController. The time delay shall be greater than the UdpNm timings, because UdpNm shall finish its shutdown handling. (Repeat Message State, Prepare Bus-Sleep state, Bus-Sleep state).		
Template Description		
EthernetCluster.couplingPortSwitchoffDelay: Switch off delay for CouplingPorts in seconds. It denotes the delay of switching off couplingPorts after the request to switch off a couplingPort was issued. (e.g. switch off of Ethernet switch ports). **EcuInstance.ethSwitchPortGroupDerivation:** Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.		
M2 Parameter		
EthernetCluster.couplingPortSwitchoffDelay, EcuInstance.ethSwitchPortGroupDerivation		
Mapping Rule		**Mapping Type**
Derivation shall only be done if EcuInstance.ethSwitchPortGroupDerivation is available and set to TRUE.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthIf_00054]

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**

▽

△

EthIfWakeUpSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Configures if wake-up handling is supported or not: TRUE: wake-up handling is supported FALSE: wake-up handling is not supported This configuration parameter also enables particular other the API at Pre-Compile-Time, e.g. EthIf_CheckWakeup.	
Template Description	
EcuInstance.wakeUpOverBusSupported: Driver support for wakeup over Bus. **EthernetCommunicationController:** Ethernet specific communication port attributes.	
M2 Parameter	
EcuInstance.wakeUpOverBusSupported, EthernetCommunicationController	
Mapping Rule	**Mapping Type**
If EcuInstance.wakeUpOverBusSupported is defined and set to TRUE, the aggregated Ethernet CommunicatonController has set to TRUE and the the aggregated EthernetCommunicaton Controller aggregate an CouplingPort with physicalLayerType set to “100Base-T1” and“1000 Base-T1”,respectively, or the aggregated CouplingPort is part of a CouplingElement with coupling Type set to “switch” and the affected CouplingPorts have a physicalLayerType set to “100Base-T1” and “1000Base-T1”, respectively, then EthIfWakeUpSupport shall set to TRUE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthIf_00040]

C.20 EthSM

BSW Module	**BSW Context**
EthSM	EthSM
BSW Parameter	**BSW Type**
EthSMNetwork	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the Ethernet network-specific parameters of each Ethernet network. It also contains the reference to combination of controller and transceiver assigned to an Ethernet network.	
Template Description	
The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an Ethernet PhysicalChannel without an aggregated VLAN.	
M2 Parameter	
EthernetPhysicalChannel	
Mapping Rule	**Mapping Type**
1:1 For each EthernetPhysicalChannel the EcuInstance is connected to, one EthSMNetwork container is created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSM_00067]

BSW Module	BSW Context
EthSM	EthSM/EthSMNetwork
BSW Parameter	**BSW Type**
EthSMWakeupSleepOnDatalineEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if an active communication request shall be forwarded to the lower layer to trigger a wake-up on the Ethernet network, e.g. trigger a wake-up on dataline if OA TC10 compliant Ethernet hardware is used. If the parameter is set to TRUE and EthSM in called with COMM_FULL_COMMUNICATION_WITH_WAKEUP_REQUEST, then EthSM call the corresponding EthIfController everytime with ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST to trigger an wake-up request in the lower layer.	
Template Description	
EcuInstance.wakeUpOverBusSupported: Driver support for wakeup over Bus. **EcuInstance.ethSwitchPortGroupDerivation:** Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done. **EthernetCommunicationController:** Ethernet specific communication port attributes.	
M2 Parameter	
EcuInstance.wakeUpOverBusSupported, EcuInstance.ethSwitchPortGroupDerivation, EthernetCommunicationController	
Mapping Rule	**Mapping Type**
If EcuInstance.wakeUpOverBusSupported is defined and set to TRUE and the aggregated EthernetCommunicatonController aggregate an CouplingPort with physicalLayerType set to “100 Base-T1” and “1000Base-T1”,respectively, or the aggregated CouplingPort is partof a Coupling Element with couplingType set to “switch” and EcuInstance.ethSwitchPortGroupDeriviation is not defined or set to FALSE and the affected CouplingPorts have a physicalLayerType set to “100 Base-T1” and “1000Base-T1”, respectively, than the corresponding ComM channel shall set Com MWakeupSleepRequestEnabled to TRUE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSM_ 00109]

C.21 EthSwt

BSW Module	BSW Context
EthSwt	EthSwt
BSW Parameter	**BSW Type**
EthSwtConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of one Ethernet Switch.	
Template Description	
A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device.	
M2 Parameter	
CouplingElement	
Mapping Rule	**Mapping Type**
For each CouplingElement with couplingType=switch one EthSwtConfig shall be created.	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthSwt_00001]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtAtsGroupInstanceTable
BSW Parameter	**BSW Type**
EthSwtAtsGroupInstanceEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
AtsGroupInstanceEntry.	
Template Description	
Defines an Asynchronous Traffic Shapter (ATS) Group for a switch.	
M2 Parameter	
SwitchAsynchronousTrafficShaperGroupEntry	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00230]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtMacAddressLearningMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the MAC address learning mode specified by [7, IEEE802.1Q] either shared VLAN learning (SVL) or independent VLAN learning (IVL).	
Template Description	
Defines the MAC address learning mode of the Ethernet switch.	
M2 Parameter	
CouplingElement.switchMacAddressLearningMode	
Mapping Rule	**Mapping Type**
Set to SVL if CouplingElement.switchMacAddressLearningMode equals sharedVlanLearning or is not defined. Set to IVL if CouplingElement.switchMacAddressLearningMode equals independentVlanLearning.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00236]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtMacAddressLearningMode
BSW Parameter	**BSW Type**
IVL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

The Ethernet switch updates the ARL table (address resolution table) with an entry consiting of source MAC address and VLAN-ID of the received Ethernet frame and the ingress port from on which the Ethernet frame was received.	
Template Description	
sent without a VLAN tag	
M2 Parameter	
EthernetSwitchVlanEgressTaggingEnum.sentUntagged	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtMacAddressLearningMode
BSW Parameter	**BSW Type**
SVL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Ethernet switch updates the ARL table (address resolution table) with an entry consiting of source MAC address of the received Ethernet frame and the ingress port on which the Ethernet frame was received.	
Template Description	
will not be sent	
M2 Parameter	
EthernetSwitchVlanEgressTaggingEnum.notSent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtMacForwardingTable	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents a MAC forwarding table.	
Template Description	
Defines a set of macMulticastAddresses to be mapped to the CouplingPort.	
M2 Parameter	
MacAddressVlanMembership.macMulticastAddress	
Mapping Rule	**Mapping Type**
If a MacMulticastGroup is referenced at least once from CouplingPort.macAddressVlan Assignment.macMulticastAddress, then create an instance of EthSwtMacForwardingTable.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00205]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtMacForwardingTable
BSW Parameter	**BSW Type**
EthSwtMacForwardingTablePortRef	ECUC-REFERENCE-DEF
BSW Description	
References the ports the MAC shall be assigned to.	
Template Description	
Defines a set of macMulticastAddresses to be mapped to the CouplingPort.	
M2 Parameter	
MacAddressVlanMembership.macMulticastAddress	
Mapping Rule	**Mapping Type**
If a CouplingPort has a reference to a MacMulticastGroup (via CouplingPort.macAddressVlan Assignment.macMulticastAddress), then a reference to the EthSwtPort container which was created for that CouplingPort shall be added to the EthSwtMacForwardingTable container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00207]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtMacForwardingTable
BSW Parameter	**BSW Type**
EthSwtPredefinedMacAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Specifies a 48-bit physical addresses (MAC addresses) network byte order, which can be reached via the referenced port and if available via the referenced VLAN . Note that further addresses can be learned during runtime.	
Template Description	
A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.	
M2 Parameter	
MacMulticastGroup.macMulticastAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00206]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtMacForwardingTable
BSW Parameter	**BSW Type**
EthSwtVlanMembershipRef	ECUC-REFERENCE-DEF
BSW Description	
References the VLAN-IDs the MAC address shall be assigned to. Please note, this reference is used if EthSwtMacAddress LearningMode is set to IVL (independent VLAN learning).	
Template Description	
Defines a set of VLANs the set of macMulticastAddress apply to.	
M2 Parameter	
MacAddressVlanMembership.vlan	

▽

△

Mapping Rule	Mapping Type
If a CouplingPort has a reference to an EthernetPhysicalChannel (representing a VLAN) via CouplingPort.macAddressVlanAssignment.vlan, then a reference to the EthSwtVlanMembership container which was created for that EthernetPhysicalChannel (VLAN) shall be added to the EthSwtMacForwardingTable container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00237]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPSCM/EthSwtAtsInstanceTable
BSW Parameter	**BSW Type**
EthSwtAtsInstanceEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of an Asynchronous Traffic Scheduler configuration in the scope of the PSFP.	
Template Description	
Defines an Asynchronous Traffic Shaper (ATS) for the stream filter.	
M2 Parameter	
SwitchAtsInstanceEntry	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00228]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPSCM/EthSwtPSFP/EthSwtFilterMaxSduSizeTable
BSW Parameter	**BSW Type**
EthSwtFilterMaxSduSizeEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
EthSwtFilterMaxSduSizeEntry defines the maximum SDU size (size of an Ethernet packet) which is acceptable to be processed by the Ethernet switch. The value of EthSwtFilterMaxSduSizeEntry consider the size of the following parts of an Ethernet packet: • Preamble (7 byte) • SFD (start of frame delimiter) (1 byte) • Ethernet frame (Dst MAC,Src MAC, VLAN-tag, TypeField, Payload, CRC Checksum) • Minimum IPG (inter package gap) (12 byte times).	
Template Description	
Defines the maximum SDU size (size of an Ethernet package) which is acceptable to be processed by the Ethernet switch.	
M2 Parameter	
SwitchStreamFilterEntry.maxSduSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00224]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPSCM/EthSwtPSFP/EthSwtFlowMeteringTable
BSW Parameter	**BSW Type**
EthSwtFlowMeteringEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a flow metering.	
Template Description	
Defines a Flow Metering Entry for a switch.	
M2 Parameter	
SwitchFlowMeteringEntry	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00157]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPSCM/EthSwtPSFP/EthSwtStreamFilterTable
BSW Parameter	**BSW Type**
EthSwtStreamFilterEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents a stream filter, where EthSwtStreamFilterPriority and EthSwtAssignedStreamHandle are used to detect a matching Ethernet frame.	
Template Description	
Defines a Stream Filter Entry.	
M2 Parameter	
SwitchStreamFilterEntry	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00217]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPSCM/EthSwtPSFP/EthSwtStreamGateTable
BSW Parameter	**BSW Type**
EthSwtStreamGateEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a stream gate.	
Template Description	
Defines a Asynchronous Traffic Shapter (ATS) Group for a switch.	
M2 Parameter	
SwitchStreamGateEntry	
Mapping Rule	**Mapping Type**
1:1 mapping	full



△

Mapping Status	ECUC Parameter ID
draft	[ECUC_EthSwt_00155]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtPort	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of one Ethernet Switch Port.	
Template Description	
A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.	
M2 Parameter	
CouplingPort	
Mapping Rule	**Mapping Type**
For each CouplingElement.couplingPort of a CouplingElement with couplingType=switch one EthSwtPort shall be created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00005]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtFramePreemptionEnable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Configures whether frame preemption for this EthSwtPort is enabled. If the Ethernet switch hardware supports frame preemption and this parameter is set TRUE, then frame preemption for the corresponding EthSwtPort is enabled. If for some traffic class(es) EthSwtTrafficClassToPreemptionStatusAssignment at the egress port queue is additionally configured to ETHSWT_TRAFFIC_CLASS_PREEMPTABLE, then frame preemption for the respective traffic class(es) at the correponding EthSwtEgressPort is possible.	
Template Description	
Defines whether frames handled by this CouplingPort may be preemted.	
M2 Parameter	
CouplingPortDetails.framePreemptionSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00254]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	**BSW Type**
EthSwtPortDefaultTrafficClass	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Represents the default traffic class assignment. All Ethernet frames, where the priority associated with this Ethernet frame is not available in a EthSwtPortTrafficClassAssignment of this egress port, are assigned to the default traffic class.	
Template Description	
Defines the default traffic class for this CouplingPort.	
M2 Parameter	
CouplingPortDetails.defaultTrafficClass	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00247]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	**BSW Type**
EthSwtPortEgressLastSchedulerRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the port scheduler which is the last in the egress port structure.	
Template Description	
Defines which CouplingPortScheduler is the last in the egress port structure.	
M2 Parameter	
CouplingPortDetails.lastEgressScheduler	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00008]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	**BSW Type**
EthSwtPortEgressScheduler	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents a Scheduler in the egress port.	
Template Description	
Defines a scheduler for the CouplingPort egress structure.	
M2 Parameter	
CouplingPortScheduler	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00017]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler
BSW Parameter	**BSW Type**
EthSwtPortEgressSchedulerPredecessor	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines an ordered list of predecessors for this scheduler.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00019]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler/EthSwtPortEgressSchedulerPredecessor
BSW Parameter	**BSW Type**
EthSwtPortEgressPredecessorRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
Choice reference to the scheduler predecessor.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	
M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00010]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler/EthSwtPortEgressSchedulerPredecessor
BSW Parameter	**BSW Type**
EthSwtPortSchedulerPredecessorOrder	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the order of the scheduler predecessors. This value has to be understood as a relative value, i.e. the value shows only the relative ordering of the elements. The highest value has the highest priority and gaps are allowed (not dense based). The values need to be unique within one EthSwtPortScheduler.	
Template Description	
Ordered List of predecessor inputs. The first element has the highest priority. The following elements have decreasing priorities.	



△

M2 Parameter	
CouplingPortScheduler.predecessor	
Mapping Rule	**Mapping Type**
Defined by the order of CouplingPortScheduler.predecessor.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00020]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler
BSW Parameter	**BSW Type**
EthSwtPortSchedulerAlgorithm	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the scheduler algorithm.	
Template Description	
Defines the schedule algorithm to be used.	
M2 Parameter	
CouplingPortScheduler.portScheduler	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00018]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler/EthSwtPortSchedulerAlgorithm
BSW Parameter	**BSW Type**
ETHSWT_SCHEDULER_ENHANCED_TRAFFIC_SHAPER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Represents a scheduler used for enhanced traffic shaping (e.g. weighted round robin).	
Template Description	
Scheduler used for enhanced traffic shaping (e.g. weighted round robin)	
M2 Parameter	
EthernetCouplingPortSchedulerEnum.enhancedTrafficShaper	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortEgressScheduler/EthSwtPortSchedulerAlgorithm
BSW Parameter	**BSW Type**
ETHSWT_SCHEDULER_STRICT_PRIORITY	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
Represents a strict priority scheduler.	
Template Description	
Schedule algorithm "strict priority"	
M2 Parameter	
EthernetCouplingPortSchedulerEnum.strictPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	**BSW Type**
EthSwtPortPriorityToTrafficClassAssignment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines a priority based traffic class assignment. All Ethernet frames with a specific priority (EthSwtPortPriorityToTrafficClass AssignmentPriority) arriving at the egress side within the forwarding process, shall be assigned to the corresponding traffic class (EthSwtPortPriorityToTrafficClassAssignmentTrafficClass).	
Template Description	
Defines the priority to traffic class assignment.	
M2 Parameter	
CouplingPortDetails.ethernetTrafficClassAssignment	
Mapping Rule	**Mapping Type**
For each defined CouplingPortTrafficClassAssignment.priority one instance of this container shall be derived.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_ 00248]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortPriorityToTrafficClassAssignment
BSW Parameter	**BSW Type**
EthSwtPortPriorityToTrafficClassAssignmentPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the priority derived from the Ethernet frame, which is used to determine the corresponding traffic class, where this Ethernet frame shall be assigned to. The upper value range is restricted to the configured value of EthSwtUsedInternal PriorityUpperValue.	
Template Description	
Defines a priority which is mapped onto a Traffic Class.	
M2 Parameter	
CouplingPortTrafficClassAssignment.priority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_ 00249]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortPriorityToTrafficClassAssignment
BSW Parameter	**BSW Type**
EthSwtPortPriorityToTrafficClassAssignmentTrafficClass	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the traffic class value where an Ethernet frame, with the corresponding priority, is assigned to. The upper value range is restricted to the configured value of EthSwtUsedTrafficClassUpperValue.	
Template Description	
Defines the Traffic Class which is assigned.	
M2 Parameter	
CouplingPortTrafficClassAssignment.trafficClass	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00250]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress
BSW Parameter	**BSW Type**
EthSwtPortQueue	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents a Queue at the egress port.	
Template Description	
Defines a FIFO for the CouplingPort egress structure.	
M2 Parameter	
CouplingPortFifo	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00182]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtPortEgressQueue TransmissionSelection
BSW Parameter	**BSW Type**
EthSwtPortEgressQueueTransmissionSelectionCBSConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents the configuration of a credit based shaper transmission selection algorithm of an egress port queue. This configuration is used if the EthSwtPortEgressQueueTransmissionSelectionAlgorithm is set to ETHSWT_TRANSMISSION_SELECTION_ALGORITHM_CBS. Lower and upper boundaries of the CBS credit are not configurable by the EthSwt driver. If the credit counter of an implementation should over- or under-run its upper or lower limit at runtime, the resulting behavior is not specified by AUTOSAR or by IEEE and remains implementation specific.	
Template Description	
Defines a Credit Based Shaper (CBS) for the CouplingPort egress structure.	

▽

△

M2 Parameter	
CouplingPortCreditBasedShaper	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00187]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtPortEgressQueue TransmissionSelection
BSW Parameter	**BSW Type**
EthSwtPortEgressQueueTransmissionSelectionETSConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Represents the configuration of an enhanced transmission selection algorithm of an egress port queue. This configuration is used if the EthSwtPortEgressQueueTransmissionSelectionAlgorithm is set to ETHSWT_SCHEDULER_ENHANCED_TRAFFIC_SHAPER. The subordinated configuration parameters allow to configure the ETS in different variants to support commonly used scheduler algorithms (e.g. weighted round robin).	
Template Description	
Defines a scheduler used for enhanced traffic shaping (e.g. weighted round robin).	
M2 Parameter	
CouplingPortEnhancedTrafficShaper	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00251]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtPortEgressQueue TransmissionSelection/EthSwtPortEgressQueueTransmissionSelectionETSConfig
BSW Parameter	**BSW Type**
EthSwtETSConfigAvailableBandwidthInPercent	ECUC-INTEGER-PARAM-DEF
BSW Description	
Represents the configuration of an enhanced transmission selection algorithm for one egress port queue, where the available bandwidth is configured in percent. The percent value represents the available bandwidth for emission opportunities to transmit Ethernet frames calculated in bits. The resolution is 1%.	
Template Description	
Defines the available bandwidth in percent of an enhanced transmission selection algorithm (ETS).	
M2 Parameter	
CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInPercent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00252]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtPortEgressQueue TransmissionSelection/EthSwtPortEgressQueueTransmissionSelectionETSConfig
BSW Parameter	**BSW Type**
EthSwtETSConfigAvailableBandwidthInWeightValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Represents the configuration of an enhanced transmission selection algorithm of an egress port queue, where the available bandwidth is configured as weight value. The weight value represents the number of emission opportunities to transmit Ethernet frames.	
Template Description	
Defines the available bandwidth as weight value of an enhanced transmission selection algorithm (ETS).	
M2 Parameter	
CouplingPortEnhancedTrafficShaper.etsAvailableBandwidthInWeightValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_ 00253]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue
BSW Parameter	**BSW Type**
EthSwtPortQueueMinimumLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Queue minimum length in Byte. This assignment is used to configure a guaranteed size of a configured Queue.	
Template Description	
FIFO minimum length in Byte. An actual configuration/hardware may use a bigger value.	
M2 Parameter	
CouplingPortFifo.minimumFifoLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_ 00184]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue
BSW Parameter	**BSW Type**
EthSwtPortQueueTrafficClassAssignment	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the traffic class where this egress port queue is assigned to. All Ethernet frames which arrive at the corresponding egress port are considered to be enqueued in this egress port queue, where the assigned traffic class of the Ethernet frame match to the assigned traffic class of this egress port queue.	
Template Description	
Defines a set of Traffic Classes which shall be handled by this FIFO.	
M2 Parameter	
CouplingPortFifo.assignedTrafficClass	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00185]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue
BSW Parameter	**BSW Type**
EthSwtTrafficClassToPreemptionStatusAssignment	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the preemption status for the traffic class which is derived from the priority via EthSwtPortPriorityToTrafficClass Assignment. If this parameter is set to ETHSWT_TRAFFIC_CLASS_PREEMPTABLE, then the Ethernet frames assigned to the corresponding traffic class could be preempted within the transmission process. If set to ETHSWT_TRAFFIC_CLASS_EXPRESS, then the Ethernet frames assigned to the corresponding traffic class will never be preempted.	
Template Description	
Defines whether frames assigned to the traffic class associated with this CouplingPortFifo may be preempted or not.	
M2 Parameter	
CouplingPortFifo.trafficClassPreemptionSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00255]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtTrafficClassTo PreemptionStatusAssignment
BSW Parameter	**BSW Type**
ETHSWT_TRAFFIC_CLASS_EXPRESS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Traffic class will never be preempted.	
Template Description	
Frames assigned to the traffic class associated with this CouplingPortFifo will never be preempted.	
M2 Parameter	
EthernetCouplingPortPreemptionEnum.express	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortEgress/EthSwtPortQueue/EthSwtTrafficClassToPreemptionStatusAssignment

BSW Parameter	BSW Type
ETHSWT_TRAFFIC_CLASS_PREEMPTABLE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Traffic class can/may be preempted.

Template Description
Frames assigned to the traffic class associated with this CouplingPortFifo may be preempted.

M2 Parameter
EthernetCouplingPortPreemptionEnum.preemptable

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress

BSW Parameter	BSW Type
EthSwtPortIngressDefaultPriority	ECUC-INTEGER-PARAM-DEF

BSW Description
Default priority for ingress.

Template Description
Standard output-priority outgoing Frames will be tagged with. Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed. In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.

M2 Parameter
VlanMembership.defaultPriority

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00096]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress

BSW Parameter	BSW Type
EthSwtPortIngressDefaultVlan	ECUC-INTEGER-PARAM-DEF

BSW Description
Default VLAN for ingress.

Template Description

M2 Parameter
CouplingPort.defaultVlan

Mapping Rule	Mapping Type



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00095]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress
BSW Parameter	**BSW Type**
EthSwtPortIngressDropUntagged	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the ingress behavior for untagged frames.	
Template Description	
Defines the handling of frames at the ingress port.	
M2 Parameter	
CouplingPort.receiveActivity	
Mapping Rule	**Mapping Type**
If CouplingPort.receiveActivity is set to "dropUntagged" then EthSwtPortIngressDropUntagged shall be set to true. If CouplingPort.receiveActivity is set to something different than "dropUntagged" then EthSwtPort IngressDropUntagged shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00097]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress
BSW Parameter	**BSW Type**
EthSwtPortIngressVlanTranslationTable	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines 0..* entries of the form (IngressVlanID, TranslatedVlanID) that define the ingress Vlan translation. The IngressVlanID is the VlanID read from the incoming frame upon reception (ingress), which is replaced by the corresponding TranslatedVlanID upon ingress Vlan translation.	
Template Description	
This element defines one ingress Vlan translation entry in which the IngressVlanID from the incoming frame is replaced by the TranslatedVlanID.	
M2 Parameter	
EthernetVlanTranslationTable	
Mapping Rule	**Mapping Type**
Define container if EthernetVlanTransitionTable exists on the CouplingPort that corresponds to Eth SwtPort that is aggregating this container via EthSwtPortIngress container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00256]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortIngressVlanTranslationTable

BSW Parameter	BSW Type
EthSwtPortIngressVlanTranslationTableEntry	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This container defines the mapping between the IngressVlanID (VlanID read from the received frame upon ingress) and the corresponding TranslatedVlanID upon ingress Vlan translation.

Template Description

This element defines one ingress Vlan translation entry in which the IngressVlanID from the incoming frame is replaced by the TranslatedVlanID.

M2 Parameter

`EthernetVlanTranslationTable`

Mapping Rule	Mapping Type
Define container for EthernetVlanTransitionTable that exists on the CouplingPort that corresponds to EthSwtPort that is aggregating this container via EthSwtPortIngress container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00257]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortIngressVlanTranslationTable/EthSwtPortIngressVlanTranslationTableEntry

BSW Parameter	BSW Type
EthSwtIngressVlanId	ECUC-INTEGER-PARAM-DEF

BSW Description

Incoming VlanID from received frame.

Template Description

Incoming VlanID from received frame

M2 Parameter

`EthernetVlanTranslationTable.ingressVlanId`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00258]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortIngressVlanTranslationTable/EthSwtPortIngressVlanTranslationTableEntry

BSW Parameter	BSW Type
EthSwtTranslatedVlanId	ECUC-INTEGER-PARAM-DEF

BSW Description

Mapped VlanID after ingress Vlan translation.

Template Description

Mapped VlanID after ingress Vlan translation

M2 Parameter

`EthernetVlanTranslationTable.translatedVlanId`

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00259]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress
BSW Parameter	**BSW Type**
EthSwtPortPriorityRegeneration	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines a priority regeneration where the EthSwtPortPriorityRegenerationIngressPCP is replaced by EthSwtPortPriority RegenerationRegeneratedPriority. The EthSwtPortPriorityRegeneration is mandatory and shall always be available. An EthSwtPortPriorityRegeneration shall have 8 mappings, one for each priority. Rational: an Ethernet switch always performs a priority regeneration.	
Template Description	
M2 Parameter	
CouplingPortDetails.ethernetPriorityRegeneration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00057]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortPriorityRegeneration
BSW Parameter	**BSW Type**
EthSwtPortPriorityRegenerationIngressPCP	ECUC-INTEGER-PARAM-DEF
BSW Description	
PCP (VLAN-priority) in the incoming message.	
Template Description	
Message priority of the incoming message. range: 0-7	
M2 Parameter	
EthernetPriorityRegeneration.ingressPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00058]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortIngress/EthSwtPortPriorityRegeneration
BSW Parameter	**BSW Type**
EthSwtPortPriorityRegenerationRegeneratedPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Message priority the incoming message will be tagged with.	
Template Description	
Regenerated message priority. range: 0-7	
M2 Parameter	
EthernetPriorityRegeneration.regeneratedPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00059]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortMacLayerType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the MAC layer type of this EthSwtPort.	
Template Description	
Specifies the mac layer type of the CouplingPort.	
M2 Parameter	
CouplingPort.macLayerType	
Mapping Rule	**Mapping Type**
ETHSWT_PORT_MAC_LAYER_TYPE_XMII => if macLayerType = xMII ETHSWT_PORT_MAC_LAYER_TYPE_XGMII => if macLayerType = xGMII ETHSWT_PORT_MAC_LAYER_TYPE_XXGMII => if macLayerType = xXGMII	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00072]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortMacSecBypassDestinationMacAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Bypassed destination MAC address (48-bit physical address). Ethernet frames, which carrying the configured destination MAC address, shall bypass the MACsec protection handling.	
Template Description	
The attribute is used to define MAC destination addresses that are bypassed by MACsec. The provided MAC addresses will not be MACsec protected. Please note, IEEE 802.1X-2020 chapter 11.1.1 (Table 11-1) define specific MAC destination addresses to be bypassed.	
M2 Parameter	



△

MacSecGlobalKayProps.bypassDestinationMacAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00265]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortMacSecBypassEtherType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed EtherType. Ethernet frames, which carrying the configured EtherType, shall bypass the MACsec protection handling.	
Template Description	
This attribute is used to define EtherTypes that are bypassed by MACsec. The providedEtherType will not be MACsec protected.	
M2 Parameter	
MacSecGlobalKayProps.bypassEtherType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00262]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortMacSecBypassVlan	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed VLAN-ID. Ethernet frames, which carrying the configured VLAN-ID, shall bypass the MACsec protection handling. Please note: VLAN-ID 0 is interpreted as no VLAN, i.e. untagged Ethernet frames bypass the MACsec protection handling.	
Template Description	
This attribute is used to define VLAN-IDs that are bypassed by MACsec. The provided VLAN-IDs will not be MACsec protected. (VLAN-ID 0 is interpreted as no-VLAN --> Bypass untagged traffic)	
M2 Parameter	
MacSecGlobalKayProps.bypassVlan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00263]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**

▽

△

EthSwtPortPhysicalLayerType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the physical layer type of this EthSwtPort.	
Template Description	
Specifies the physical layer type of the CouplingPort.	
M2 Parameter	
CouplingPort.physicalLayerType	
Mapping Rule	**Mapping Type**
• ETHSWT_PORT_100BASE_TX => if physicalLayerType = 100BASE-TX • ETHSWT_PORT_1000BASE_T => if physicalLayerType = 1000BASE-T • ETHSWT_PORT_100BASE_T1 => if physicalLayerType = 100BASE-T1 • ETHSWT_PORT_1000BASE_T1 => if physicalLayerType = 1000BASE-T1 • ETHSWT_PORT_10BASE_T1S => if physicalLayerType = 10BASE-T1S • ETHSWT_PORT_2500BASE_T1 => if physicalLayerType = 2500BASE_T1 • ETHSWT_PORT_5000BASE_T1 => if physicalLayerType = 5000BASE_T1 • ETHSWT_PORT_10000BASE_T1 => if physicalLayerType = 10000BASE_T1	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00054]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_10000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 10GBASE-T1 (10Gbit/s, 1pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 10000BASE_T1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_1000BASE_T	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	

▽

△

Mapping Rule	Mapping Type
if physicalLayerType = 1000BASE-T	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_1000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 1000BASE-T1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_100BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 100BASE-T1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_100BASE_TX	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	

▽

△

EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 100BASE-TX	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_10BASE_T1S	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 10BASE-T1S (10Mbit/s, 1 pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 10BASE-T1S	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_2500BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 2.5GBASE-T1 (2.5Gbit/s, 1pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 2500BASE_T1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtPort/EthSwtPortPhysicalLayerType
BSW Parameter	**BSW Type**
ETHSWT_PORT_5000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 5GBASE-T1 (5Gbit/s, 1pair). Used for automotive.	
Template Description	
Specifies physical layer types of Ethernet transceiver links.	

▽

△

M2 Parameter	
EthernetPhysicalLayerTypeEnum	
Mapping Rule	**Mapping Type**
if physicalLayerType = 5000BASE_T1	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortRole	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Set a special role of the Ethernet switch port. It is either a host port or a up link port. If not configured it is a standard port.	
Template Description	
Defines the role this CouplingPort takes in the context of the CouplingElement.	
M2 Parameter	
CouplingPort.couplingPortRole	
Mapping Rule	**Mapping Type**
hostPort mapps to ETHSWT_HOST_PORT. upLinkPort mapps to ETHSWT_UP_LINK_PORT. standardPort mapps to non configured EthSwtPortRole.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00101]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtPort
BSW Parameter	**BSW Type**
EthSwtPortTrcvRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the Ethernet transceiver driver this EthSwtPort is connected with.	
Template Description	
Specifies the physical layer type of the CouplingPort.	
M2 Parameter	
CouplingPort.physicalLayerType	
Mapping Rule	**Mapping Type**
The reference shall be set if the CouplingPort for which the EthSwtPort is created has a defined physicalLayerType. The value of the parameter EthTrcvPhysLayerType that is defined within the referenced EthTrcv Config container shall be derived from CouplingPort.physicalLayerType.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00041]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable
BSW Parameter	**BSW Type**
EthSwtStreamIdentificationEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a stream identification.	
Template Description	
SwitchStreamIdentification	
M2 Parameter	
SwitchStreamIdentification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00140]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry
BSW Parameter	**BSW Type**
EthSwtStreamFilterAction	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a stream filter action.	
Template Description	
SwitchStreamIdentification	
M2 Parameter	
SwitchStreamIdentification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00143]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction
BSW Parameter	**BSW Type**
EthSwtStreamFilterActionBlockSource	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables Blocking all frames from the MAC address.	
Template Description	
Enables Blocking all frames from the MAC address.	
M2 Parameter	
SwitchStreamIdentification.filterActionBlockSource	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00145]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction
BSW Parameter	**BSW Type**
EthSwtStreamFilterActionDestinationPortModification	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines the action to modify the destination port vector determined by the frame forwarding process for a particular Ethernet frame. The egress destination port vector of an Ethernet frame can be extended, limited, or overwritten.	
Template Description	
Defines the action to modify the destination port(s) determined by the frame forwarding process for an particular Ethernet frame. Either the egress destination of an Ethernet frame is extended or overwritten.	
M2 Parameter	
SwitchStreamFilterActionDestPortModification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00148]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction/EthSwtStreamFilterActionDestinationPortModification
BSW Parameter	**BSW Type**
EthSwtStreamFilterActionDestinationPortModificationEgressPortRef	ECUC-REFERENCE-DEF
BSW Description	
Defines a set of destination ports (egress ports) used for the modification of the egress destination of an Ethernet frame.	
Template Description	
Reference to the egress ports used as the target of the filter action to modify the egress port.	
M2 Parameter	
SwitchStreamFilterActionDestPortModification.egressPort	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00149]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction/EthSwtStreamFilterActionDestinationPortModification
BSW Parameter	**BSW Type**
EthSwtStreamFilterActionDestinationPortModificationType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the method to modify the egress destination port vector.	
Template Description	
Defines the method to modify the egress destination. Either overwrite or extend the egress destination.	
M2 Parameter	
SwitchStreamFilterActionDestPortModification.modification	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00150]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction/EthSwtStreamFilterActionDestinationPortModification/EthSwtStreamFilterActionDestinationPortModificationType
BSW Parameter	**BSW Type**
ETHSWT_STREAM_EGRESS_DESTINATION_EXTEND	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Extend the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This corresponds to the union of the existing egress destination port vector and the configured egress destination port(s)	
Template Description	
Extend the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This corresponds to the union of the existing egress destination port vector and the configured egress destination port(s).	
M2 Parameter	
SwitchStreamFilterActionPortModificationEnum.extend	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction/EthSwtStreamFilterActionDestinationPortModification/EthSwtStreamFilterActionDestinationPortModificationType
BSW Parameter	**BSW Type**
ETHSWT_STREAM_EGRESS_DESTINATION_LIMIT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Limit the egress destination port vector of an Ethernet frame to the configured egress destination port(s). This corresponds to the intersection of the existing egress destination port vector and the configured egress destination port(s)	
Template Description	
Limit the egress destination port vector of an Ethernet frame to the configured egress destination port(s). This corresponds to the intersection of the existing egress destination port vector and the configured egress destination port(s).	
M2 Parameter	
SwitchStreamFilterActionPortModificationEnum.limit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction/EthSwtStreamFilterActionDestinationPortModification/EthSwtStreamFilterActionDestinationPortModificationType

BSW Parameter	BSW Type
ETHSWT_STREAM_EGRESS_DESTINATION_OVERWRITE	ECUC-ENUMERATION-LITERAL-DEF

BSW Description
Overwrite the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This results in the existing egress destination port vector being replaced by the configured egress destination port(s).

Template Description
Overwrite the egress destination port vector of an Ethernet frame with the configured egress destination port(s). This results in the existing egress destination port vector being replaced by the configured egress destination port(s).

M2 Parameter
SwitchStreamFilterActionPortModificationEnum.overwrite

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction

BSW Parameter	BSW Type
EthSwtStreamFilterActionDropFrame	ECUC-BOOLEAN-PARAM-DEF

BSW Description
Enables Drop Frame action.

Template Description
Enables Drop Frame action.

M2 Parameter
SwitchStreamIdentification.filterActionDropFrame

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00144]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterAction

BSW Parameter	BSW Type
EthSwtStreamFilterActionVlanModification	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Defines the action to modify the VLAN-ID within a VLAN-tag of an Ethernet frame.

Template Description
Defines the action to modify the VLAN-ID within a VLAN tag of an Ethernet frame.

M2 Parameter
SwitchStreamIdentification.filterActionVlanModification



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00146]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry
BSW Parameter	**BSW Type**
EthSwtStreamFilterRule	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a filter rule.	
Template Description	
Definition of a stream filter rule for this SwitchStreamIdentification.	
M2 Parameter	
SwitchStreamIdentification.streamFilterRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00141]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterRule
BSW Parameter	**BSW Type**
EthSwtStreamFilterIEEE1722StreamId	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of the filter IEEE1722 Stream Id. Specifies a 64 bit Stream Id. Please note: According to IEEE802.1Q specification, the 48 most significant bits represents a source MAC address and the 16 least significant bits a unique id. AUTOSAR deviates from this definition and allow to have any kind of MAC address (e.g. source or destination or MAC multicast address)	
Template Description	
Configuration of filter rules for IP and TP.	
M2 Parameter	
StreamFilterIEEE1722Tp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00232]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterRule/EthSwtStreamFilterIEEE1722StreamId
BSW Parameter	**BSW Type**
EthSwtStreamFilterStreamMacId	ECUC-STRING-PARAM-DEF

▽

△

BSW Description	
Specifies the 48-bit physical address (MAC address) part of the IEEE1722 Stream Id.	
Template Description	
Filter to match IEEE1722Tp packets with the stream Id number. Defined as 64bit stream id.	
M2 Parameter	
StreamFilterIEEE1722Tp.streamId	
Mapping Rule	**Mapping Type**
Take the MAC address part from the 64bit streamId.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00233]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtStreamIdentificationTable/EthSwtStreamIdentificationEntry/EthSwtStreamFilterRule/EthSwtStreamFilterIEEE1722StreamId
BSW Parameter	**BSW Type**
EthSwtStreamFilterStreamUniqueId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the 16-bit unique part of the IEEE1722 Stream Id.	
Template Description	
Filter to match IEEE1722Tp packets with the stream Id number. Defined as 64bit stream id.	
M2 Parameter	
StreamFilterIEEE1722Tp.streamId	
Mapping Rule	**Mapping Type**
Take the unique id part from the 64bit streamId.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00234]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtUnknownMacDestAddressConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition to which EthSwtPorts an Ethernet frame shall be forwarded if the destination MAC address is not present in the address resolution lookup (ARL) table.	
Template Description	
A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device.	
M2 Parameter	
CouplingElement	
Mapping Rule	**Mapping Type**
For each CouplingElement with couplingType=switch one EthSwtConfig shall be created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00239]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtUnknownMacDestAddressConfig
BSW Parameter	**BSW Type**
EthSwtDestPortsForUnknownMulticastMacDestAddressRef	ECUC-REFERENCE-DEF
BSW Description	
This parameter specifies the egress ports frames with unknown multicast MAC destination addresses (without a matching ARL entry) are forwarded on. Note that "Drop Unknown Multicast" behavior can be achieved by not referencing any EthSwtPort.	
Template Description	
Defines a set of macMulticastAddresses to be mapped to the CouplingPort.	
M2 Parameter	
MacAddressVlanMembership.macMulticastAddress	
Mapping Rule	**Mapping Type**
If a CouplingPort has a reference to a MacMulticastGroup (via CouplingPort.macAddressVlan Assignment.macMulticastAddress), then a reference to the EthSwtPort container which was created for that CouplingPort shall be added to the EthSwtMacForwardingTable container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00241]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtUnknownMacDestAddressConfig
BSW Parameter	**BSW Type**
EthSwtDestPortsForUnknownUnicastMacDestAddressRef	ECUC-REFERENCE-DEF
BSW Description	
This parameter specifies the egress ports frames with unknown unicast MAC destination addresses (without a matching ARL entry) are forwarded on. Note that "Flooding" can be achieved by referencing all EthSwtPorts.	
Template Description	
Defines a set of macMulticastAddresses to be mapped to the CouplingPort.	
M2 Parameter	
MacAddressVlanMembership.macMulticastAddress	
Mapping Rule	**Mapping Type**
If a CouplingPort has a reference to a MacMulticastGroup (via CouplingPort.macAddressVlan Assignment.macMulticastAddress), then a reference to the EthSwtPort container which was created for that CouplingPort shall be added to the EthSwtMacForwardingTable container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00240]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtUnknownMacDestAddressConfig
BSW Parameter	**BSW Type**
EthSwtDestVlanForUnknownMacDestAddressRef	ECUC-REFERENCE-DEF
BSW Description	
Optional reference to a set of VLANs to define that the owning EthSwtUnknownMacDestAddressConfig is applicable for these VLANs only.	

▽

△

Template Description	
Defines a set of VLANs the set of macMulticastAddress apply to.	
M2 Parameter	
MacAddressVlanMembership.vlan	
Mapping Rule	**Mapping Type**
If a CouplingPort has a reference to an EthernetPhysicalChannel (representing a VLAN) via CouplingPort.macAddressVlanAssignment.vlan, then a reference to the EthSwtVlanMembership container which was created for that EthernetPhysicalChannel (VLAN) shall be added to the EthSwtMacForwardingTable container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00242]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtUsedInternalPriorityUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configure the upper value of the used internal priority range in the Ethernet switch. The range of used internal priority values is defined from 0 to EthSwtUsedInternalPriorityUpperValue, where zero is the lowest and EthSwtUsedInternalPriorityUpperValue the highest internal priority value. Each priority is mapped to at least one traffic class.	
Template Description	
CouplingPortTrafficClassAssignment.priority: Defines a priority which is mapped onto a Traffic Class. **EthernetPriorityRegeneration.regeneratedPriority:** Regenerated message priority. range: 0-7	
M2 Parameter	
CouplingPortTrafficClassAssignment.priority, EthernetPriorityRegeneration.regeneratedPriority	
Mapping Rule	**Mapping Type**
The value shall be derived to have at least the highes value used on M2 upstream attributes.	partial
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00245]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtUsedTrafficClassUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configure the upper value of the used traffic class range in the Ethernet switch. The range of used traffic classes is defined from 0 to EthSwtUsedTrafficClassUpperValue. A traffic class is associated with exactly one egress queue at an egress port.	
Template Description	

▽

△

CouplingPortTrafficClassAssignment.trafficClass: Defines the Traffic Class which is assigned. **CouplingPortFifo.assignedTrafficClass:** Defines a set of Traffic Classes which shall be handled by this FIFO.	
M2 Parameter	
CouplingPortTrafficClassAssignment.trafficClass, CouplingPortFifo.assignedTrafficClass	
Mapping Rule	**Mapping Type**
The value shall be derived to have at least the highes value used on M2 upstream attributes.	partial
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_EthSwt_00246]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig
BSW Parameter	**BSW Type**
EthSwtVlanMembership	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Determines the membership of this Ethernet switch and the referenced ports to the virtual network, i.e. frames with this VID can be received and transmitted via the referenced ports.	
Template Description	
Static logical channel or VLAN binding to a switch-port. The reference to an EthernetPhysicalChannel without a VLAN defined represents the handling of untagged frames.	
M2 Parameter	
VlanMembership	
Mapping Rule	**Mapping Type**
For each CouplingElement.vlanMembership of a CouplingElement with couplingType=switch one EthSwtVlanMembership shall be created, merged based on VlanConfig.vlanIdentifier.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00199]

BSW Module	**BSW Context**
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership
BSW Parameter	**BSW Type**
EthSwtVlanMembershipId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Determines the VID of the virtual network this port belongs to.	
Template Description	
A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.	
M2 Parameter	
VlanConfig.vlanIdentifier	
Mapping Rule	**Mapping Type**
if a EthernetPhysicalChannel.vlan.vlanIdentifier is defined the value of vlanIdentifier shall be used for EthSwtPortVlanMembershipId. If no EthernetPhysicalChannel.vlan or EthernetPhysical Channel.vlan.vlanIdentifier is defined then EthSwtPortVlanMembershipId shall be set to 0.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00202]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership
BSW Parameter	**BSW Type**
EthSwtVlanMembershipPortRefEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Determines the VLAN membership of one referenced ports to the virtual network and the according forwarding type (NOT_SENT, SENT_UNTAGGED, SENT_TAGGED).	
Template Description	
M2 Parameter	
VlanMembership.vlan	
Mapping Rule	**Mapping Type**
If a CouplingPort has a CouplingPort.vlanMembership defined, then the EthSwtVlanMembership created for that specific VlanConfig.vlanIdentifier shall have an EthSwtVlanMembershipPortRef Entry container for that CouplingPort.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00203]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership/EthSwtVlanMembershipPortRefEntry
BSW Parameter	**BSW Type**
EthSwtVlanForwardingType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines how the message with a specific VLAN-ID at the referenced port shall be handled.	
Template Description	
Attribute denotes whether a VLAN tagged ethernet frame will be 1. sent with its VLAN tag (sentTagged) 2. sent without a VLAN tag (sentUntagged) 3. will be dropped at this port (notSent or VLAN not member of this list)	
M2 Parameter	
VlanMembership.sendActivity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_00026]

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership/EthSwtVlanMembershipPortRefEntry/EthSwtVlanForwardingType
BSW Parameter	**BSW Type**
ETHSWT_NOT_SENT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The message with the specific VLAN Id shall not be sent at the referenced port.	
Template Description	
will not be sent	

▽

△

M2 Parameter	
EthernetSwitchVlanEgressTaggingEnum.notSent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership/EthSwtVlanMembershipPortRefEntry/EthSwtVlanForwardingType
BSW Parameter	**BSW Type**
ETHSWT_SENT_TAGGED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The message with the specific VLAN Id shall be sent with its VLAN Id at the referenced port.	
Template Description	
sent with its VLAN tag	
M2 Parameter	
EthernetSwitchVlanEgressTaggingEnum.sentTagged	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership/EthSwtVlanMembershipPortRefEntry/EthSwtVlanForwardingType
BSW Parameter	**BSW Type**
ETHSWT_SENT_UNTAGGED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The message with the specific VLAN Id shall be sent untagged at the referenced port.	
Template Description	
sent without a VLAN tag	
M2 Parameter	
EthernetSwitchVlanEgressTaggingEnum.sentUntagged	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthSwt	EthSwt/EthSwtConfig/EthSwtVlanMembership/EthSwtVlanMembershipPortRefEntry
BSW Parameter	**BSW Type**
EthSwtVlanMembershipPortRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to one port the VLAN shall be assigned to.	

▽

△

Template Description	
M2 Parameter	
VlanMembership.vlan	
Mapping Rule	**Mapping Type**
If a CouplingPort has a CouplingPort.vlanMembership defined, then the EthSwtVlanMembership PortRefEntry created for that specific VlanConfig.vlanIdentifier shall have an EthSwtVlan MembershipPortRef reference to the EthSwtPort container which was created for that Coupling Port.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthSwt_ 00204]

C.22 EthTSyn

BSW Module	BSW Context
EthTSyn	EthTSyn
BSW Parameter	**BSW Type**
EthTSynGlobalTimeDomain	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This represents a global time domain on Ethernet. The EthTSyn module can manage several global time domains. Only one container shall be configured per global time domain ID. If the EthTSyn exists it is assumed that at least one global time domain exists.	
Template Description	
This represents the ability to define a global time domain.	
M2 Parameter	
GlobalTimeDomain	
Mapping Rule	**Mapping Type**
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the EcucInstance for which the ECU configuration is created. Additionally, in case of an Ethernet Switch: The container shall exist if at least one EthGlobalTimeManagedCouplingPort is defined in the scope of the GlobalTimeDomain.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_ 00004]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
EthTSynGlobalTimeDebounceTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents the configuration of a TX debounce time for Sync, Follow_Up, and pDelay messages compared to a message before with the same PDU. Unit: seconds	
Template Description	
Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	

▽

△

GlobalTimeDomain.debounceTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00048]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
EthTSynGlobalTimeDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The global time domain ID.	
Template Description	
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.	
M2 Parameter	
GlobalTimeDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00005]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
EthTSynGlobalTimeFollowUpDataIDList	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
EthGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00030]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynGlobalTimeFollowUpDataIDList
BSW Parameter	**BSW Type**
EthTSynGlobalTimeFollowUpDataIDListElement	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Element of the DataIDList for Follow_Up message ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for FUP messages to calculate CRC.	
M2 Parameter	
EthGlobalTimeDomainProps.fupDataIDList	
Mapping Rule	**Mapping Type**
Value shall be derived from element of the ordered fupDataIDList.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00031]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeDomain/EthTSynPTPPort
BSW Parameter	**BSW Type**
EthTSynPortRole	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Parameter to set the port's role to TimeSlave, TimeMaster, or Dynamic (i.e., TimeSlave or TimeMaster is determined at runtime).	
Template Description	
This attribute defines the port behavior.	
M2 Parameter	
EthGlobalTimeManagedCouplingPort.globalTimePortRole	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00116]

BSW Module	**BSW Context**
EthTSyn	EthTSyn
BSW Parameter	**BSW Type**
EthTSynGlobalTimeMasterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a global-time master. The exact role of the port is defined by EthTSynPortRole. A global-time master may or may not exist on the configured ECU. Each global-time domain may have multiple ports in the time master role (see Time Gateway) to e.g. relay global-time from a global time slave port.	
Template Description	
This represents the generic concept of a global time master.	
M2 Parameter	
GlobalTimeMaster	
Mapping Rule	**Mapping Type**
The existence of a EthTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00008]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcCorrectionField	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
correctionField from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcCorrectionField	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00042]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcDomainNumber	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
domainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcDomainNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00041]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcMessageLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
messageLength from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
MessageLength from the Follow_Up Message Header shall be included in CRC calculation.	

▽

△

M2 Parameter	
EthTSynCrcFlags.crcMessageLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00040]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcPreciseOriginTimestamp	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.	
Template Description	
PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcPreciseOriginTimestamp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00045]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcSequenceId	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
sequenceId from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
SequenceId from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcSequenceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00044]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynCrcTimeFlagsTxSecured
BSW Parameter	**BSW Type**
EthTSynCrcSourcePortIdentity	ECUC-BOOLEAN-PARAM-DEF

▽

△

BSW Description	
sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcSourcePortIdentity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00043]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynCyclicMsgResumeTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds	
Template Description	
Defines the minimum time between an "immediate" message and the next periodic message.	
M2 Parameter	
GlobalTimeMaster.immediateResumeTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00047]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynGlobalTimeTxCrcSecured	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This represents the configuration of whether or not CRC is supported.	
Template Description	
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.	
M2 Parameter	
GlobalTimeEthMaster.crcSecured	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00039]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynGlobalTimeTxCrcSecured	
BSW Parameter	**BSW Type**	
CRC_NOT_SUPPORTED	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This represents a configuration where CRC is not supported.		
Template Description		
This indicates that CRC is not supported		
M2 Parameter		
GlobalTimeCrcSupportEnum.crcNotSupported		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig/EthTSynGlobalTimeTxCrcSecured	
BSW Parameter	**BSW Type**	
CRC_SUPPORTED	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This represents a configuration where CRC is supported.		
Template Description		
This indicates that CRC is supported		
M2 Parameter		
GlobalTimeCrcSupportEnum.crcSupported		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig	
BSW Parameter	**BSW Type**	
EthTSynGlobalTimeTxPeriod	ECUC-FLOAT-PARAM-DEF	
BSW Description		
This represents configuration of the TX period. Unit: seconds		
Template Description		
This attribute defines the TX period in seconds		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.globalTimeTxPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_ 00010]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynHoldOverTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Parameter to define timeout for transmission of Sync and Follow_Up messages on Master ports in absence of reception of Sync and Follow_Up messages on Slave port.Unit: seconds	
Template Description	
This attribute defines the timeout for transmission of Sync and Follow_Up messages on Master ports in absence of reception of Sync and Follow_Up messages on Slave port.	
M2 Parameter	
GlobalTimeEthMaster.holdOverTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00115]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynTxSubTLVStatus	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Definition of whether (true) or not (false) a Sub-TLV:Status Secured or Sub-TLV:Status Not Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.	
Template Description	
Defines whether an AUTOSAR Follow_Up TLV Status Sub-TLV is used.	
M2 Parameter	
EthTSynSubTlvConfig.statusSubTlv	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00036]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynTxSubTLVTime	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Definition of whether (true) or not (false) a Sub-TLV:Time Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.	
Template Description	
Defines whether an AUTOSAR Follow_Up TLV Time Sub-TLV is used.	
M2 Parameter	
EthTSynSubTlvConfig.timeSubTlv	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00035]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeMasterConfig
BSW Parameter	**BSW Type**
EthTSynTxSubTLVUserData	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Definition of whether (true) or not (false) a Sub-TLV:UserData Secured or Sub-TLV:UserData Not Secured shall be sent in the AUTOSAR TLV of a Follow_Up message.	
Template Description	
Defines whether an AUTOSAR Follow_Up TLV UserData Sub-TLV is used.	
M2 Parameter	
EthTSynSubTlvConfig.userDataSubTlv	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00037]

BSW Module	BSW Context
EthTSyn	EthTSyn
BSW Parameter	**BSW Type**
EthTSynGlobalTimeSlaveConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a global-time slave. The exact role of the port is defined by EthTSynPortRole. A TIME_SLAVE port will exist on every configured ECU, except potentially on the SystemWideGlobalTimeMaster. Each global-time domain should have at least one port of one ECU in the TIME_MASTER role and one port of another ECU in the TIME_SLAVE role.	
Template Description	
This represents the generic concept of a global time slave.	
M2 Parameter	
GlobalTimeSlave	
Mapping Rule	**Mapping Type**
The existence of the EthTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00009]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated
BSW Parameter	**BSW Type**
EthTSynCrcCorrectionField	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
correctionField from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
CorrectionField from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcCorrectionField	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00042]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated
BSW Parameter	**BSW Type**
EthTSynCrcDomainNumber	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
domainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
DomainNumber from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcDomainNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00041]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated
BSW Parameter	**BSW Type**
EthTSynCrcMessageLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
messageLength from the Follow_Up Message Header shall be included in CRC calculation.	
Template Description	
MessageLength from the Follow_Up Message Header shall be included in CRC calculation.	
M2 Parameter	
EthTSynCrcFlags.crcMessageLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00040]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated	
BSW Parameter		**BSW Type**
EthTSynCrcPreciseOriginTimestamp		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
preciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
Template Description		
PreciseOriginTimestamp from the Follow_Up Message Field shall be included in CRC calculation.		
M2 Parameter		
EthTSynCrcFlags.crcPreciseOriginTimestamp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00045]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated	
BSW Parameter		**BSW Type**
EthTSynCrcSequenceId		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
sequenceId from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
SequenceId from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
EthTSynCrcFlags.crcSequenceId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00044]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynCrcFlagsRxValidated	
BSW Parameter		**BSW Type**
EthTSynCrcSourcePortIdentity		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
sourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
Template Description		
SourcePortIdentity from the Follow_Up Message Header shall be included in CRC calculation.		
M2 Parameter		
EthTSynCrcFlags.crcSourcePortIdentity		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00043]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig	
BSW Parameter		**BSW Type**
EthTSynGlobalTimeFollowUpTimeout		ECUC-FLOAT-PARAM-DEF
BSW Description		
Timeout value of the Follow_Up message (of the subsequent Sync message). A value of 0 deactivates this timeout observation. Unit: seconds		
Template Description		
Rx timeout for the follow-up message.		
M2 Parameter		
GlobalTimeSlave.followUpTimeoutValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00007]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig	
BSW Parameter		**BSW Type**
EthTSynRxCrcValidated		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Definition of whether or not validation of the CRC takes place.		
Template Description		
Definition of whether or not validation of the CRC is supported.		
M2 Parameter		
GlobalTimeEthSlave.crcValidated		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00049]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxCrcValidated	
BSW Parameter		**BSW Type**
CRC_IGNORED		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
EthTSyn ignores any CRC inside the Sub-TLVs.		
Template Description		
The CRC is supposed to be ignored		
M2 Parameter		
GlobalTimeCrcValidationEnum.crcIgnored		
Mapping Rule		**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_NOT_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60.	
Template Description	
The CRC is not supposed to be present. If CRC is present the message is ignored.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcNotValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_OPTIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60, that contain an incorrect CRC value.	
Template Description	
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcOptional	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
If EthTSynMessageCompliance is set to FALSE: EthTSyn discards Follow_Up messages with Sub-TLVs of Type 0x28, 0x50 or 0x60, that contain an incorrect CRC value. EthTSyn rejects Follow_Up messages with Sub-TLVs of Type 0x51 or 0x61.	
Template Description	
This CRC is supposed to be validated.	

▽

△

M2 Parameter	
GlobalTimeCrcValidationEnum.crcValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig
BSW Parameter	**BSW Type**
EthTSynRxIcvVerificationType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter controls whether or not ICV verification shall be supported.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00103]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxIcvVerificationType
BSW Parameter	**BSW Type**
ICV_IGNORED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module shall not verify the ICV.	
Template Description	
If the ICV is present, then it is ignored	
M2 Parameter	
GlobalTimeIcvVerificationEnum.icvIgnored	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxIcvVerificationType
BSW Parameter	**BSW Type**
ICV_NOT_VERIFIED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are not ICV secured. All other Time Synchronization messages are ignored.	

▽

△

Template Description	
The ICV is not supposed to be present. If the ICV is present, then the message is ignored.	
M2 Parameter	
GlobalTimeIcvVerificationEnum.icvNotVerified	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxIcvVerificationType
BSW Parameter	**BSW Type**
ICV_OPTIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages which are not ICV secured and Time Synchronization messages which are ICV secured and have the correct ICV. All other Time Synchronization messages are ignored.	
Template Description	
If the ICV is present, then it will be verified. If the ICV is not present, then this is also a valid reception (no verification required).	
M2 Parameter	
GlobalTimeIcvVerificationEnum.icvOptional	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynGlobalTimeSlaveConfig/EthTSynRxIcvVerificationType
BSW Parameter	**BSW Type**
ICV_VERIFIED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are ICV secured and have the correct ICV. All other Time Synchronization messages are ignored.	
Template Description	
The ICV is required and will be verified.	
M2 Parameter	
GlobalTimeIcvVerificationEnum.icvVerified	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTSyn	EthTSyn
BSW Parameter	**BSW Type**

▽

△

EthTSynPTPPortDetails	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a PTP Port. The same configuration can be used by multiple PTP ports through referencing this configuration.	
Template Description	
Specifies a CouplingPort which is managed by an Ethernet Global Time Domain.	
M2 Parameter	
EthGlobalTimeManagedCouplingPort	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00145]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynPTPPortDetails
BSW Parameter	**BSW Type**
EthTSynDestPhyAddr	ECUC-STRING-PARAM-DEF
BSW Description	
Destination Physical Address (MAC-Address). Destination Physical Hardware Address (MAC-Address) of EthTSyn-gPTP Frames. Input format has to match xx:xx:xx:xx:xx:xx, where x stands for a hex value between 0 and F.	
Template Description	
Defines the MAC multicast address the Ethernet time sync messages are communicated on.	
M2 Parameter	
EthGlobalTimeDomainProps.destinationPhysicalAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00058]

BSW Module	**BSW Context**
EthTSyn	EthTSyn/EthTSynPTPPortDetails
BSW Parameter	**BSW Type**
EthTSynFramePrio	ECUC-INTEGER-PARAM-DEF
BSW Description	
This optional parameter, if present, indicates the priority of outgoing EthTSyn message (i.e., it equals the 3-bit PCP field of a tagged VLAN message). If a VLAN is not configured, this parameter is also not configured.	
Template Description	
Defines which VLAN priority shall be assigned to a time sync message in case the message is sent using a VLAN tag.	
M2 Parameter	
EthGlobalTimeDomainProps.vlanPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_EthTSyn_00034]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails
BSW Parameter	**BSW Type**
EthTSynGlobalTimeRxIcvVerification	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container collects configuration that shall be used for ICV verification.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
Create this container if GlobalTimeSlave.icvVerification is defined as the value GlobalTimeIcvVerificationEnum. icvVerified or GlobalTimeIcvVerificationEnum.icvOptional.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00104]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails/EthTSynGlobalTimeRxIcvVerification
BSW Parameter	**BSW Type**
EthTSynIcvVerificationAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the number of ICV verification attempts that are to be carried out when the verification of the ICV failed for a given secured Follow_Up message. If zero is set, then only one ICV verification attempt is done.	
Template Description	
This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given message. If zero is set than only one authentication attempt is done.	
M2 Parameter	
SecOcSecureComProps.authenticationVerifyAttempts	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00110]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails/EthTSynGlobalTimeRxIcvVerification
BSW Parameter	**BSW Type**
EthTSynIcvVerificationFvIdRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the FV taken to verify the ICV.	

▽

△

Template Description	
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeSlave.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00105]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails/EthTSynGlobalTimeRxIcvVerification
BSW Parameter	**BSW Type**
EthTSynIcvVerificationJobRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the CSM job to fetch the CSM job ID.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
If GlobalTimeSlave.icvVerification is set to the value icvVerified or icvOptional, then the referenced CsmJob needs to be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00108]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails
BSW Parameter	**BSW Type**
EthTSynGlobalTimeTxIcvGeneration	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container collects configuration that shall be used for ICV generation.	
Template Description	
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.	
M2 Parameter	
GlobalTimeMaster.icvSecured	
Mapping Rule	**Mapping Type**
Create this container if GlobalTimeMaster.icvSecured is defined as the value GlobalTimeIcvSupportEnum.icvSupported.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00096]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails/EthTSynGlobalTimeTxIcvGeneration
BSW Parameter	**BSW Type**
EthTSynIcvGenerationFvIdRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the FV taken to generate the ICV generation.	
Template Description	
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeMaster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00097]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails/EthTSynGlobalTimeTxIcvGeneration
BSW Parameter	**BSW Type**
EthTSynIcvGenerationJobRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the CSM job to fetch the CSM job ID.	
Template Description	
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.	
M2 Parameter	
GlobalTimeMaster.icvSecured	
Mapping Rule	**Mapping Type**
If GlobalTimeMaster.icvSecured is set to the value icvSupported, then the referenced CsmJob needs to be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00100]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPTPPortDetails
BSW Parameter	**BSW Type**
EthTSynMessageCompliance	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
• true: IEEE 802.1AS compliant message format will be used. • false: IEEE 802.1AS message format with AUTOSAR extension will be used.	
Template Description	
Defines the compliance of the Ethernet time sync messages to specific standards.	
M2 Parameter	
EthGlobalTimeDomainProps.messageCompliance	
Mapping Rule	**Mapping Type**

▽

△

If EthGlobalTimeDomainProps.messageFormat = IEEE802_1AS then EthTSynMessage Compliance shall be true. If EthGlobalTimeDomainProps.messageFormat = IEEE802_1AS_AUTOSAR then EthTSyn MessageCompliance shall be false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_ 00029]

BSW Module	**BSW Context**	
EthTSyn	EthTSyn/EthTSynPdelayConfig	
BSW Parameter		**BSW Type**
EthTSynGlobalTimePdelayRespEnable		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Determines if a Pdelay Response message is transmitted upon reception of a Pdelay Request message.		
Template Description		
This attribute defines the port behavior.		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.globalTimePortRole		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_ 00129]

BSW Module	**BSW Context**	
EthTSyn	EthTSyn/EthTSynPdelayConfig	
BSW Parameter		**BSW Type**
EthTSynGlobalTimePropagationDelay		ECUC-FLOAT-PARAM-DEF
BSW Description		
If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter replaces a measured propagation delay by a fixed value. Unit: seconds		
Template Description		
If cyclic propagation delay measurement is enabled, this parameter represents the default value of the propagation delay until the first actually measured propagation delay is available. If cyclic propagation delay measurement is disabled, this parameter defines a fixed value for the propagation delay.		
M2 Parameter		
GlobalTimeCouplingPortProps.propagationDelay		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_ 00070]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynPdelayConfig	
BSW Parameter		**BSW Type**
EthTSynGlobalTimeTxPdelayReqEnable		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Determines if Pdelay Request messages are transmitted.		
Template Description		
This attribute defines the port behavior.		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.globalTimePortRole		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00144]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynPdelayConfig	
BSW Parameter		**BSW Type**
EthTSynGlobalTimeTxPdelayReqPeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
This represents configuration of the TX period for Pdelay_Req messages. A value of 0 disables the cyclic Pdelay measurement. Unit: seconds		
Template Description		
Defines the period for the pdelay request messages.		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00071]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynPdelayConfig	
BSW Parameter		**BSW Type**
EthTSynPdelayLatencyThreshold		ECUC-FLOAT-PARAM-DEF
BSW Description		
Threshold for calculated Pdelay. If a measured Pdelay exceeds EthTSynPdelayLatencyThreshold, this value is discarded. Unit: seconds		
Template Description		
Threshold for calculated Pdelay. If a measured Pdelay exceeds pdelayLatencyThreshold, the measured Pdelay value is discarded.		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.pdelayLatencyThreshold		



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00076]

BSW Module	BSW Context
EthTSyn	EthTSyn/EthTSynPdelayConfig
BSW Parameter	**BSW Type**
EthTSynPdelayRespAndRespFollowUpTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 deactivates this timeout observation. Unit: seconds	
Template Description	
Timeout value for Pdelay_Resp and Pdelay_Resp_Follow_Up after a Pdelay_Req has been transmitted resp. a Pdelay_Resp has been received. A value of 0 or not defining this attribute deactivates this timeout observation.	
M2 Parameter	
EthGlobalTimeManagedCouplingPort.pdelayRespAndRespFollowUpTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00074]

BSW Module	BSW Context
EthTSyn	EthTSyn
BSW Parameter	**BSW Type**
EthTSynPhysicalPortConfiguration	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a physical port. The same configuration can be used by multiple physical ports through referencing this configuration.	
Template Description	
Defines which CouplingPort is managed by this EthGlobalTimeManagedCouplingPort.	
M2 Parameter	
EthGlobalTimeManagedCouplingPort.couplingPort	
Mapping Rule	**Mapping Type**
Create one entry for each referenced CouplingPort.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTSyn_00139]

BSW Module	BSW Context	
EthTSyn	EthTSyn/EthTSynPhysicalPortConfiguration	
BSW Parameter		**BSW Type**
EthTSynSwitchManagementEthSwitchPortRef		ECUC-REFERENCE-DEF
BSW Description		
In an AVB-Bridge config, this reference is used to assign the EthTSynPTPPort to an Ethernet Switch-Port.		
Template Description		
Defines which CouplingPort is managed by this EthGlobalTimeManagedCouplingPort.		
M2 Parameter		
EthGlobalTimeManagedCouplingPort.couplingPort		
Mapping Rule		**Mapping Type**
If the referenced CouplingPort is aggregated by a CouplingElement with couplingType = switch then the reference EthTSynSwitchManagementEthSwitchPortRef shall be defined and refer to the EthSwtPort which was derived from the CouplingPort.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTSyn_00066]

C.23 EthTrcv

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		**BSW Type**
EthTrcvActAsSlavePassiveEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Specifies if the ECU is acting as a passive communication slave on the corresponding ComM channel (corresponding ComM channel has ComMNMVariant set to SLAVE_PASSIVE). If the parameter is set to TRUE, the Ethernet transceiver driver shall poll the maintained Ethernet hardware for a signaled Sleep.Indication (according to OA TC10) in the context of the EthTrcv_MainFunction.		
Template Description		
This attribute specifies if the EcuInstance is acting as a passive communication slave on the connected PhysicalChannel. This is used for EthernetCommunicationControllers that use Ethernet hardware which supports wake-up and sleep on the network (e.g. Open Alliance TC10 compliant Ethernet hardware).		
M2 Parameter		
EthernetCommunicationController.slaveActAsPassiveCommunicationSlave		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_EthTrcv_00071]

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		**BSW Type**
EthTrcvConnNeg		ECUC-ENUMERATION-PARAM-DEF

▽

△

BSW Description	
Specifies the connection negotiation of the Ethernet transceiver link.	
Template Description	
Specifies the connection negotiation of the CouplingPort.	
M2 Parameter	
CouplingPort.connectionNegotiationBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00025]

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg	
BSW Parameter	**BSW Type**	
TRCV_CONN_NEG_AUTO	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Automatic Negotiation		
Template Description		
Automatic Negotiation		
M2 Parameter		
EthernetConnectionNegotiationEnum.auto		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg	
BSW Parameter	**BSW Type**	
TRCV_CONN_NEG_MASTER	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Master		
Template Description		
Master		
M2 Parameter		
EthernetConnectionNegotiationEnum.master		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg	
BSW Parameter	**BSW Type**	
TRCV_CONN_NEG_NONE	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
PLCA		
Template Description		
Master		
M2 Parameter		
EthernetConnectionNegotiationEnum.master		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvConnNeg	
BSW Parameter	**BSW Type**	
TRCV_CONN_NEG_SLAVE	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Slave		
Template Description		
Slave		
M2 Parameter		
EthernetConnectionNegotiationEnum.slave		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter	**BSW Type**	
EthTrcvMacLayerType	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Defines the MAC layer type of the ethernet transceiver.		
Template Description		
Specifies the mac layer type of the CouplingPort.		
M2 Parameter		
CouplingPort.macLayerType		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_EthTrcv_00035]

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	
BSW Parameter	**BSW Type**	
TRCV_MAC_LAYER_TYPE_XGMII	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
MAC layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)		
Template Description		
Mac layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII)		
M2 Parameter		
EthernetMacLayerTypeEnum.xGMII		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	
BSW Parameter	**BSW Type**	
TRCV_MAC_LAYER_TYPE_XMII	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
MAC layer interface (data) bandwith class 10-100Mbit/s (e.g. RMII, RvMII, SMII, MII)		
Template Description		
Mac layer interface (data) bandwith class 100Mbit/s and 10Mbit/s (e.g. RMII, RvMII, SMII, RvMII)		
M2 Parameter		
EthernetMacLayerTypeEnum.xMII		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvMacLayerType	
BSW Parameter	**BSW Type**	
TRCV_MAC_LAYER_TYPE_XXGMII	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
MAC layer interface (data) bandwith class 10Gbit/s		
Template Description		
Mac layer interface (data) bandwith class 10Gbit/s		
M2 Parameter		
EthernetMacLayerTypeEnum.xXGMII		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvMacSecBypassDestinationMacAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Bypassed destination MAC address (48-bit physical address). Ethernet frames, which carrying the configured destination MAC address, shall bypass the MACsec protection handling.	
Template Description	
The attribute is used to define MAC destination addresses that are bypassed by MACsec. The provided MAC addresses will not be MACsec protected. Please note, IEEE 802.1X-2020 chapter 11.1.1 (Table 11-1) define specific MAC destination addresses to be bypassed.	
M2 Parameter	
MacSecGlobalKayProps.bypassDestinationMacAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00080]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvMacSecBypassEtherType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed EtherType. Ethernet frames, which carrying the configured EtherType, shall bypass the MACsec protection handling.	
Template Description	
This attribute is used to define EtherTypes that are bypassed by MACsec. The providedEtherType will not be MACsec protected.	
M2 Parameter	
MacSecGlobalKayProps.bypassEtherType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00077]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvMacSecBypassVlan	ECUC-INTEGER-PARAM-DEF
BSW Description	
Bypassed VLAN-ID. Ethernet frames, which carrying the configured VLAN-ID, shall bypass the MACsec protection handling. Please note: VLAN-ID 0 is interpreted as no VLAN, i.e. untagged Ethernet frames bypass the MACsec protection handling.	
Template Description	
This attribute is used to define VLAN-IDs that are bypassed by MACsec. The provided VLAN-IDs will not be MACsec protected. (VLAN-ID 0 is interpreted as no-VLAN --> Bypass untagged traffic)	



△

M2 Parameter	
MacSecGlobalKayProps.bypassVlan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00078]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvPhysLayerType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies the physical layer type of the Ethernet transceiver link.	
Template Description	
Specifies the physical layer type of the CouplingPort.	
M2 Parameter	
CouplingPort.physicalLayerType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00024]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_10000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 10GBASE-T1 (10Gbit/s, 1pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_1000BASE_T	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

physical layer interface 1000BASE-T (1Gbit/s, 4 pairs). Used for consumer electronic.	
Template Description	
Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_1000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 1000BASE-T1 (1Gbit/s, 1 pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_100BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 100BASE-T1 (100Mbit/s, 1 pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100BASE-T1 is the IEEE Standardized version of BroadRReach.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.100BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**

▽

△

TRCV_PHYS_LAYER_TYPE_100BASE_TX	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 100BASE-TX (100Mbit/s, 2 pairs). Used for consumer electronic.	
Template Description	
Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.100BASE-TX	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_10BASE_T1S	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Physical layer interface 10BASE-T1S (10Mbit/s, 1 pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_2500BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 2.5GBASE-T1 (2.5Gbit/s, 1pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvPhysLayerType
BSW Parameter	**BSW Type**
TRCV_PHYS_LAYER_TYPE_5000BASE_T1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
physical layer interface 5GBASE-T1 (5Gbit/s, 1pair). Used for automotive.	
Template Description	
Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable.	
M2 Parameter	
EthernetPhysicalLayerTypeEnum.1000BASE-T1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvSleepModeExecutionDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the time delay in seconds to execute a sleep (see OA TC10) for a Ethernet hardware (PHY), if a pending wake-up was detected while a ETH_MODE_DOWN was requested. The value shall be an integral multiple of EthTrcvMainFunctionPeriod.	
Template Description	
Delay in seconds to perform a sleep request if the Ethernet hardware (PHY) detect a pending wake-up. This is used to avoid the race condition, if a sleep was requested while a wake-up of a neighboring PHY was received via a local wake-up connection (e.g. I/O pin).	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.sleepModeExecutionDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00063]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvSleepRequestNumberOfRepetitions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the repetitions to trigger a Sleep.Request (according to OA TC10) if a release of the communication channel was triggered by the upper layer (ETH_MODE_DOWN) and a SleepFail.Indication was signaled. Thus, the Ethernet hardware (PHY) of the connected communication partner did not accept the Sleep.Request.	
Template Description	
Count of repetitions for a sleep on dataline. If a sleep is rejected by the linked communication partner, the sleep is repeated until the count of repetitions exceed. If count of repetitions exceed, the Ethernet hardware (PHY) transit to sleep without acknowledgement of the connected link partner.	

▽

△

M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.sleepRepetitionsOfSleepRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00072]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvSleepRequestRepetitionPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the repetition period in seconds of repetitions for a Sleep Request (according to OA TC10). The value shall be a integral multiple of EthTrcvMainFunctionPeriod.	
Template Description	
Delay in seconds for a repetition of a sleep request. This is used to retry a synchronized shutdown of the connected Ethernet hardware (PHY) of the link partner.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.sleepRepetitionDelayOfSleepRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00073]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeUpRequestRepetitionPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the repetition period in seconds of a wake-up request, if an active communication request (ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST) was triggered by the upperlayer. The value shall be an integral multiple of EthTrcvMainFunctionPeriod.	
Template Description	
Delay in seconds for a repetition of a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupRepetitionDelayOfWakeupRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00062]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupForwardLocalEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if remote wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver. If the parameter is set to TRUE, the Ethernet hardware (PHY) activate a local wake up (e.g. via I/O pin), if a remote wake-up on data line (e.g. 100Base-T1) was received (either WUP or WUR).	
Template Description	
If enabled, then a remote wake-up received on the physical dataline (e.g. 100BASE-T1) is forwarded as local wake-up (e.g. via an I/O pin). If disabled, then a remote wake-up is not forwarded as local wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupForwardLocalEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00067]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupForwardRemoteEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if local wake up forwarding is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet Transceiver. If the parameter is set to TRUE, the Ethernet hardware (PHY) transmit a wake-up (WUP or WUR on the data line (e.g. 100 Base-T1), when a local wake-up occurred (e.g. via I/O pin), which was triggered by neighboring Ethernet hardware (PHY).	
Template Description	
If enabled, then a local wake-up is forwarded to the physical dataline (e.g. 100BASE-T1). If disabled, then a local wake-up is not forwarded to the physical dataline.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupForwardRemoteEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00068]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupLocalDetectionTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time in seconds when a local wake-up (e.g. via I/O pin) triggered by a neighboring PHY is evalutated as a valid wake-up.	
Template Description	

▽

△

Specify the detection time if a local wake-up in seconds is present on the local wake-up connection (e.g. I/O pin). A local wake-up has to be present at least for wakeupLocalDetectionTime to be detected a valid local wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupLocalDetectionTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00069]

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupLocalDurationTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the duration time in seconds how long a local wake-up should be present on the local wake-up connection (e.g. via I/O pin) to indicate the neighboring PHYs about a wake-up.	
Template Description	
Specify the duration of a local wake-up in seconds to be present on the local wake-up connection (e.g. I/O pin).	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupLocalDurationTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00070]

BSW Module	**BSW Context**
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupLocalEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if local wake-up is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet hardware (PHY). If the parameter is set to TRUE, the Ethernet hardware (PHY) is able to detect and react on a wake-up received by a neighboring PHY (e.g. via I/O pin).	
Template Description	
If enabled, then a local wake-up received via a local connection (e.g. I/O pin) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reacting on a local wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupLocalEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00065]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupRemoteEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if remote wake-up is enabled (TRUE) or disabled (FALSE) for OA TC10 compliant Ethernet hardware (PHY). If the parameter is set to TRUE, the Ethernet hardware (PHY) wake up when receiving a remote wake-up (e.g. via 100 Base-T1 data line) triggered by the connected communication partner.	
Template Description	
If enabled, then a remote wake-up received via the physical dataline (e.g. 100BASE-T1) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reaction on a remote wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupRemoteEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00066]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupRequestNumberOfRepetitions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the repetitions to trigger a wake-up request (according to OA TC10), if an active communication request (ETH_MODE_ACTIVE_WITH_WAKEUP_REQUEST) was triggered by the upperlayer.	
Template Description	
Count of repetitions for a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.	
M2 Parameter	
EthernetWakeupSleepOnDatalineConfig.wakeupRepetitionsOfWakeupRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00075]

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig
BSW Parameter	**BSW Type**
EthTrcvWakeupSleepOnDatalineEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Specifies if wake-up on data line according to OA TC10 is supported by the used Ethernet hardware (PHY) and if the functionality (wake-up and sleep on dataline) is enabled. If the parameter is configured, wake-up and sleep on data line is supported according to OA TC10 by the used Ethernet hardware (PHY). If the parameter is set to TRUE, the wake-up and sleep on data line functionality according to OA TC10 is enabled. If the parameter is set to FALSE, the wake-up and sleep on data line functionality according to OA TC10 is disabled. Note: Disabling of the wake-up and sleep functionality is used if Ethernet hardware (PHY) parts are connected, where one PHY do not support the OA TC10 wake-up and sleep on dataline functionality. This may needed for legacy scenarios, where ECUs are taken over from previous car lines and no OA TC10 Ethernet hardware (PHY) were used.
Template Description
Optional reference to EthernetWakeupSleepOnDatalineConfig.
M2 Parameter
CouplingPort.wakeupSleepOnDatalineConfig

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_EthTrcv_00064]

C.24 Firewall

BSW Module	**BSW Context**
Firewall	Firewall

BSW Parameter	**BSW Type**
FirewallRule	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Firewall Rule that defines the control information in individual packets.	
Template Description	
Firewall Rule that defines the control information in individual packets.	
M2 Parameter	
FirewallRule	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00011]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule

BSW Parameter	**BSW Type**
FirewallBucketSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
This attribute defines the capacity of the queue for rate limitation (leaky-bucket Algorithm).	
Template Description	
This attribute defines the capacity of the queue for rate limitation (leaky-bucket Algorithm).	

▽

△

M2 Parameter	
FirewallRule.bucketSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00027]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule	
BSW Parameter		**BSW Type**
FirewallDataLinkFilterConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of filter rules on the DataLink layer		
Template Description		
Configuration of filter rules on the DataLink layer		
M2 Parameter		
DataLinkLayerRule		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00139]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig	
BSW Parameter		**BSW Type**
FirewallFilterEtherType		ECUC-INTEGER-PARAM-DEF
BSW Description		
Definition of the filter Ether Type.		
Template Description		
Filter to match packets based on the EtherType field in the Ethernet frame. The EtherType is used to indicate which protocol is encapsulated in the payload of the frame.		
M2 Parameter		
DataLinkLayerRule.etherType		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00017]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig	
BSW Parameter		**BSW Type**
FirewallFilterMACDestAddress		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of one MAC destination filter.		

▽

△

Template Description	
DataLinkLayerRule.destinationMacAddress: Filter to match packets with the destination MAC address. **DataLinkLayerRule.destinationMacAddressMask:** Filter to match packets with the destination MAC address range. The destinationMacAddress with the destinationMacAddress Mask defines the MAC address range.	
M2 Parameter	
DataLinkLayerRule.destinationMacAddress, DataLinkLayerRule.destinationMacAddressMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00013]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig/FirewallFilterMACDestAddress
BSW Parameter	**BSW Type**
FirewallFilterMACAddressKey	ECUC-STRING-PARAM-DEF
BSW Description	
Specifies the 48-bit physical address (MAC address) key value.	
Template Description	
DataLinkLayerRule.sourceMacAddress: Filter to match packets with the source MAC address. **DataLinkLayerRule.destinationMacAddress:** Filter to match packets with the destination MAC address.	
M2 Parameter	
DataLinkLayerRule.sourceMacAddress, DataLinkLayerRule.destinationMacAddress	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00014]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig/FirewallFilterMACDestAddress
BSW Parameter	**BSW Type**
FirewallFilterMACAddressMask	ECUC-STRING-PARAM-DEF
BSW Description	
Specifies the 48-bit physical address (MAC address) mask value.	
Template Description	
DataLinkLayerRule.sourceMacAddressMask: Filter to match packets with the source MAC address range. The sourceMacAddress with the sourceMacAddressMask defines the MAC address range. **DataLinkLayerRule.destinationMacAddressMask:** Filter to match packets with the destination MAC address range. The destinationMacAddress with the destinationMacAddress Mask defines the MAC address range.	
M2 Parameter	
DataLinkLayerRule.sourceMacAddressMask, DataLinkLayerRule.destinationMacAddressMask	
Mapping Rule	**Mapping Type**

▽

△

depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00015]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterMACSrcAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of one MAC source filter.	
Template Description	
DataLinkLayerRule.sourceMacAddress: Filter to match packets with the source MAC address. **DataLinkLayerRule.sourceMacAddressMask:** Filter to match packets with the source MAC address range. The sourceMacAddress with the sourceMacAddressMask defines the MAC address range.	
M2 Parameter	
DataLinkLayerRule.sourceMacAddress, DataLinkLayerRule.sourceMacAddressMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00012]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig/FirewallFilterMACSrcAddress
BSW Parameter	**BSW Type**
FirewallFilterMACAddressKey	ECUC-STRING-PARAM-DEF
BSW Description	
Specifies the 48-bit physical address (MAC address) key value.	
Template Description	
DataLinkLayerRule.sourceMacAddress: Filter to match packets with the source MAC address. **DataLinkLayerRule.destinationMacAddress:** Filter to match packets with the destination MAC address.	
M2 Parameter	
DataLinkLayerRule.sourceMacAddress, DataLinkLayerRule.destinationMacAddress	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00014]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig/FirewallFilterMACSrcAddress
BSW Parameter	**BSW Type**
FirewallFilterMACAddressMask	ECUC-STRING-PARAM-DEF

▽

△

BSW Description	
Specifies the 48-bit physical address (MAC address) mask value.	
Template Description	
DataLinkLayerRule.sourceMacAddressMask: Filter to match packets with the source MAC address range. The sourceMacAddress with the sourceMacAddressMask defines the MAC address range. **DataLinkLayerRule.destinationMacAddressMask:** Filter to match packets with the destination MAC address range. The destinationMacAddress with the destinationMacAddress Mask defines the MAC address range.	
M2 Parameter	
DataLinkLayerRule.sourceMacAddressMask, DataLinkLayerRule.destinationMacAddressMask	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00015]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterVlanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter VLAN ID.	
Template Description	
Filter of packets with a specific VlanId.	
M2 Parameter	
DataLinkLayerRule.vlanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00016]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDataLinkFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterVlanPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter VLAN Priority.	
Template Description	
Filter of packets with a specific Vlan priority.	
M2 Parameter	
DataLinkLayerRule.vlanPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00018]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule	
BSW Parameter		**BSW Type**
FirewallDdsFilterConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of filter rules for Dds		
Template Description		
Configuration of a DDS firewall rule		
M2 Parameter		
DdsRule		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00092]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig	
BSW Parameter		**BSW Type**
FirewallDdsAppId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Filter for DDSI-RTPS messages in which the appId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.		
Template Description		
Filter for DDSI-RTPS messages in which the appId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.		
M2 Parameter		
DdsRule.appId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00093]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig	
BSW Parameter		**BSW Type**
FirewallDdsHostId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Filter for DDSI-RTPS messages in which the hostId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.		
Template Description		
Filter for DDSI-RTPS messages in which the hostId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.		
M2 Parameter		
DdsRule.hostId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00094]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the instanceId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.	
Template Description	
Filter for DDSI-RTPS messages in which the instanceId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.	
M2 Parameter	
DdsRule.instanceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00095]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsMajorProtocolVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the majorProtocolVersion in the DDSI-RTPS header matches.	
Template Description	
Filter for DDSI-RTPS messages in which the majorProtocolVersion in the DDSI-RTPS header matches.	
M2 Parameter	
DdsRule.majorProtocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00096]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsMinorProtocolVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the minorProtocolVersion in the DDSI-RTPS header matches.	
Template Description	
Filter for DDSI-RTPS messages in which the minorProtocolVersion in the DDSI-RTPS header matches.	
M2 Parameter	
DdsRule.minorProtocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00097]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsProductId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the productId in the DDSI-RTPS header matches.	
Template Description	
Filter for DDSI-RTPS messages in which the productId in the DDSI-RTPS header matches.	
M2 Parameter	
DdsRule.productId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00098]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsReaderEntityId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the readerEntityID in a DDSI-RTPS submessage matches	
Template Description	
Filter for DDSI-RTPS messages in which the readerEntityID in a DDSI-RTPS submessage matches	
M2 Parameter	
DdsRule.readerEntityId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00099]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsSubmessageType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the allowed submessage type in the DDSI-RTPS message	
Template Description	
Defines the allowed submessage type in the DDSI-RTPS message	
M2 Parameter	
DdsRule.submessageType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00100]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsVendorId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the vendorId in the DDSI-RTPS header matches.	
Template Description	
Filter for DDSI-RTPS messages in which the vendorId in the DDSI-RTPS header matches.	
M2 Parameter	
DdsRule.vendorId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00101]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDdsFilterConfig
BSW Parameter	**BSW Type**
FirewallDdsWriterEntityId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for DDSI-RTPS messages in which the writerEntityID in a DDSI-RTPS submessage matches	
Template Description	
Filter for DDSI-RTPS messages in which the writerEntityID in a DDSI-RTPS submessage matches	
M2 Parameter	
DdsRule.writerEntityId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00102]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallDoipFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules for DoIP	
Template Description	
Configuration of a generic firewall rule	
M2 Parameter	
DoIpRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00079]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipDestAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a source port filter.	
Template Description	
DoIpRule.destinationMinAddress: Filter to match DoIP messages in which the destinationAddress is greater or equal than destinationMinAddress. **DoIpRule.destinationMaxAddress:** Filter to match DoIP messages in which the destinationAddress is smaller or equal than destinationMaxAddress.	
M2 Parameter	
DoIpRule.destinationMinAddress, DoIpRule.destinationMaxAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00082]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig/FirewallDoipDestAddress
BSW Parameter	**BSW Type**
FirewallDoipDestAddressLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the destinationAddress is greater or equal than FirwallDoipDestAddressLowerValue	
Template Description	
Filter to match DoIP messages in which the destinationAddress is greater or equal than destinationMinAddress.	
M2 Parameter	
DoIpRule.destinationMinAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00085]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig/FirewallDoipDestAddress
BSW Parameter	**BSW Type**
FirewallDoipDestAddressUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the destinationAddress is smaller or equal than FirewallDoipDestAddressUpperValue	
Template Description	
Filter to match DoIP messages in which the destinationAddress is smaller or equal than destinationMaxAddress.	
M2 Parameter	
DoIpRule.destinationMaxAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Fw_00086]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipInverseProtocolVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the inverseprotocolVersion in the DoIP header matches.	
Template Description	
Filter to match DoIP messages in which the inverseprotocolVersion in the DoIP header matches.	
M2 Parameter	
DoIpRule.inverseProtocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00088]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipPayloadLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the payloadLength in the DoIP header matches.	
Template Description	
Filter to match DoIP messages in which the payloadLength in the DoIP header matches.	
M2 Parameter	
DoIpRule.payloadLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00089]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipPayloadType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the payloadType in the DoIP header matches.	
Template Description	
Filter to match DoIP messages in which the payloadType in the DoIP header matches.	
M2 Parameter	
DoIpRule.payloadType	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00090]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipProtocolVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the protocolVersion in the DoIP header matches.	
Template Description	
Filter to match DoIP messages in which the protocolVersion in the DoIP header matches.	
M2 Parameter	
DoIpRule.protocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00087]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipSrcAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a source port filter.	
Template Description	
DoIpRule.sourceMinAddress: Filter to match DoIP messages in which the sourceAddress is greater or equal than sourceMinAddress.. **DoIpRule.sourceMaxAddress:** Filter to match DoIP messages in which the sourceAddress is smaller or equal than sourceMaxAddress.	
M2 Parameter	
DoIpRule.sourceMinAddress, DoIpRule.sourceMaxAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00081]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig/FirewallDoipSrcAddress
BSW Parameter	**BSW Type**
FirewallDoipSrcAddressLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages in which the sourceAddress is greater or equal than FirwallDoipDestAddressLowerValue	

▽

△

Template Description	
Filter to match DoIP messages in which the sourceAddress is greater or equal than sourceMinAddress..	
M2 Parameter	
DoIpRule.sourceMinAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00083]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig/FirewallDoipSrcAddress
BSW Parameter	**BSW Type**
FirewallDoipSrcAddressUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port upper value.	
Template Description	
Filter to match DoIP messages in which the sourceAddress is smaller or equal than sourceMaxAddress.	
M2 Parameter	
DoIpRule.sourceMaxAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00084]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallDoipFilterConfig
BSW Parameter	**BSW Type**
FirewallDoipUdsService	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match DoIP messages that contain the udsService.	
Template Description	
Filter to match DoIP messages that contain the udsService.	
M2 Parameter	
DoIpRule.udsService	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00091]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallNetworkLayerFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Configuration of filter rules on the Network layer	
Template Description	
Configuration of filter rules on the Network layer	
M2 Parameter	
NetworkLayerRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00030]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig
BSW Parameter	**BSW Type**
FirewallNetworkLayerIpv4FilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules for IPv6 on the Network layer	
Template Description	
Configuration of filter rules on IPv4 level.	
M2 Parameter	
Ipv4Rule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00140]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallChecksumVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether checksum verification is performed or not.	
Template Description	
TransportLayerRule.checksumVerification: Defines whether checksum verification is performed or not. **Ipv4Rule.checksumVerification:** Defines whether a Ipv4 header checksum verification is performed or not. **IcmpRule.checksumVerification:** Defines whether a Icmp header checksum verification is performed or not.	
M2 Parameter	
TransportLayerRule.checksumVerification, Ipv4Rule.checksumVerification, IcmpRule.checksumVerification	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00025]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig	
BSW Parameter		**BSW Type**
FirewallDifferentiatedServiceCodePoint		ECUC-INTEGER-PARAM-DEF
BSW Description		
Filter to match packets with a DSCP value.		
Template Description		
Filter to match packets with a DSCP value.		
M2 Parameter		
Ipv4Rule.differentiatedServiceCodePoint		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00040]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig	
BSW Parameter		**BSW Type**
FirewallDoNotFragment		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Filter to match packets that have the doNotFragment bit in the Header set.		
Template Description		
Filter to match packets that have the doNotFragment bit in the Header set.		
M2 Parameter		
Ipv4Rule.doNotFragment		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00041]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig	
BSW Parameter		**BSW Type**
FirewallExplicitCongestionNotification		ECUC-INTEGER-PARAM-DEF
BSW Description		
Filter to match packets with a ECN code point.		
Template Description		
Filter to match packets with a ECN code point.		
M2 Parameter		
Ipv4Rule.explicitCongestionNotification		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00045]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig

BSW Parameter	BSW Type
FirewallFilterIPDestAddress	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Configuration of one IP destination filter.

Template Description

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv4Rule.destinationNetworkMask:
Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

Ipv6Rule.destinationNetworkMask:
Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.

M2 Parameter

`Ipv4Rule.destinationIpAddress`, `Ipv4Rule.destinationNetworkMask`, `Ipv6Rule.destinationIpAddress`, `Ipv6Rule.destinationNetworkMask`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00032]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallFilterIPDestAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressKey	ECUC-STRING-PARAM-DEF

BSW Description

IP address key pattern.

Template Description

Ipv4Rule.sourceIpAddress:
Filter to match packets with the source IPv4 address.

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv6Rule.sourceIpAddress:
Filter to match packets with the source IPv6 address.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

M2 Parameter

`Ipv4Rule.sourceIpAddress`, `Ipv4Rule.destinationIpAddress`, `Ipv6Rule.sourceIpAddress`, `Ipv6Rule.destinationIpAddress`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00033]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallFilterIPDestAddress
BSW Parameter	**BSW Type**
FirewallFilterIPAddressMask	ECUC-STRING-PARAM-DEF
BSW Description	
IP address mask pattern.	
Template Description	
Ipv4Rule.sourceNetworkMask: Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv4Rule.destinationNetworkMask:** Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range. **Ipv6Rule.sourceNetworkMask:** Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv6Rule.destinationNetworkMask:** Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.	
M2 Parameter	
Ipv4Rule.sourceNetworkMask, Ipv4Rule.destinationNetworkMask, Ipv6Rule.sourceNetworkMask, Ipv6Rule.destinationNetworkMask	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00034]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallFilterIPSrcAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of one IP source filter.	
Template Description	
Ipv4Rule.sourceIpAddress: Filter to match packets with the source IPv4 address. **Ipv4Rule.sourceNetworkMask:** Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv6Rule.sourceIpAddress:** Filter to match packets with the source IPv6 address. **Ipv6Rule.sourceNetworkMask:** Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.	
M2 Parameter	
Ipv4Rule.sourceIpAddress, Ipv4Rule.sourceNetworkMask, Ipv6Rule.sourceIpAddress, Ipv6Rule.sourceNetworkMask	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00031]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallFilterIPSrcAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressKey	ECUC-STRING-PARAM-DEF

BSW Description

IP address key pattern.

Template Description

Ipv4Rule.sourceIpAddress:
Filter to match packets with the source IPv4 address.

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv6Rule.sourceIpAddress:
Filter to match packets with the source IPv6 address.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

M2 Parameter

`Ipv4Rule.sourceIpAddress`, `Ipv4Rule.destinationIpAddress`, `Ipv6Rule.sourceIpAddress`, `Ipv6Rule.destinationIpAddress`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00033]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallFilterIPSrcAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressMask	ECUC-STRING-PARAM-DEF

BSW Description

IP address mask pattern.

Template Description

Ipv4Rule.sourceNetworkMask:
Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.

Ipv4Rule.destinationNetworkMask:
Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range.

Ipv6Rule.sourceNetworkMask:
Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.

Ipv6Rule.destinationNetworkMask:
Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.

M2 Parameter

`Ipv4Rule.sourceNetworkMask`, `Ipv4Rule.destinationNetworkMask`, `Ipv6Rule.sourceNetworkMask`, `Ipv6Rule.destinationNetworkMask`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00034]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallFilterIPv4Ttl	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Filter to match packets with a ttl value (TimeToLive defines the lifetime of data on the network).	
Template Description	
Ipv4Rule.ttlMin: Filter to match packets with a minimum ttl value (TimeToLive defines the lifetime of data on the network). **Ipv4Rule.ttlMax:** Filter to match packets with a maximum ttl value (TimeToLive defines the lifetime of data on the network).	
M2 Parameter	
Ipv4Rule.ttlMin, Ipv4Rule.ttlMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00046]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/ FirewallFilterIPv4Ttl
BSW Parameter	**BSW Type**
FirewallIpv4TtlMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a max ttl value.	
Template Description	
Filter to match packets with a maximum ttl value (TimeToLive defines the lifetime of data on the network).	
M2 Parameter	
Ipv4Rule.ttlMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00047]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/ FirewallFilterIPv4Ttl
BSW Parameter	**BSW Type**
FirewallIpv4TtlMin	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a min ttl value.	
Template Description	
Filter to match packets with a minimum ttl value (TimeToLive defines the lifetime of data on the network).	
M2 Parameter	
Ipv4Rule.ttlMin	
Mapping Rule	**Mapping Type**

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00048]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallInternetHeaderLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a minimum ipv4 header length.	
Template Description	
Filter to match packets with a minimum ipv4 header length.	
M2 Parameter	
`Ipv4Rule.internetHeaderLength`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00042]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallIpProtocolNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a IP protocol number .	
Template Description	
Filter to match packets with a IP protocol number .	
M2 Parameter	
`Ipv4Rule.protocol`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00044]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallMoreFragments	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Filter to match packets that have the moreFragments flag in the Header set.	
Template Description	
Filter to match packets that have the moreFragments flag in the Header set.	
M2 Parameter	

▽

△

Ipv4Rule.moreFragments	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00043]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig
BSW Parameter	**BSW Type**
FirewallNetworkLayerIcmpConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules for ICMP (Internet Control Message Protocol).	
Template Description	
Ipv4Rule.icmpRule: Configuration of filter rules for ICMP (Internet Control Message Protocol). **Ipv6Rule.icmpRule:** Configuration of filter rules for ICMP (Internet Control Message Protocol).	
M2 Parameter	
Ipv4Rule.icmpRule, Ipv6Rule.icmpRule	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00130]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/ FirewallNetworkLayerIcmpConfig
BSW Parameter	**BSW Type**
FirewallChecksumVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether checksum verification is performed or not.	
Template Description	
TransportLayerRule.checksumVerification: Defines whether checksum verification is performed or not. **Ipv4Rule.checksumVerification:** Defines whether a Ipv4 header checksum verification is performed or not. **IcmpRule.checksumVerification:** Defines whether a Icmp header checksum verification is performed or not.	
M2 Parameter	
TransportLayerRule.checksumVerification, Ipv4Rule.checksumVerification, IcmpRule.checksumVerification	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00025]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallNetworkLayerIcmpConfig
BSW Parameter	**BSW Type**
FirewallIcmpCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with the Icmp code.	
Template Description	
Filter to match packets with the Icmp code.	
M2 Parameter	
IcmpRule.code	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00132]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv4FilterConfig/FirewallNetworkLayerIcmpConfig
BSW Parameter	**BSW Type**
FirewallIcmpType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with the Icmp type.	
Template Description	
Filter to match packets with the Icmp type.	
M2 Parameter	
IcmpRule.type	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00131]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig
BSW Parameter	**BSW Type**
FirewallNetworkLayerIpv6FilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules on the Network layer	
Template Description	
Configuration of filter rules on IPv6 level.	
M2 Parameter	
Ipv6Rule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00049]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig

BSW Parameter	BSW Type
FirewallFilterIPDestAddress	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Configuration of one IP destination filter.

Template Description

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv4Rule.destinationNetworkMask:
Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

Ipv6Rule.destinationNetworkMask:
Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.

M2 Parameter

`Ipv4Rule.destinationIpAddress`, `Ipv4Rule.destinationNetworkMask`, `Ipv6Rule.destinationIpAddress`, `Ipv6Rule.destinationNetworkMask`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00032]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallFilterIPDestAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressKey	ECUC-STRING-PARAM-DEF

BSW Description

IP address key pattern.

Template Description

Ipv4Rule.sourceIpAddress:
Filter to match packets with the source IPv4 address.

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv6Rule.sourceIpAddress:
Filter to match packets with the source IPv6 address.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

M2 Parameter

`Ipv4Rule.sourceIpAddress`, `Ipv4Rule.destinationIpAddress`, `Ipv6Rule.sourceIpAddress`, `Ipv6Rule.destinationIpAddress`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00033]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallFilterIPDestAddress
BSW Parameter	**BSW Type**
FirewallFilterIPAddressMask	ECUC-STRING-PARAM-DEF
BSW Description	
IP address mask pattern.	
Template Description	
Ipv4Rule.sourceNetworkMask: Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv4Rule.destinationNetworkMask:** Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range. **Ipv6Rule.sourceNetworkMask:** Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv6Rule.destinationNetworkMask:** Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.	
M2 Parameter	
Ipv4Rule.sourceNetworkMask, Ipv4Rule.destinationNetworkMask, Ipv6Rule.sourceNetworkMask, Ipv6Rule.destinationNetworkMask	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00034]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallFilterIPSrcAddress	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of one IP source filter.	
Template Description	
Ipv4Rule.sourceIpAddress: Filter to match packets with the source IPv4 address. **Ipv4Rule.sourceNetworkMask:** Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range. **Ipv6Rule.sourceIpAddress:** Filter to match packets with the source IPv6 address. **Ipv6Rule.sourceNetworkMask:** Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.	
M2 Parameter	
Ipv4Rule.sourceIpAddress, Ipv4Rule.sourceNetworkMask, Ipv6Rule.sourceIpAddress, Ipv6Rule.sourceNetworkMask	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00031]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallFilterIPSrcAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressKey	ECUC-STRING-PARAM-DEF

BSW Description

IP address key pattern.

Template Description

Ipv4Rule.sourceIpAddress:
Filter to match packets with the source IPv4 address.

Ipv4Rule.destinationIpAddress:
Filter to match packets with the destination IPv4 address.

Ipv6Rule.sourceIpAddress:
Filter to match packets with the source IPv6 address.

Ipv6Rule.destinationIpAddress:
Filter to match packets with the destination IPv6 address.

M2 Parameter

`Ipv4Rule.sourceIpAddress`, `Ipv4Rule.destinationIpAddress`, `Ipv6Rule.sourceIpAddress`, `Ipv6Rule.destinationIpAddress`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00033]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallFilterIPSrcAddress

BSW Parameter	BSW Type
FirewallFilterIPAddressMask	ECUC-STRING-PARAM-DEF

BSW Description

IP address mask pattern.

Template Description

Ipv4Rule.sourceNetworkMask:
Filter to match packets with the source IPv4 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.

Ipv4Rule.destinationNetworkMask:
Filter to match packets with the destination IPv4 address range. The destinationIpAddress with the destinationNetworkMask defines the IP address range.

Ipv6Rule.sourceNetworkMask:
Filter to match packets with the source IPv6 address range. The sourceIpAddress with the sourceNetworkMask defines the IP address range.

Ipv6Rule.destinationNetworkMask:
Filter to match packets with the destination IPv6 address range. The destinationIpAddress with the destinationNetworkMask defines the MAC address range.

M2 Parameter

`Ipv4Rule.sourceNetworkMask`, `Ipv4Rule.destinationNetworkMask`, `Ipv6Rule.sourceNetworkMask`, `Ipv6Rule.destinationNetworkMask`

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00034]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallITrafficClass	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a defined traffic class or priority.	
Template Description	
Filter to match packets with a defined traffic class or priority.	
M2 Parameter	
Ipv6Rule.trafficClass	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00056]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallIpFlowLabel	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a defined flow label.	
Template Description	
Filter to match packets with a defined flow label.	
M2 Parameter	
Ipv6Rule.flowLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00051]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallIpHopLimit	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a minimum hop limit.	
Template Description	
Filter to match packets with a minimum hop limit.	
M2 Parameter	
Ipv6Rule.hopLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00052]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallIpNextHeader	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with a defined type of an extension header.	
Template Description	
Filter to match packets with a defined type of an extension header.	
M2 Parameter	
Ipv6Rule.nextHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00055]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig
BSW Parameter	**BSW Type**
FirewallNetworkLayerIcmpConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules for ICMP (Internet Control Message Protocol).	
Template Description	
Ipv4Rule.icmpRule: Configuration of filter rules for ICMP (Internet Control Message Protocol). **Ipv6Rule.icmpRule:** Configuration of filter rules for ICMP (Internet Control Message Protocol).	
M2 Parameter	
Ipv4Rule.icmpRule, Ipv6Rule.icmpRule	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00130]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/ FirewallNetworkLayerIcmpConfig
BSW Parameter	**BSW Type**
FirewallChecksumVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether checksum verification is performed or not.	
Template Description	

▽

△

TransportLayerRule.checksumVerification: Defines whether checksum verification is performed or not. **Ipv4Rule.checksumVerification:** Defines whether a Ipv4 header checksum verification is performed or not. **IcmpRule.checksumVerification:** Defines whether a Icmp header checksum verification is performed or not.
M2 Parameter
TransportLayerRule.checksumVerification, Ipv4Rule.checksumVerification, IcmpRule.checksumVerification

Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00025]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallNetworkLayerIcmpConfig

BSW Parameter	**BSW Type**
FirewallIcmpCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with the Icmp code.	
Template Description	
Filter to match packets with the Icmp code.	
M2 Parameter	
IcmpRule.code	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00132]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallNetworkLayerFilterConfig/FirewallNetworkLayerIpv6FilterConfig/FirewallNetworkLayerIcmpConfig

BSW Parameter	**BSW Type**
FirewallIcmpType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter to match packets with the Icmp type.	
Template Description	
Filter to match packets with the Icmp type.	
M2 Parameter	
IcmpRule.type	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00131]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallPayloadBytePatternFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a generic firewall rule that defines the individual bytes of a message that shall match.	
Template Description	
Configuration of a generic firewall rule that defines the individual bytes of a message that shall match.	
M2 Parameter	
PayloadBytePatternRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00076]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallPayloadBytePatternFilterConfig
BSW Parameter	**BSW Type**
FirewallPayloadByteConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a single byte in the datagram.	
Template Description	
Configuration of one byte in the datagram,	
M2 Parameter	
PayloadBytePatternRulePart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00077]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallPayloadBytePatternFilterConfig/FirewallPayloadByteConfig
BSW Parameter	**BSW Type**
FirewallPaylodByteOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the byte offset in the datagram (start byte of the Ethernet frame, i.e. offset 0 corresponds to the first byte of the destination MAC address).	
Template Description	
This attribute defines the byte offset in the datagram (start byte of the Ethernet frame, i.e. offset 0 corresponds to the first byte of the destination MAC address).	
M2 Parameter	
PayloadBytePatternRulePart.offset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00078]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallPayloadBytePatternFilterConfig/FirewallPayloadByteConfig
BSW Parameter	**BSW Type**
FirewallPaylodByteValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
This attribute defines the byteValue in the datagram.	
Template Description	
This attribute defines the byteValue (0..255) in the datagram.	
M2 Parameter	
PayloadBytePatternRulePart.value	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00080]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallRefillAmount	ECUC-INTEGER-PARAM-DEF
BSW Description	
This attribute defines the output rate that describes how many packets leave the queue per second (leaky-bucket Algorithm).	
Template Description	
This attribute defines the output rate that describes how many packets leave the queue per second (leaky-bucket Algorithm).	
M2 Parameter	
FirewallRule.refillAmount	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00026]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallSomeipProtocolFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of SOME/IP Protocol firewall rules	
Template Description	
Configuration of SOME/IP firewall rules	
M2 Parameter	
SomeipProtocolRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00068]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipClientId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the clientId in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the clientId in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.clientId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00070]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipLengthVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether length verification is performed or not.	
Template Description	
Defines whether length verification is performed or not.	
M2 Parameter	
SomeipProtocolRule.lengthVerification	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00075]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipMajorVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the majorVersion in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the majorVersion in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.majorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00069]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipMessageType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the message type in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the messageType in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.messageType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00072]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipMethodId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the methodId in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the methodId in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.methodId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00071]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipProtocolVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the protocol version in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the protocolVersion in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.protocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00074]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipReturnCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the return code in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP messages in which the returnCode in the SOME/IP header matches.	
M2 Parameter	
SomeipProtocolRule.returnCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00073]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipProtocolFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipServiceInterfaceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the serviceInterfaceId in the SOME/IP header matches.	
Template Description	
SomeipSdRule.serviceInterfaceId: Filter for SOME/IP SD messages in which the serviceInterfaceId in the SOME/IP header matches. **SomeipProtocolRule.serviceInterfaceId:** Filter for SOME/IP messages in which the serviceInterfaceId in the SOME/IP header matches.	
M2 Parameter	
SomeipSdRule.serviceInterfaceId, SomeipProtocolRule.serviceInterfaceId	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00065]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallSomeipSdFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of SOME/IP Service Discovery firewall rules	
Template Description	
Configuration of SOME/IP Service Discovery firewall rules	
M2 Parameter	
SomeipSdRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Fw_00057]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipEntryType	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the entryType in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP SD messages in which the entryType in the SOME/IP header matches.	
M2 Parameter	
SomeipSdRule.entryType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00067]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipEventGroupId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the eventGroupId in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP SD messages in which the eventGroupId in the SOME/IP header matches.	
M2 Parameter	
SomeipSdRule.eventGroupId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00066]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipMajorVersion	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is in the configured max and min value.	
Template Description	

▽

△

SomeipSdRule.maxMajorVersion: Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is smaller or equal than maxMajorVersion. **SomeipSdRule.minMajorVersion:** Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is greater or equal than minMajorVersion.	
M2 Parameter	
SomeipSdRule.maxMajorVersion, SomeipSdRule.minMajorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00058]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig/FirewallSomeipMajorVersion
BSW Parameter	**BSW Type**
FirewallMajorVersionMaxValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is smaller or equal than this value.	
Template Description	
Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is smaller or equal than maxMajorVersion.	
M2 Parameter	
SomeipSdRule.maxMajorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00060]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig/FirewallSomeipMajorVersion
BSW Parameter	**BSW Type**
FirewallMajorVersionMinValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is greater or equal than this value.	
Template Description	
Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is greater or equal than minMajorVersion.	
M2 Parameter	
SomeipSdRule.minMajorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00061]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**

▽

△

FirewallSomeipMinorVersion	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is in the configured max and min value.	
Template Description	
SomeipSdRule.maxMinorVersion: Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is smaller or equal than maxMinorVersion. **SomeipSdRule.minMinorVersion:** Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is greater or equal than minMinorVersion.	
M2 Parameter	
SomeipSdRule.maxMinorVersion, SomeipSdRule.minMinorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00062]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig/FirewallSomeipMinorVersion
BSW Parameter	**BSW Type**
FirewallMinorVersionMaxValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is smaller or equal than than this value.	
Template Description	
Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is smaller or equal than maxMinorVersion.	
M2 Parameter	
SomeipSdRule.maxMinorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00063]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig/FirewallSomeipMinorVersion
BSW Parameter	**BSW Type**
FirewallMinorVersionMinValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is greater or equal than this value.	
Template Description	
Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is greater or equal than minMinorVersion.	
M2 Parameter	
SomeipSdRule.minMinorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00064]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipServiceInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP SD messages in which the serviceInstanceId in the SOME/IP header matches.	
Template Description	
Filter for SOME/IP SD messages in which the serviceInstanceId in the SOME/IP header matches.	
M2 Parameter	
SomeipSdRule.serviceInstanceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00059]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallSomeipSdFilterConfig
BSW Parameter	**BSW Type**
FirewallSomeipServiceInterfaceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Filter for SOME/IP messages in which the serviceInterfaceId in the SOME/IP header matches.	
Template Description	
SomeipSdRule.serviceInterfaceId: Filter for SOME/IP SD messages in which the serviceInterfaceId in the SOME/IP header matches. **SomeipProtocolRule.serviceInterfaceId:** Filter for SOME/IP messages in which the serviceInterfaceId in the SOME/IP header matches.	
M2 Parameter	
SomeipSdRule.serviceInterfaceId, SomeipProtocolRule.serviceInterfaceId	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00065]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule
BSW Parameter	**BSW Type**
FirewallTransportLayerFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules on Transport Layer level.	
Template Description	
Configuration of filter rules on Transport Layer level.	
M2 Parameter	
TransportLayerRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Fw_00138]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig
BSW Parameter	**BSW Type**
FirewallTransportLayerTcpFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of filter rules for TCP on Transport Layer level.	
Template Description	
Configuration of TCP filter rules.	
M2 Parameter	
TcpRule	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00019]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig
BSW Parameter	**BSW Type**
FirewallChecksumVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether checksum verification is performed or not.	
Template Description	
TransportLayerRule.checksumVerification: Defines whether checksum verification is performed or not. **Ipv4Rule.checksumVerification:** Defines whether a Ipv4 header checksum verification is performed or not. **IcmpRule.checksumVerification:** Defines whether a Icmp header checksum verification is performed or not.	
M2 Parameter	
TransportLayerRule.checksumVerification, Ipv4Rule.checksumVerification, IcmpRule.checksumVerification	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00025]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterDestPort	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a destination port filter.	

▽

△

Template Description	
TransportLayerRule.minDestinationPortNumber: Filter to match packets with the minimum destination UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minDestinationPortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00022]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig/FirewallFilterDestPort
BSW Parameter	**BSW Type**
FirewallFilterPortLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port lower value.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.minDestinationPortNumber:** Filter to match packets with the minimum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.minDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00028]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig/FirewallFilterDestPort
BSW Parameter	**BSW Type**
FirewallFilterPortUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port upper value.	
Template Description	
TransportLayerRule.maxSourcePortNumber: Filter to match packets with the maximum source UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.maxSourcePortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Fw_00029]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterSrcPort	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a source port filter.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.maxSourcePortNumber:** Filter to match packets with the maximum source UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.maxSourcePortNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00020]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig/ FirewallFilterSrcPort
BSW Parameter	**BSW Type**
FirewallFilterPortLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port lower value.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.minDestinationPortNumber:** Filter to match packets with the minimum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.minDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00028]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig/ FirewallFilterSrcPort
BSW Parameter	**BSW Type**
FirewallFilterPortUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Definition of the filter port upper value.	
Template Description	
TransportLayerRule.maxSourcePortNumber: Filter to match packets with the maximum source UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.maxSourcePortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00029]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig
BSW Parameter	**BSW Type**
FirewallNumberOfParallelTcpSessions	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the maximal number of TCP Sessions that are allowed to be established.	
Template Description	
This attribute defines the maximal number of TCP Sessions that are allowed to be established.	
M2 Parameter	
TcpRule.numberOfParallelTcpSessions	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00035]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig
BSW Parameter	**BSW Type**
FirewallStateManagementBasedOnTcpFlags	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This attribute defines whether the StateManagement is based on TCP flags or not.	
Template Description	
This attribute defines whether the StateManagement is based on TCP flags or not.	
M2 Parameter	
TcpRule.stateManagementBasedOnTcpFlags	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00037]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerTcpFilterConfig	
BSW Parameter	**BSW Type**	
FirewallTimeoutCheck	ECUC-INTEGER-PARAM-DEF	
BSW Description		
This parameter defines the TCP Session timeout in seconds		
Template Description		
This attribute defines the TCP Session timeout in seconds		
M2 Parameter		
TcpRule.timeoutCheck		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00036]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig	
BSW Parameter	**BSW Type**	
FirewallTransportLayerUdpFilterConfig	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
Configuration of filter rules for UDP on Transport Layer level.		
Template Description		
Configuration of UDP filter rules.		
M2 Parameter		
UdpRule		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Fw_00038]

BSW Module	BSW Context	
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig	
BSW Parameter	**BSW Type**	
FirewallChecksumVerification	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		
Defines whether checksum verification is performed or not.		
Template Description		
TransportLayerRule.checksumVerification: Defines whether checksum verification is performed or not. **Ipv4Rule.checksumVerification:** Defines whether a Ipv4 header checksum verification is performed or not. **IcmpRule.checksumVerification:** Defines whether a Icmp header checksum verification is performed or not.		
M2 Parameter		
TransportLayerRule.checksumVerification, Ipv4Rule.checksumVerification, IcmpRule.checksumVerification		



△

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00025]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterDestPort	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a destination port filter.	
Template Description	
TransportLayerRule.minDestinationPortNumber: Filter to match packets with the minimum destination UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minDestinationPortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00022]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig/ FirewallFilterDestPort
BSW Parameter	**BSW Type**
FirewallFilterPortLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port lower value.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.minDestinationPortNumber:** Filter to match packets with the minimum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.minDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00028]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig/ FirewallFilterDestPort
BSW Parameter	**BSW Type**

▽

△

FirewallFilterPortUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port upper value.	
Template Description	
TransportLayerRule.maxSourcePortNumber: Filter to match packets with the maximum source UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.maxSourcePortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00029]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig
BSW Parameter	**BSW Type**
FirewallFilterSrcPort	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of a source port filter.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.maxSourcePortNumber:** Filter to match packets with the maximum source UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.maxSourcePortNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00020]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig/ FirewallFilterSrcPort
BSW Parameter	**BSW Type**
FirewallFilterPortLowerValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port lower value.	
Template Description	
TransportLayerRule.minSourcePortNumber: Filter to match packets with the minimum source UDP/TCP port number. **TransportLayerRule.minDestinationPortNumber:** Filter to match packets with the minimum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.minSourcePortNumber, TransportLayerRule.minDestinationPortNumber	

▽

△

Mapping Rule	Mapping Type
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00028]

BSW Module	BSW Context
Firewall	Firewall/FirewallRule/FirewallTransportLayerFilterConfig/FirewallTransportLayerUdpFilterConfig/FirewallFilterSrcPort
BSW Parameter	**BSW Type**
FirewallFilterPortUpperValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the filter port upper value.	
Template Description	
TransportLayerRule.maxSourcePortNumber: Filter to match packets with the maximum source UDP/TCP port number. **TransportLayerRule.maxDestinationPortNumber:** Filter to match packets with the maximum destination UDP/TCP port number.	
M2 Parameter	
TransportLayerRule.maxSourcePortNumber, TransportLayerRule.maxDestinationPortNumber	
Mapping Rule	**Mapping Type**
depends on usage context	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00029]

BSW Module	BSW Context
Firewall	Firewall
BSW Parameter	**BSW Type**
FirewallStateDependentRules	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Firewall rules that are defined in a firewall state	
Template Description	
Firewall rules that are defined in a firewall state	
M2 Parameter	
StateDependentFirewall	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00006]

BSW Module	BSW Context
Firewall	Firewall/FirewallStateDependentRules
BSW Parameter	**BSW Type**
FirewallActionForMatchingRules	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Firewall action that is performed if the referenced pattern matches.	

▽

△

Template Description	
Firewall rule that is defined by an action that is performed if the referenced pattern matches.	
M2 Parameter	
FirewallRuleProps	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00008]

BSW Module	BSW Context
Firewall	Firewall/FirewallStateDependentRules/FirewallActionForMatchingRules
BSW Parameter	**BSW Type**
FirewallAction	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Action that is performed by the firewall if the matchingRule is fulfilled.	
Template Description	
Action that is performed by the firewall if the matchingRule is fulfilled.	
M2 Parameter	
FirewallRuleProps.action	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00009]

BSW Module	BSW Context
Firewall	Firewall/FirewallStateDependentRules/FirewallActionForMatchingRules/FirewallAction
BSW Parameter	**BSW Type**
ALLOW	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Allow	
Template Description	
Firewall allows the communication	
M2 Parameter	
FirewallActionEnum.allow	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Firewall	Firewall/FirewallStateDependentRules/FirewallActionForMatchingRules/FirewallAction
BSW Parameter	**BSW Type**
BLOCK	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Block	
Template Description	
Firewall blocks the communication	
M2 Parameter	
FirewallActionEnum.block	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	**BSW Context**
Firewall	Firewall/FirewallStateDependentRules
BSW Parameter	**BSW Type**
FirewallDefaultAction	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This attribute defines a defaultAction.	
Template Description	
This attribute defines a defaultAction in case that the VehicleMode is not yet set.	
M2 Parameter	
StateDependentFirewall.defaultAction	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Fw_00007]

BSW Module	**BSW Context**
Firewall	Firewall/FirewallStateDependentRules/FirewallDefaultAction
BSW Parameter	**BSW Type**
ALLOW	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Allow	
Template Description	
Firewall allows the communication	
M2 Parameter	
FirewallActionEnum.allow	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	**BSW Context**
Firewall	Firewall/FirewallStateDependentRules/FirewallDefaultAction
BSW Parameter	**BSW Type**
BLOCK	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Block	
Template Description	
Firewall blocks the communication	
M2 Parameter	
FirewallActionEnum.block	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

C.25 Fr

BSW Module	**BSW Context**
Fr	Fr
BSW Parameter	**BSW Type**
FrGeneral	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
General configuration (parameters) of the FlexRay Driver module.	
Template Description	
FlexRay specific attributes to the physicalCluster	
M2 Parameter	
FlexrayCluster	
Mapping Rule	**Mapping Type**
Container shall be created if the ECU is connected to a FlexRay Cluster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00392]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration
BSW Parameter	**BSW Type**
FrController	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of the individual controller.	
Template Description	
FlexRay bus specific communication port attributes.	
M2 Parameter	
FlexrayCommunicationController	
Mapping Rule	**Mapping Type**
Container shall be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00083]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrFifo	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.	
Template Description	
One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.	
M2 Parameter	
FlexrayFifoConfiguration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00009]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrAdmitWithoutMessageId	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.	
Template Description	
Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.	
M2 Parameter	
FlexrayFifoConfiguration.admitWithoutMessageId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00006]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrBaseCycle	ECUC-INTEGER-PARAM-DEF
BSW Description	
FIFO cycle counter acceptance criteria.	
Template Description	
FIFO cycle counter acceptance criteria.	
M2 Parameter	
FlexrayFifoConfiguration.baseCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00007]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrChannels	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
FIFO channel admittance criteria.	
Template Description	
M2 Parameter	
FlexrayFifoConfiguration.channel	
Mapping Rule	**Mapping Type**
If channelA is referenced set Parameter to FR_CHANNEL_A. If channelB is referenced set parameter to FR_CHANNEL_B. If two identical FlexrayFifoConfiguration elements exist with references to A and B only one FrFifo container shall be created (FR_CHANNEL_AB	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00449]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrCycleRepetition	ECUC-INTEGER-PARAM-DEF
BSW Description	
FIFO cycle counter acceptance criteria. Valid values are 1,2,4,5,8,10,16,20,32,40,50,64. Remark: Values 1,2,4,8,16,32,64 are valid only for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
FIFO cycle counter acceptance criteria.	
M2 Parameter	
FlexrayFifoConfiguration.cycleRepetition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00008]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrFifoDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.	
Template Description	
FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.	
M2 Parameter	

▽

△

FlexrayFifoConfiguration.fifoDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00010]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrMsgIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
FIFO message identifier acceptance criteria (Mask filter).	
Template Description	
FIFO message identifier acceptance criteria (Mask filter).	
M2 Parameter	
FlexrayFifoConfiguration.msgIdMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00011]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrMsgIdMatch	ECUC-INTEGER-PARAM-DEF
BSW Description	
FIFO message identifier acceptance criteria (Match filter).	
Template Description	
FIFO message identifier acceptance criteria (Match filter).	
M2 Parameter	
FlexrayFifoConfiguration.msgIdMatch	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00012]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrRange	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
FIFO Frame Id range acceptance criteria.	
Template Description	

▽

△

FIFO Frame Id range acceptance criteria.	
M2 Parameter	
FlexrayFifoRange	
Mapping Rule	**Mapping Type**
create container for each Fifo configuration	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00013]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange
BSW Parameter	**BSW Type**
FrRangeMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Last FrameId of this range that will be accepted by the FIFO.	
Template Description	
Max Range.	
M2 Parameter	
FlexrayFifoRange.rangeMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00014]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange
BSW Parameter	**BSW Type**
FrRangeMin	ECUC-INTEGER-PARAM-DEF
BSW Description	
First FrameId of this range that will be accepted by the FIFO.	
Template Description	
Min Range.	
M2 Parameter	
FlexrayFifoRange.rangeMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00015]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPAllowHaltDueToClock	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the CC is allowed to transition to POC:halt. If set to false, the CC will not transition to the POC:halt state but will enter or remain in the POC:normal passive state (self healing would still be possible)	
Template Description	
Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the Communication Controller is allowed to transition to POC:halt. If set to false, the Communication Controller will not transition to the POC:halt state but will enter or remain in the normal POC (passive State).	
M2 Parameter	
FlexrayCommunicationController.allowHaltDueToClock	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00402]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPAllowPassiveToActive	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to zero, the CC is not allowed to transition from POC:normal passive to POC:normal active	
Template Description	
Number of consecutive even/odd cycle pairs that shall have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to 0, the Communication Controller is not allowed to transition from POC:norm	
M2 Parameter	
FlexrayCommunicationController.allowPassiveToActive	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00403]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPChannels	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Channels to which the node is connected. Implementation Type: Fr_ChannelType	
Template Description	
M2 Parameter	
PhysicalChannel.commConnector	
Mapping Rule	**Mapping Type**
If channelA refers the connector set parameter to FR_CHANNEL_A. If ChannelB refers the connector set parameter to FR_CHANNEL_B. If channelA and channelB refer the connector set parameter to FR_CHANNEL_AB,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00404]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPClusterDriftDamping	ECUC-INTEGER-PARAM-DEF
BSW Description	
Local cluster drift damping factor used for rate correction [Microticks]. Remark: Upper limit 10 for FlexRay Protocol 3.0 compliance.	
Template Description	
The cluster drift damping factor used in clock synchronization rate correction in microticks	
M2 Parameter	
FlexrayCommunicationController.clusterDriftDamping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00405]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPDecodingCorrection	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point [Microticks]. Remark: Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance. Upper limit 136 for FlexRay Protocol 3.0 compliance.	
Template Description	
Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)	
M2 Parameter	
FlexrayCommunicationController.decodingCorrection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00406]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPDelayCompensationA	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to c PropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Value used to compensate for reception delays on channel A Unit: Microticks. This optional parameter shall only be filled out if channel A is used.	
M2 Parameter	
FlexrayCommunicationController.delayCompensationA	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00407]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPDelayCompensationB	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to c PropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Value used to compensate for reception delays on channel B. Unit: Microticks. This optional parameter shall only be filled out if channel B is used.	
M2 Parameter	
FlexrayCommunicationController.delayCompensationB	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00408]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPExternalSync	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether the node is externally synchronized (operating as time gateway sink in an TT-E cluster) or locally synchronized. If FrPExternalSync is set to 'true' then FrPTwoKeySlotMode must also be set to 'true'. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TT-E Time Triggered External Sync cluster) or locally synchronized.	
M2 Parameter	
FlexrayCommunicationController.externalSync	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00448]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**

▽

△

FrPFallBackInternal	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether a time gateway sink node will switch to local clock operation when synchronization with the time gateway source node is lost (FrPFallBackInternal = true) or will instead go to POC:ready (FrPFallBackInternal =false). Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).	
M2 Parameter	
FlexrayCommunicationController.fallBackInternal	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00447]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPKeySlotId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID of the key slot, i.e., the slot used to transmit the startup frame, sync frame, or designated key slot frame. If this parameter is set to zero the node does not have a key slot. For Fr3.0: if the value is not provided in System Description it shall be configured to 0. For Fr2.1: if the value is not provided in System Description it is driver implementation specific which value to configure.	
Template Description	
ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedFor StartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory.	
M2 Parameter	
FlexrayCommunicationController.keySlotID	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00411]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPKeySlotOnlyEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether or not the node shall enter key slot only mode following startup. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pSingleSlotEnabled.	
Template Description	
Flag indicating whether or not the node shall enter key slot only mode following startup.	
M2 Parameter	
FlexrayCommunicationController.keySlotOnlyEnabled	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00425]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPKeySlotUsedForStartup	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether the key slot is used to transmit a startup frame. If FrPKeySlotUsedForStartup is set to true then Fr PKeySlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true.	
Template Description	
Flag indicating whether the Key Slot is used to transmit a startup frame.	
M2 Parameter	
FlexrayCommunicationController.keySlotUsedForStartUp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00412]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPKeySlotUsedForSync	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether the key slot is used to transmit a sync frame. If FrPKeySlotUsedForStartup is set to true then FrPKey SlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and Fr PKeySlotUsedForStartup must also be set to true.	
Template Description	
Flag indicating whether the Key Slot is used to transmit a sync frame.	
M2 Parameter	
FlexrayCommunicationController.keySlotUsedForSync	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00413]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPLatestTx	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of the last minislot in which a frame transmission can start in the dynamic segment. Remark: Upper limit 7980 for FlexRay Protocol 2.1 Rev A compliance.	

▽

△

Template Description	
The number of the last minislot in which a transmission can start in the dynamic segment for the respective node	
M2 Parameter	
FlexrayCommunicationController.latestTX	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00414]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPMacroInitialOffsetA	ECUC-INTEGER-PARAM-DEF
BSW Description	
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].	
Template Description	
Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel A is used.	
M2 Parameter	
FlexrayCommunicationController.macroInitialOffsetA	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00415]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPMacroInitialOffsetB	ECUC-INTEGER-PARAM-DEF
BSW Description	
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].	
Template Description	
Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel B is used.	
M2 Parameter	
FlexrayCommunicationController.macroInitialOffsetB	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00416]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPMicroInitialOffsetA	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationA and therefore it has to be set independently for each channel [Microticks].	
Template Description	
Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel A is used.	
M2 Parameter	
FlexrayCommunicationController.microInitialOffsetA	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00417]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPMicroInitialOffsetB	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationB and therefore it has to be set independently for each channel [Microticks].	
Template Description	
Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel B is used.	
M2 Parameter	
FlexrayCommunicationController.microInitialOffsetB	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00418]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPMicroPerCycle	ECUC-INTEGER-PARAM-DEF
BSW Description	
Nominal number of microticks in the communication cycle of the local node. If nodes have different microtick durations this number will differ from node to node [Microticks]. Remark: Lower limit 960 for FlexRay Protocol 3.0 compliance. Upper limit 640000 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
The nominal number of microticks in a communication cycle	

▽

△

M2 Parameter	
FlexrayCommunicationController.microPerCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00419]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPNmVectorEarlyUpdate	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating when the update of the Network Management Vector in the CHI shall take place. If FrPNmVectorEarlyUpdate is set to false, the update shall take place after the NIT. If FrPNmVectorEarlyUpdate is set to true, the update shall take place after the end of the static segment. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Flag indicating when the update of the Network Management Vector in the CHI shall take place. If set to false, the update shall take place after the NIT. If set to true, the update shall take place after the end of the static segment.	
M2 Parameter	
FlexrayCommunicationController.nmVectorEarlyUpdate	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00444]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPOffsetCorrectionOut	ECUC-INTEGER-PARAM-DEF
BSW Description	
Magnitude of the maximum permissible offset correction value [Microticks]. Remark: Upper limit 15567 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 15 for FlexRay Protocol 3.0 compliance.	
Template Description	
Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)	
M2 Parameter	
FlexrayCommunicationController.offsetCorrectionOut	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00421]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPOffsetCorrectionStart	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Start of the offset correction phase within the NIT, expressed as the number of macroticks from the start of cycle [Macroticks]. Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gOffsetCorrectionStart. Remark: Lower limit 9 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Start of the offset correction phase within the Network Idle Time (NIT), expressed as the number of macroticks from the start of cycle. Unit: macroticks	
M2 Parameter	
FlexrayCluster.offsetCorrectionStart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00450]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPPayloadLengthDynMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum payload length for dynamic frames [16 bit words].	
Template Description	
Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.	
M2 Parameter	
FlexrayCommunicationController.maximumDynamicPayloadLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00422]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPRateCorrectionOut	ECUC-INTEGER-PARAM-DEF
BSW Description	
Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle [Microticks]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift. Lower limit 3 for FlexRay Protocol 3.0 compliance. Upper limit 1923 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrectionOut) Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift.	
M2 Parameter	
FlexrayCommunicationController.rateCorrectionOut	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00423]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPSamplesPerMicrotick	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Number of samples per microtick. Remark: Allowed range N1SAMPLES, N2SAMPLES for FlexRay Protocol 3.0 compliance.	
Template Description	
Number of samples per microtick	
M2 Parameter	
FlexrayCommunicationController.samplesPerMicrotick	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00424]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPSecondKeySlotId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID of the second key slot, in which a second startup frame shall be sent when operating as a coldstart node in a TT-L or TT-D cluster. If this parameter is set to zero the node does not have a second key slot. Remark: Set to 0 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.	
M2 Parameter	
FlexrayCommunicationController.secondKeySlotId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00445]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPTwoKeySlotMode	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Flag indicating whether node operates as a coldstart node in a TT-E or TT-L cluster. If pTwoKeySlotMode is set to true then both pKeySlotUsedForSync and pKeySlotUsedForStartup must also be set to true. If pExternalSync is set to true then pTwo KeySlotMode must also be set to true. Remark: Set to false for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	

▽

△

Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.	
M2 Parameter	
FlexrayCommunicationController.twoKeySlotMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00446]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPWakeupChannel	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Channel used by the node to send a wakeup pattern. FrPWakeupChannel must be selected from among the channels configured by FrPChannels.	
Template Description	
Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)	
M2 Parameter	
FlexrayCommunicationConnector.wakeUpChannel	
Mapping Rule	**Mapping Type**
If channelA refers to the FlexrayCommunicationConnector and wakeUpChannel=true then Fr PWakeupChannel = FR_CHANNEL_A. If channelB refers to the FlexrayCommunicationConnector and wakeUpChannel = true then FrPWakeupChanel = FR_CHANNEL_B.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00426]

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPWakeupPattern	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of repetitions of the wakeup symbol that are combined to form a wakeup pattern when the node enters the POC:wakeup send state. Remark: Lower limit 2 for FlexRay Protocol 2.1 Rev A compliance.	
Template Description	
Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster	
M2 Parameter	
FlexrayCommunicationController.wakeUpPattern	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00427]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPdAcceptedStartupRange	ECUC-INTEGER-PARAM-DEF
BSW Description	
Expanded range of measured clock deviation allowed for startup frames during integration [Microticks]. Remark: Upper limit 1875 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 29 for FlexRay Protocol 3.0 compliance.	
Template Description	
Expanded range of measured clock deviation allowed for startup frames during integration. Unit:microtick	
M2 Parameter	
FlexrayCommunicationController.acceptedStartupRange	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00428]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPdListenTimeout	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster [Microticks]. Remark: Lower limit 1926 for FlexRay Protocol 3.0 compliance. Upper limit 1283846 for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks	
M2 Parameter	
FlexrayCommunicationController.listenTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fr_00429]

BSW Module	BSW Context
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPdMicrotick	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Duration of a microtick. Remark: Allowed range T12_5NS, T25NS, T50NS for FlexRay Protocol 3.0 compliance.	
Template Description	
Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds	
M2 Parameter	
FlexrayCommunicationController.microtickDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Fr_00431]

C.26 FrIf

BSW Module	BSW Context	
FrIf	FrIf	
BSW Parameter		**BSW Type**
FrIfConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR FrIf module.		
Template Description		
FlexRay specific attributes to the physicalCluster		
M2 Parameter		
FlexrayCluster		
Mapping Rule		**Mapping Type**
Container shall be created if the ECU is connected to a FlexRay Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06001]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig	
BSW Parameter		**BSW Type**
FrIfCluster		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container specifies a FrIf Cluster and all related data which is required to enable communication of the Cluster. A Cluster may consist of more than one Controller.		
Template Description		
FlexRay specific attributes to the physicalCluster		
M2 Parameter		
FlexrayCluster		
Mapping Rule		**Mapping Type**
Container shall be created if the ECU is connected to a FlexRay Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_05366]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfController		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration of FlexRay CC.		

▽

△

Template Description	
FlexRay bus specific communication port attributes.	
M2 Parameter	
FlexrayCommunicationController	
Mapping Rule	**Mapping Type**
Container shall be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_05363]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController
BSW Parameter	**BSW Type**
FrIfFrameTriggering	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
A Frame triggering contains the communication parameters of the FlexRay Frame as well as a reference to the Frame Construction Plan.	
Template Description	
FlexRay specific attributes to the FrameTriggering	
M2 Parameter	
FlexrayFrameTriggering	
Mapping Rule	**Mapping Type**
If a FlexrayFrameTriggering exists in the System Extract that is connected via a FramePort reference to the regarded Ecu the following two cases exist: 1) If the FlexrayFrameTriggering contains exactly one FlexrayAbsolutelyScheduledTiming then only one FrIfFrameTriggering container shall be created. 2) If the FlexrayFrameTriggering contains more than one Flexray AbsolutelyScheduledTiming (e.g. to describe a multiple sending within one communication cycle) this FrIfFrameTriggering container shall be created once per defined FlexrayAbsolutelyScheduled Timing. Each created FrIfFrameTriggering container shall refer to the same FrIfFrameStructure.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06090]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfAllowDynamicLSduLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Allows L-PDU length reduction ('FrIfLSduLength' defines max. length) and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU.	
Template Description	
Allows L-PDU length reduction and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU. If this attribute is set to true than the referenced Frame length attribute defines the max. length.	
M2 Parameter	
FlexrayFrameTriggering.allowDynamicLSduLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Frlf_06049]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering
BSW Parameter	**BSW Type**
FrlfBaseCycle	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the FlexRay Base Cycle used to transmit this FlexRay Frame.	
Template Description	
The first communication cycle where the frame is sent. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.	
M2 Parameter	
CycleRepetition.BaseCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06051]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering
BSW Parameter	**BSW Type**
FrlfChannel	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter contains the FlexRay Channel used to transmit this FlexRay Frame.	
Template Description	
M2 Parameter	
PhysicalChannel.frameTriggering	
Mapping Rule	**Mapping Type**
FrameTriggering element in the System Template is aggregated by the PhysicalChannel that is used to transmit this FlexRay Frame	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06052]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController/FrlfFrameTriggering
BSW Parameter	**BSW Type**
FrlfCycleRepetition	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the FlexRay Cycle Repetition used to transmit this FlexRay Frame. Possible values for FlexRay Protocol version 2.1: 1,2,4,8,16,32,64 Possible values for FlexRay Protocol version 3.0: 1,2,4,5,8,10,16,20,32,40,50,64	
Template Description	

▽

△

The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.	
M2 Parameter	
CycleRepetition.CycleRepetition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06053]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfFrameStructureRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the Construction Plan of the FlexRay Frame.	
Template Description	
M2 Parameter	
FrameTriggering.frame	
Mapping Rule	**Mapping Type**
Reference shall comply to the reference in the System Description between the FrameTriggering element and the Frame.element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06048]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfLSduLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
The payload length of the Frame is given here. This parameter is required for validation if configured PDUs and update information fits into the Frame at configuration time [bytes].	
Template Description	
The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay). The frameLength of zero bytes is allowed. Please consider also TPS_SYST_02255.	
M2 Parameter	
Frame.frameLength	
Mapping Rule	**Mapping Type**
Find Frame that is referenced by the regarded FrameTriggering and use the frameLength attribute	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06054]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfMessageId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.	
Template Description	
The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.	
M2 Parameter	
FlexrayFrameTriggering.messageId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_00010]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfPayloadPreamble	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Switching the Payload Preamble bit.	
Template Description	
Switching the Payload Preamble bit.	
M2 Parameter	
FlexrayFrameTriggering.payloadPreambleIndicator	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06055]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfSlotId	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter contains the FlexRay Slot ID used to transmit this FlexRay Frame.	
Template Description	
In the static part the SlotID defines the slot in which the frame is transmitted. The SlotID also determines, in combination with FlexrayCluster::numberOfStaticSlots, whether the frame is sent in static or dynamic segment. In the dynamic part, the slot id is equivalent to a priority. Lower dynamic slot ids are all sent until the end of the dynamic segment. Higher numbers, which were ignored that time, have to wait one cycle and then shall try again. minValue: 1 maxValue: 2047	
M2 Parameter	

△

FlexrayAbsolutelyScheduledTiming.slotID	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06056]

BSW Module	**BSW Context**
Frlf	Frlf/FrlfConfig/FrlfCluster/FrlfController
BSW Parameter	**BSW Type**
FrlfLPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Reference to a L-PDU index	
Template Description	
Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.	
M2 Parameter	
Frame	
Mapping Rule	**Mapping Type**
Create container for each FlexRay Frame that is transmitted or received via the regarded communication controller..	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_05364]

BSW Module	**BSW Context**
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfDetectNITError	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Indicates whether NIT error status of each cluster shall be detected or not.	
Template Description	
Indicates whether NIT error status of each cluster shall be detected or not.	
M2 Parameter	
FlexrayCluster.detectNitError	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_00003]

BSW Module	**BSW Context**
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGChannels	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The channels that are used by the cluster. Implementation Type: Fr_ChannelType	

▽

△

Template Description	
FlexRay specific attributes to the physicalChannel	
M2 Parameter	
FlexrayPhysicalChannel	
Mapping Rule	**Mapping Type**
The channels that are used by the cluster are described in the System Template by the CommunicationCluster-PhysicalChannel relationship.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06006]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGColdStartAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of times a node in the cluster is permitted to attempt to start the cluster by initiating schedule synchronization	
Template Description	
The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization	
M2 Parameter	
FlexrayCluster.coldStartAttempts	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06008]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGCycleCountMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.	
Template Description	
Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Parameter	
FlexrayCluster.cycleCountMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06086]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGListenNoise		ECUC-INTEGER-PARAM-DEF
BSW Description		
Upper limit for the start up listen timeout and wake up listen timeout in the presence of noise. It is used as a multiplier of the node parameter pdListenTimeout.		
Template Description		
Upper limit for the start up and wake up listen timeout in the presence of noise. Expressed as a multiple of the cluster constant pdListenTimeout. Unit microticks		
M2 Parameter		
FlexrayCluster.listenNoise		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06009]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGMacroPerCycle		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of macroticks in a communication cycle. Note: Lower limit 10 for FlexRay Protocol 2.1 Rev. A compliance		
Template Description		
The number of macroticks in a communication cycle		
M2 Parameter		
FlexrayCluster.macroPerCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06010]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGMaxWithoutClockCorrectFatal		ECUC-INTEGER-PARAM-DEF
BSW Description		
Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state. [Even/odd cycle pairs].		
Template Description		
Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state.		
M2 Parameter		
FlexrayCluster.maxWithoutClockCorrectionFatal		



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06011]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGMaxWithoutClockCorrectPassive	ECUC-INTEGER-PARAM-DEF
BSW Description	
Threshold used for testing the vClockCorrectionFailed counter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state. [Even/Odd cycle pairs]	
Template Description	
Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state.	
M2 Parameter	
`FlexrayCluster.maxWithoutClockCorrectionPassive`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06012]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGNetworkManagementVectorLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the Network Management vector in a cluster [bytes]	
Template Description	
Length of the Network Management vector in a cluster [bytes]	
M2 Parameter	
`FlexrayCluster.networkManagementVectorLength`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06013]

BSW Module	BSW Context
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGNumberOfMinislots	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Number of minislots in the dynamic segment Remark: Upper limit 7986 for FlexRay Protocol 2.1 Rev. A compliance	
Template Description	
Number of Minislots in the dynamic segment.	
M2 Parameter	
FlexrayCluster.numberOfMinislots	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06014]

BSW Module	**BSW Context**
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGNumberOfStaticSlots	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of static slots in the static segment	
Template Description	
The number of static slots in the static segment.	
M2 Parameter	
FlexrayCluster.numberOfStaticSlots	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06015]

BSW Module	**BSW Context**
Frlf	Frlf/FrlfConfig/FrlfCluster
BSW Parameter	**BSW Type**
FrlfGPayloadLengthStatic	ECUC-INTEGER-PARAM-DEF
BSW Description	
Payload length of a static frame [16 bit words]	
Template Description	
Globally configured payload length of a static frame. Unit: 16-bit WORDS.	
M2 Parameter	
FlexrayCluster.payloadLengthStatic	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06018]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGSyncFrameIDCountMax		ECUC-INTEGER-PARAM-DEF
BSW Description		
Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.		
Template Description		
Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.		
M2 Parameter		
FlexrayCluster.syncFrameIdCountMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06019]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdActionPointOffset		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of macroticks the action point is offset from the beginning of a static slot.		
Template Description		
The offset of the action point in networks		
M2 Parameter		
FlexrayCluster.actionPointOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06020]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdBit		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Nominal bit time in seconds		
Template Description		
Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPerBit * gdSampleClockPeriod. Unit: seconds (gdBit)		
M2 Parameter		
FlexrayCluster.bit		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06021]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdCasRxLowMax		ECUC-INTEGER-PARAM-DEF
BSW Description		
Upper limit of the CAS acceptance windows [gdBit] Remark: Range 67 to 99 for FlexRay Protocol 2.1 Rev. A compliance		
Template Description		
Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration		
M2 Parameter		
FlexrayCluster.casRxLowMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06024]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdCycle		ECUC-FLOAT-PARAM-DEF
BSW Description		
Length of the cycle, expressed in [s] Remark: Lower limit 0.000024 for FlexRay Protocol 3.0 compliance.		
Template Description		
Length of the cycle. Unit: seconds		
M2 Parameter		
FlexrayCluster.cycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06025]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdDynamicSlotIdlePhase		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration of the idle phase within a dynamic slot [Minislots].		
Template Description		
The duration of the dynamic slot idle phase in minislots.		
M2 Parameter		
FlexrayCluster.dynamicSlotIdlePhase		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06026]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdIgnoreAfterTx		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration for which the bitstrobing is paused after transmission [gdBit]. Remark: Set to 0 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Duration for which the bitstrobing is paused after transmission [gdBit].		
M2 Parameter		
FlexrayCluster.ignoreAfterTx		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_00012]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdMacrotick		ECUC-FLOAT-PARAM-DEF
BSW Description		
Duration of the cluster wide nominal macrotick, expressed in s		
Template Description		
Duration of the cluster wide nominal macrotick, expressed in s.		
M2 Parameter		
FlexrayCluster.macrotickDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06027]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdMiniSlotActionPointOffset		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of Macroticks the Minislot action point is offset from the beginning of a Minislot [Macroticks].		
Template Description		
The Offset of the action point within a minislot. Unit: macroticks		
M2 Parameter		
FlexrayCluster.minislotActionPointOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06032]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdMinislot		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration of a minislot [Macroticks]		
Template Description		
The duration of a minislot (dynamic segment). Unit: macroticks.		
M2 Parameter		
FlexrayCluster.minislotDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06033]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdNit		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration of the Network Idle Time [Macroticks] Remark: Upper limit 805 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
The duration of the network idle time in macroticks		
M2 Parameter		
FlexrayCluster.networkIdleTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06034]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdSampleClockPeriod		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Sample clock period		
Template Description		
Sample clock period. Unit: seconds		
M2 Parameter		
FlexrayCluster.sampleClockPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06035]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdStaticSlot		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration of a static slot [Macroticks]. Remark: Range 4-661 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
The duration of a slot in the static segment. Unit: macroticks		
M2 Parameter		
FlexrayCluster.staticSlotDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06036]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdSymbolWindow		ECUC-INTEGER-PARAM-DEF
BSW Description		
Duration of the symbol window [Macroticks]. Remark: Range 0-142 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits in the Transmission Start Sequence [gdBits].		
M2 Parameter		
FlexrayCluster.transmissionStartSequenceDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06037]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdSymbolWindowActionPointOffset		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks]. Remark: Set to GdActionPointOffset for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].		
M2 Parameter		
FlexrayCluster.symbolWindowActionPointOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_00011]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdTSSTransmitter		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of bits in the Transmission Start Sequence [gdBits]. Remark: Lower limit 3 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits in the Transmission Start Sequence [gdBits].		
M2 Parameter		
FlexrayCluster.transmissionStartSequenceDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06038]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupRxIdle		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle. Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit:bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.		
M2 Parameter		
FlexrayCluster.wakeupRxIdle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06039]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupRxLow		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of bits used by the node to test the duration of the LOW phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow. Lower limit 11 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit:bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow.		
M2 Parameter		



△

FlexrayCluster.wakeupRxLow	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Frlf_06040]

BSW Module	**BSW Context**	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		**BSW Type**
FrlfGdWakeupRxWindow		ECUC-INTEGER-PARAM-DEF
BSW Description		
The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow. Upper limit 301 for FlexRay Protocol 2.1 Rev. A compliance.		
Template Description		
The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow.		
M2 Parameter		
FlexrayCluster.wakeupRxWindow		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Frlf_06041]

BSW Module	**BSW Context**	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		**BSW Type**
FrlfGdWakeupTxActive		ECUC-INTEGER-PARAM-DEF
BSW Description		
Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxLow.		
Template Description		
Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP. Unit:bitDuration		
M2 Parameter		
FlexrayCluster.wakeupTxActive		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Frlf_06043]

BSW Module	**BSW Context**	
Frlf	Frlf/FrlfConfig/FrlfCluster	
BSW Parameter		**BSW Type**

▽

△

FrIfGdWakeupTxIdle	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of bits used by the node to transmit the 'idle' part of a wakeup symbol [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxIdle.	
Template Description	
Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gDbit	
M2 Parameter	
FlexrayCluster.wakeupTxIdle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06042]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfSafetyMargin	ECUC-INTEGER-PARAM-DEF
BSW Description	
Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has be resynchronized.	
Template Description	
Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has be resynchronized.	
M2 Parameter	
FlexrayCluster.safetyMargin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_00004]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig
BSW Parameter	**BSW Type**
FrIfFrameStructure	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The Frame structure specifies a Construction Plan how a Frame is assembled with PDUs and their respective Update-Bits.	
Template Description	
Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.	
M2 Parameter	
Frame	
Mapping Rule	**Mapping Type**
Create container for each FlexRay Frame that is transmitted or received by the regarded ECU. IPduToFrameMapping element in the System Template contains the construction plan.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_05370]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfFrameStructure
BSW Parameter	**BSW Type**
FrIfByteOrder	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the ByteOrder of all Pdus that are mapped into the Frame. The absolute position of a Pdu in the Frame is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the FrIfPduOffset indicates the position of the most significant bit in the Frame. If LITTLE_ENDIAN is specified, the FrIfPduOffset indicates the position of the least significant bit in the Frame.	
Template Description	
This attribute defines the order of the bytes of the Pdu and the packing into the Frame. Please consider that [constr_3246] and [constr_3222] are restricting the usage of this attribute.	
M2 Parameter	
PduToFrameMapping.packingByteOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06113]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfFrameStructure
BSW Parameter	**BSW Type**
FrIfPdusInFrame	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container holds all the information about a PDU in a FlexRay Frame.	
Template Description	
A PduToFrameMapping defines the composition of Pdus in each frame.	
M2 Parameter	
PduToFrameMapping	
Mapping Rule	**Mapping Type**
Container shall be created for each IPduToFrameMapping element inside the frame.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_05371]

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame
BSW Parameter	**BSW Type**
FrIfPduOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
The value specifies the offset of the PDU within the Frame [bytes].	
Template Description	

▽

△

This attribute describes the bitposition of a Pdu within a Frame. Please note that the absolute position of the Pdu in the Frame is determined by the definition of the packingByteOrder attribute. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the Frame. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the Frame. The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. The Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed.	
M2 Parameter	
PduToFrameMapping.startPosition	
Mapping Rule	**Mapping Type**
Please note that the startPosition attribute is defined in bits and the FrIfPduOffset parameter is defined in bytes.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06070]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame
BSW Parameter	**BSW Type**
FrIfPduUpdateBitOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
This value specifies where the PDU's Update-Bit is stored in the Frame (bit location of PDU's Update-Bit in the FlexRay Frame).	
Template Description	
Indication to the receivers that the corresponding Pdu was updated by the sender. This attribute describes the position of the update bit in the frame that aggregates this PDUToFrameMapping. Length is always one bit. Note that the exact bit position of the updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByte Last. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing Frame still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.	
M2 Parameter	
PduToFrameMapping.updateIndicationBitPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_06071]

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig
BSW Parameter	**BSW Type**
FrIfPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.	
Template Description	
Collection of all Pdus that can be routed through a bus interface.	

▽

△

M2 Parameter	
Pdu	
Mapping Rule	**Mapping Type**
The container shall be created for each Pdu that is contained in a FlexRay Frame.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrIf_05372]

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu	
BSW Parameter		**BSW Type**
FrIfPduDirection		ECUC-CHOICE-CONTAINER-DEF
BSW Description		
A PDU is either transmit or receive		
Template Description		
Communication Direction of the Connector Port (input or output Port).		
M2 Parameter		
CommConnectorPort.communicationDirection		
Mapping Rule		**Mapping Type**
The PduTriggering contains a reference to a IPduPort with the communicationDirection.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_06072]

BSW Module	BSW Context	
FrIf	FrIf	
BSW Parameter		**BSW Type**
FrIfGeneral		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the general configuration parameters of the FlexRay Interface.		
Template Description		
FlexRay specific attributes to the physicalCluster		
M2 Parameter		
FlexrayCluster		
Mapping Rule		**Mapping Type**
Container shall be created if the ECU is connected to a FlexRay Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrIf_05360]

C.27 FrNm

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig	
BSW Parameter		**BSW Type**

▽

△

FrNmChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters for a FlexRay NM Channel.	
Template Description	
FlexRay specific NM cluster attributes.	
M2 Parameter	
FlexrayNmCluster	
Mapping Rule	**Mapping Type**
Create Container for each existing FlexrayNmCluster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00006]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmCarWakeUpBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the Bit position of the CWU within the NM-Message.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
FlexrayNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCar WakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00076]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmCarWakeUpBytePosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the Byte position of the CWU within the NM-Message.	
Template Description	
Specifies the bit position of the CarWakeUp within the NmPdu.	
M2 Parameter	
FlexrayNmCluster.nmCarWakeUpBitPosition	
Mapping Rule	**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCar WakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00075]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmCarWakeUpFilterEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilter NodeId is considered as CWU request. FALSE - CWU Filtering is not supported TRUE - CWU Filtering is supported	
Template Description	
If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.	
M2 Parameter	
FlexrayNmCluster.nmCarWakeUpFilterEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00077]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmCarWakeUpFilterNodeId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilterNodeId is considered as CWU request.	
Template Description	
Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.	
M2 Parameter	
FlexrayNmCluster.nmCarWakeUpFilterNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00078]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmCarWakeUpRxEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of CarWakeUp bit evaluation in received NM messages. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported	
Template Description	
If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.	
M2 Parameter	
FlexrayNmCluster.nmCarWakeUpRxEnabled	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00074]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmDynamicPncToChannelMappingEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping If M2 Parameter not defined then do not create FrNmDynamicPncToChannelMapping Enabled	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_FrNm_00092]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmNodeDetectionEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter is used to enable or disable node detection support for a FrNm Channel.	
Template Description	
Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter	
NmCluster.nmNodeDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00086]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmNodeId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

NM node identifier configured for the respective FlexRay Channel. It is used for identifying the respective NM node in the NM-cluster. It must be unique for each NM node within one NM cluster.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
NmNode.nmNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00017]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmPduScheduleVariant	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the PDU scheduling variant that should be used for this channel. Option 1 NM-Vote and NM-Data in static segment (one PDU) Option 2 NM-Vote and NM-Data in dynamic segment (one PDU) Option 3 NM-Vote and NM-Data in static segment (separate PDU) Option 4 NM-Vote in static segment and NM-Data in dynamic segment Option 5 NM-Vote in dynamic segment and NM-Data in static segment Option 6 NM-Vote and NM-Data in dynamic segment (separate PDU) Option 7 Combined NM-Vote and CBV in static segment and NM-Data in dynamic segment	
Template Description	
FrNm schedule variant according to FrNm SWS.	
M2 Parameter	
FlexrayNmClusterCoupling.nmScheduleVariant	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00022]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmPnEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
NmCluster.nmPncParticipation	
Mapping Rule	**Mapping Type**
If NmCluster.nmPncParticipation has the value "true" or is not defined then FrNmPnEnabled shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00072]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the FlexRay NM RX PDU:s.	
Template Description	
M2 Parameter	
NmNode.rxNmPdu	
Mapping Rule	**Mapping Type**
Create Container if the regarded NmNode recieves a Pdu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00010]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu
BSW Parameter	**BSW Type**
FrNmRxPduContainsData	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines if the PDU contains NM Data.	
Template Description	
Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain User Data that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.	
M2 Parameter	
NmPdu.nmDataInformation, NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00027]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu
BSW Parameter	**BSW Type**
FrNmRxPduContainsVote	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines if the PDU contains NM Vote information.	
Template Description	
Defines if the Pdu contains NM Vote information.	
M2 Parameter	
NmPdu.nmVoteInformation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00026]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmSourceNodeIdentifierEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter is used to enable or disable SourceNodeIdentifier support for a FrNm Channel.	
Template Description	
Enables the source node identifier.	
M2 Parameter	
NmCluster.nmNodeIdEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00085]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmSynchronizationPointEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines if this channel shall provide the synchronization point indication to the NM Interface.	
Template Description	
If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.	
M2 Parameter	
NmCluster.nmSynchronizingNetwork	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00021]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the FlexRay NM TX PDU:s.	
Template Description	
M2 Parameter	
NmNode.txNmPdu	
Mapping Rule	**Mapping Type**
Create Container if the regarded NmNode transmits a Pdu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00009]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu
BSW Parameter	**BSW Type**
FrNmTxPduContainsData	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameted defines if the PDU contains NM Data.	
Template Description	
Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain User Data that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.	
M2 Parameter	
NmPdu.nmDataInformation, NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00024]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu
BSW Parameter	**BSW Type**
FrNmTxPduContainsVote	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameted defines if the PDU contains NM Vote information.	
Template Description	
Defines if the Pdu contains NM Vote information.	
M2 Parameter	
NmPdu.nmVoteInformation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00023]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers
BSW Parameter	**BSW Type**
FrNmUserDataTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This optional container is used to configure the UserNm PDU. This container is only available if FrNmComUserDataSupport is enabled.	
Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrNm_00055]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter	**BSW Type**	
FrNmDataCycle	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Number of FlexRay Schedule Cycles needed to transmit the NM Data of all ECUs on the FlexRay bus		
Template Description		
Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNm Cluster.		
M2 Parameter		
FlexrayNmCluster.nmDataCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrNm_00031]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter	**BSW Type**	
FrNmMainFunctionPeriod	ECUC-FLOAT-PARAM-DEF	
BSW Description		
This parameter defines the processing cycle of the main function of FrNm module in seconds.		
Template Description		
Defines the processing cycle of the main function of FrNm module.		
M2 Parameter		
FlexrayNmCluster.nmMainFunctionPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrNm_00035]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter	**BSW Type**	
FrNmReadySleepCnt	ECUC-INTEGER-PARAM-DEF	
BSW Description		
FrNm switches to bus sleep mode at the end of the FrNmReadySleepCnt+1 repetition cycle without any NM vote. E.g. on a value of "1", the NM-State Machine will leave the Ready Sleep State after two NM Repetition Cycles with no "keep awake" votes.		
Template Description		
The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReady SleepTime seconds after the completion of the last repetition cycle containing a NM vote.		

▽

△

M2 Parameter	
FlexrayCommunicationConnector.nmReadySleepTime	
Mapping Rule	**Mapping Type**
FrNmReadySleepCnt = ((Float2Int(nmReadySleepTime/cycle))/nmRepetitionCycle)-1	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00051]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming
BSW Parameter	**BSW Type**
FrNmRemoteSleepIndTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep. The value "0" denotes that no Remote Sleep Indication functionality is configured.	
Template Description	
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.	
M2 Parameter	
FlexrayNmCluster.nmRemoteSleepIndicationTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00029]

BSW Module	BSW Context
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming
BSW Parameter	**BSW Type**
FrNmRepeatMessageTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Repeat Message State. Defines the time in seconds how long the NM shall stay in the Repeat Message State. The value "0" denotes that no Repeat Message State is configured, which means that Repeat Message State is transient and implies that it is left immediately after entry and consequently no startup stability is guaranteed and no node detection procedure is possible.	
Template Description	
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
M2 Parameter	
FlexrayNmCluster.nmRepeatMessageTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00030]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter	**BSW Type**	
FrNmRepetitionCycle	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Number of Flexray Schedule Cycles used to repeat the transmission of the Nm vote of all ECUs on the Flexray Bus.		
Template Description		
Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all FlexRay NmEcus of this FlexRayNmCluster. This value shall be an integral multiple of nmVotingCycle.		
M2 Parameter		
FlexrayNmCluster.nmRepetitionCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrNm_00033]

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter	**BSW Type**	
FrNmVotingCycle	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Number of FlexRay Schedule Cycles needed to transmit the Nm vote of all ECUs on the FlexRay Bus.		
Template Description		
Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNm Cluster.		
M2 Parameter		
FlexrayNmCluster.nmVotingCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrNm_00032]

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter	**BSW Type**	
FrNmBusSynchronizationEnabled	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		
Pre-processor switch for enabling the bus synchronization.		
Template Description		
Enables bus synchronization support.		
M2 Parameter		
NmEcu.nmBusSynchronizationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrNm_00048]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmComUserDataSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.	
Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00054]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmDynamicPncToChannelMappingSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
If at least one dynamicPncToChannelMappingEnabled attribute is defined and if at least one CommunicationConnector of the EcuInstance has dynamicPncToChannelMappingEnabled set to true, then FrNmDynamicPncToChannelMappingSupport shall be set to true. Otherwise FrNm DynamicPncToChannelMappingSupport shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00090]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmHwVoteEnable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the processing of FlexRay Hardware aggregated NM-Votes. This switch enables/disables the optional API FrIf_GetNmVector.	
Template Description	
Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.	
M2 Parameter	

▽

△

FlexrayNmEcu.nmHwVoteEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00050]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmPassiveModeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling Passive Node Configuration support.	
Template Description	
Enables support of the Passive Mode. The passive mode is configurable per channel.	
M2 Parameter	
NmNode.nmPassiveModeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping. nmNode.nmPassiveModeEnabled shall always have the same value in all Nm Clusters with the same bus protocol in the scope of one EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00043]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmPduRxIndicationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling PDU reception indication.	
Template Description	
Switch for enabling the PDU Rx Indication.	
M2 Parameter	
NmEcu.nmPduRxIndicationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00046]

BSW Module	**BSW Context**
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmRemoteSleepIndicationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling remote sleep indication. calculationFormula = If (FrNmPassiveModeEnabled == True) then Equal(False) else Equal(False or True)	

▽

△

Template Description	
Switch for enabling remote sleep indication support.	
M2 Parameter	
NmEcu.nmRemoteSleepIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00044]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmStateChangeIndicationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling state change indication.	
Template Description	
Enables the CAN Network Management state change notification.	
M2 Parameter	
NmEcu.nmStateChangeIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00047]

BSW Module	BSW Context
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures
BSW Parameter	**BSW Type**
FrNmUserDataEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling user data support.	
Template Description	
Switch for enabling user data support.	
M2 Parameter	
NmEcu.nmUserDataEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrNm_00039]

C.28 FrSM

BSW Module	BSW Context
FrSM	FrSM/FrSMConfig/FrSMCluster
BSW Parameter	**BSW Type**
FrSMIsColdstartEcu	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
True: The ECU is a coldstart node for this FlexRay cluster. False: The ECU is no coldstart node for this FlexRay cluster.	
Template Description	
FlexrayCommunicationController.keySlotID: ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedFor StartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory. **FlexrayCommunicationController.keySlotUsedForStartUp:** Flag indicating whether the Key Slot is used to transmit a startup frame.	
M2 Parameter	
FlexrayCommunicationController.keySlotID, FlexrayCommunicationController. keySlotUsedForStartUp	
Mapping Rule	**Mapping Type**
<= TRUE if keySlotId existing and valid (i.e. not 0) and keySlotUsedForStartUp set to true <= FALSE otherwise	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrSM_00068]

BSW Module	BSW Context
FrSM	FrSM/FrSMConfig/FrSMCluster
BSW Parameter	**BSW Type**
FrSMTrcvStdbyDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (FrIfGdCycle). A value of 0 shall imply that the timer is not used.	
Template Description	
The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value. Not specifying a value or a value of 0 shall imply that the timer is not used.	
M2 Parameter	
FlexrayCluster.tranceiverStandbyDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrSM_00170]

C.29 FrTSyn

BSW Module	BSW Context
FrTSyn	FrTSyn
BSW Parameter	**BSW Type**
FrTSynGlobalTimeDomain	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This represents the existence of a global time domain on Flexray. The FrTSyn module can administrate several global time domains at the same time that in itself form a hierarchy of domains and sub-domains. If the FrTSyn exists it is assumed that at least one global time domain exists.	
Template Description	
This represents the ability to define a global time domain.	
M2 Parameter	
GlobalTimeDomain	
Mapping Rule	**Mapping Type**
The container shall exist if a GlobalTimeDomain exists that has a slave or master that refers to a CommunicationConnector that in turn is aggregated by the EcucInstance for which the ECU configuration is created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00004]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
FrTSynGlobalTimeDomainId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The global time domain ID.	
Template Description	
This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.	
M2 Parameter	
GlobalTimeDomain.domainId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00005]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
FrTSynGlobalTimeMaster	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration of the global time master. Each global time domain is required to have exactly one global time master. This master may or may not exist on the configured ECU.	
Template Description	
This represents the generic concept of a global time master.	

▽

△

M2 Parameter	
GlobalTimeMaster	
Mapping Rule	**Mapping Type**
The existence of the FrTSynGlobalTimeMaster is bound to the existence of a GlobalTimeMaster that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00006]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
FrTSynCyclicMsgResumeTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the time where the 1st regular cycle time based message transmission takes place, after an immediate transmission before. Unit: seconds	
Template Description	
Defines the minimum time between an "immediate" message and the next periodic message.	
M2 Parameter	
GlobalTimeMaster.immediateResumeTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00032]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
FrTSynGlobalTimeDebounceTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents the configuration of a TX debounce time for SYNC messages compared to a message before with the same PDU. Unit: seconds	
Template Description	
Defines the minimum amount of time between two time sync messages are transmitted.	
M2 Parameter	
GlobalTimeDomain.debounceTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00033]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeMasterPdu
BSW Parameter	**BSW Type**
FrTSynGlobalTimePduRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated Global TimeSlaves.	
M2 Parameter	
GlobalTimeDomain.pduTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00020]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
FrTSynGlobalTimeTxCrcSecured	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This represents the configuration of whether or not CRC is supported.	
Template Description	
Definition of whether or not CRC is supported. This is only relevant for selected bus systems.	
M2 Parameter	
GlobalTimeFrMaster.crcSecured	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00013]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTxCrcSecured
BSW Parameter	**BSW Type**
CRC_NOT_SUPPORTED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
This represents a configuration where CRC is not supported.	
Template Description	
This indicates that CRC is not supported	
M2 Parameter	
GlobalTimeCrcSupportEnum.crcNotSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full



△

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTxCrcSecured	
BSW Parameter		**BSW Type**
CRC_SUPPORTED		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
This represents a configuration where CRC is supported.		
Template Description		
This indicates that CRC is supported		
M2 Parameter		
GlobalTimeCrcSupportEnum.crcSupported		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster	
BSW Parameter		**BSW Type**
FrTSynGlobalTimeTxIcvGeneration		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container collects configuration that shall be used for ICV generation.		
Template Description		
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.		
M2 Parameter		
GlobalTimeMaster.icvSecured		
Mapping Rule		**Mapping Type**
Create this container if GlobalTimeMaster.icvSecured is defined as the value GlobalTimeIcv SupportEnum.icvSupported.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTSyn_ 00049]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTxIcvGeneration	
BSW Parameter		**BSW Type**
FrTSynIcvGenerationFvIdRef		ECUC-REFERENCE-DEF
BSW Description		
This represents the reference to the FV taken to generate the ICV generation.		
Template Description		
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.		
M2 Parameter		

▽

△

GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeMaster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00050]

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster/FrTSynGlobalTimeTxIcvGeneration
BSW Parameter	**BSW Type**
FrTSynIcvGenerationJobRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the CSM job to fetch the CSM job ID.	
Template Description	
Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages.	
M2 Parameter	
GlobalTimeMaster.icvSecured	
Mapping Rule	**Mapping Type**
If GlobalTimeMaster.icvSecured is set to the value icvSupported, then the referenced CsmJob needs to be configured.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00053]

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeMaster
BSW Parameter	**BSW Type**
FrTSynGlobalTimeTxPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents the TX period. Unit: seconds	
Template Description	
This represents the period. Unit: seconds	
M2 Parameter	
GlobalTimeMaster.syncPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00014]

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
FrTSynGlobalTimeNetworkSegmentId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

This represents the numerical identifier of the network on system level scope where this Global Time has been communicated on.	
Template Description	
This attribute represents the numerical identifier of a PhysicalChannel on system level scope.	
M2 Parameter	
NetworkSegmentIdentification.networkSegmentId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00042]

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
FrTSynGlobalTimeSlave	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This represents the time slave for the enclosing global time domain.	
Template Description	
This represents the generic concept of a global time slave.	
M2 Parameter	
GlobalTimeSlave	
Mapping Rule	**Mapping Type**
The existence of the FrTSynGlobalTimeSlave is bound to the existence of a GlobalTimeSlave that refers to a CommunicationConnector that in turn in aggregated at the EcuInstance for which the ECU configuration exists,	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00010]

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
FrTSynGlobalTimeRxIcvVerification	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container collects configuration required for ICV verification.	
Template Description	
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.	
M2 Parameter	
GlobalTimeSlave.icvVerification	
Mapping Rule	**Mapping Type**
Create this container if GlobalTimeSlave.icvVerification is defined as the value GlobalTimeIcv VerificationEnum. icvVerified or GlobalTimeIcvVerificationEnum.icvOptional.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00056]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeRxIcvVerification	
BSW Parameter		**BSW Type**
FrTSynIcvVerificationAttempts		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter specifies the number of ICV verification attempts that are to be carried out when the verification of the ICV failed for a given secured SYNC message. If zero is set, then only one ICV verification attempt is done.		
Template Description		
This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given message. If zero is set than only one authentication attempt is done.		
M2 Parameter		
SecOcSecureComProps.authenticationVerifyAttempts		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTSyn_00062]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeRxIcvVerification	
BSW Parameter		**BSW Type**
FrTSynIcvVerificationFvIdRef		ECUC-REFERENCE-DEF
BSW Description		
This represents the reference to the FV taken to generate the ICV generation.		
Template Description		
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.		
M2 Parameter		
GlobalTimeDomain.icvFreshnessValueId		
Mapping Rule		**Mapping Type**
Reference to the StbMFreshnessValue created for this GlobalTimeSlave.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTSyn_00057]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeRxIcvVerification	
BSW Parameter		**BSW Type**
FrTSynIcvVerificationJobRef		ECUC-REFERENCE-DEF
BSW Description		
This represents the reference to the CSM job to fetch the CSM job ID.		
Template Description		
Defines how an Integrity Check Value (ICV) shall be handled at the receiver.		
M2 Parameter		
GlobalTimeSlave.icvVerification		
Mapping Rule		**Mapping Type**
If GlobalTimeSlave.icvVerification is set to the value icvVerified or icvOptional, then the referenced CsmJob needs to be configured.		full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_FrTSyn_00060]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
FrTSynGlobalTimeSequenceCounterJumpWidth	ECUC-INTEGER-PARAM-DEF
BSW Description	
The SequenceCounterJumpWidth specifies the maximum allowed gap of the Sequence Counter between two SYNC messages.	
Template Description	
Specifies the maximum allowed gap of the sequence counter between two SYNC messages.	
M2 Parameter	
GlobalTimeFrSlave.sequenceCounterJumpWidth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00022]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynGlobalTimeSlavePdu
BSW Parameter	**BSW Type**
FrTSynGlobalTimePduRef	ECUC-REFERENCE-DEF
BSW Description	
This represents the reference to the Pdu taken to transmit the global time information. The global time master of a global time domain acts as the sender of the Pdu while all the time slaves are supposed to receive the Pdu.	
Template Description	
This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated Global TimeSlaves.	
M2 Parameter	
GlobalTimeDomain.pduTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00020]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave
BSW Parameter	**BSW Type**
FrTSynRxCrcValidated	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter controls whether or not CRC validation shall be supported.	

▽

△

Template Description	
Definition of whether or not validation of the CRC is supported.	
M2 Parameter	
GlobalTimeFrSlave.crcValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00017]

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_IGNORED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts Time Synchronization messages, which are CRC secured (without actually validating the CRC) and those, which are not CRC secured. That means, the Timesync module ignores the CRC.	
Template Description	
The CRC is supposed to be ignored	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcIgnored	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_NOT_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are not CRC secured. All other Time Synchronization messages are ignored.	
Template Description	
The CRC is not supposed to be present. If CRC is present the message is ignored.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcNotValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrcValidated
BSW Parameter	**BSW Type**

▽

△

CRC_OPTIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages which are not CRC secured and Time Synchronization messages which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Template Description	
Either the CRC is present and then shall be validated or the CRC is not present and no CRC check is done.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcOptional	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSlave/FrTSynRxCrcValidated
BSW Parameter	**BSW Type**
CRC_VALIDATED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The Timesync module accepts only Time Synchronization messages, which are CRC secured and have the correct CRC. All other Time Synchronization messages are ignored.	
Template Description	
This CRC is supposed to be validated.	
M2 Parameter	
GlobalTimeCrcValidationEnum.crcValidated	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain
BSW Parameter	**BSW Type**
FrTSynGlobalTimeSyncDataIDList	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.	
Template Description	
The DataIDList for SYNC messages to calculate CRC.	
M2 Parameter	
FrGlobalTimeDomainProps.syncDataIDList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTSyn_00023]

BSW Module	BSW Context	
FrTSyn	FrTSyn/FrTSynGlobalTimeDomain/FrTSynGlobalTimeSyncDataIDList	
BSW Parameter		**BSW Type**
FrTSynGlobalTimeSyncDataIDListElement		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Element of the DataIDList for SYNC messages ensures the identification of data elements due to CRC calculation and message authentication process.		
Template Description		
The DataIDList for SYNC messages to calculate CRC.		
M2 Parameter		
FrGlobalTimeDomainProps.syncDataIDList		
Mapping Rule		**Mapping Type**
Value shall be derived from element of the ordered syncDataIDList.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTSyn_00025]

C.30 FrTp

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpFullDuplexEnable		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Preprocessor switch for enabling full duplex mechanisms for all channels. True: Full duplex is enabled False: Fullduplex is disabled (Half duplex is enabled)		
Template Description		
The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.		
M2 Parameter		
FlexrayTpEcu.fullDuplexEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTp_00051]

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpMainFuncCycle		ECUC-FLOAT-PARAM-DEF
BSW Description		
This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.		
Template Description		
The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.		

▽

△

M2 Parameter	
FlexrayTpEcu.cycleTimeMainFunction	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00011]

BSW Module	BSW Context
FrTp	FrTp/FrTpGeneral
BSW Parameter	**BSW Type**
FrTpTransmitCancellation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation. True: Transmit/Receive Cancellation is enabled False: Transmit/Receive Cancellation is disabled	
Template Description	
With this switch Tx and Rx Cancellation can be turned on or off.	
M2 Parameter	
FlexrayTpEcu.cancellation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00036]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig
BSW Parameter	**BSW Type**
FrTpConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the connection specific parameters to transfer N-PDUs via FlexRay TP.	
Template Description	
A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
FlexrayTpConnection	
Mapping Rule	**Mapping Type**
Create container for each FlexRayTpConnection that is described in the ECU Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00006]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**

▽

△

FrTpBandwidthLimitation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter indicates whether the connection requires a bandwidth limitation or not. If FrTpBandwidthLimitation=True the sender shall send a StartFrame always on the first PDU of a PDU-Pool.	
Template Description	
Specifies whether the connection requires a bandwidth limitation or not.	
M2 Parameter	
FlexrayTpConnection.bandwidthLimitation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00050]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpConCtrlRef	ECUC-REFERENCE-DEF
BSW Description	
FrTpConnectionControlReference: This parameter defines a reference to a connection control container.	
Template Description	
M2 Parameter	
FlexrayTpConnection.tpConnectionControl	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00005]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpLa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the Local Address for the respective connection. When the local instance is the sender, this is the Source Address within the TP frame. When the local instance is the receiver, this is the Target Address within the TP frame. If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16.	
Template Description	
M2 Parameter	
FlexrayTpConnection.transmitter, FlexrayTpConnection.receiver	
Mapping Rule	**Mapping Type**
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00010]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpMultipleReceiverCon	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection. If data segmentation is required this parameter is used to check whether segmentation is possible or not. If the connection is 1:n segmentation is not possible and an error will occur.	
Template Description	
M2 Parameter	
FlexrayTpConnection.multicast	
Mapping Rule	**Mapping Type**
If FlexRayTpConnection contains a mulicast reference to TpAddress than set this parameter to true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00019]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpRa	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame. If this parameter is not configured, all related Rx N-SDUs must be configured to use the meta data item SOURCE_ADDRESS_16, and all related Tx-N-SDUs must be configured to use the meta data item TARGET_ADDRESS_16.	
Template Description	
M2 Parameter	
FlexrayTpConnection.transmitter, FlexrayTpConnection.receiver	
Mapping Rule	**Mapping Type**
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00021]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpRxPduPoolRef	ECUC-REFERENCE-DEF
BSW Description	
This parameter defines a reference to a RxPduPool.	
Template Description	
M2 Parameter	
FlexrayTpConnection.rxPduPool, FlexrayTpConnection.txPduPool	
Mapping Rule	**Mapping Type**

▽

△

Depending whether the regarded Ecu is the transmitter or the receiver this reference shall be created if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the txPdu Pool or rxPduPool reference. If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus. If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00025]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpRxSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This parameter defines the Rx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR. This N-SDU can produce meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.	
Template Description	

M2 Parameter	
FlexrayTpConnection.directTpSdu	
Mapping Rule	**Mapping Type**
Create container if an Rx Pdu is referenced by the FlexRayTpConnection	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00027]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpTxPduPoolRef	ECUC-REFERENCE-DEF
BSW Description	
This parameter defines a reference to a TxPduPool.	
Template Description	

M2 Parameter	
FlexrayTpConnection.rxPduPool, FlexrayTpConnection.txPduPool	
Mapping Rule	**Mapping Type**
Depending whether the regarded Ecu is the transmitter or the receiver this reference shall be created if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the txPdu Pool or rxPduPool reference. If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus. If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00039]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection
BSW Parameter	**BSW Type**
FrTpTxSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This parameter defines the Tx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR. This N-SDU can consume meta data items of type SOURCE_ADDRESS_16 and TARGET_ADDRESS_16.	
Template Description	
M2 Parameter	
FlexrayTpConnection.directTpSdu	
Mapping Rule	**Mapping Type**
Create container if an Tx Pdu is referenced by the FlexRayTpConnection	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00041]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig
BSW Parameter	**BSW Type**
FrTpConnectionControl	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters to control a FlexRay TP connection.	
Template Description	
Configuration parameters to control a FlexRay TP connection.	
M2 Parameter	
FlexrayTpConnectionControl	
Mapping Rule	**Mapping Type**
Create container for each FlexRayTpConnectionControl that is described in the ECU Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00007]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpAckType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the type of acknowledgement which is used for the specific channel.	
Template Description	
This parameter defines the type of acknowledgement which is used for the specific channel.	
M2 Parameter	
FlexrayTpConnectionControl.ackType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00003]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxFCWait	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the maximum number of FlowControl N-PDUs with FlowState "WAIT"	
Template Description	
This attribute defines the maximum number of FlowControl N-PDUs with FlowState "WAIT".	
M2 Parameter	
FlexrayTpConnectionControl.maxFcWait	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00014]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxNbrOfNPduPerCycle	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
Template Description	
This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
M2 Parameter	
FlexrayTpConnectionControl.maxNumberOfNpduPerCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00029]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxRn	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the maximum number of retries (if retry is configured).	
Template Description	
This parameter defines the maximum number of retries (if retry is configured for the particular channel).	
M2 Parameter	
FlexrayTpConnectionControl.maxRetries	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00017]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpSCexp	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It represents the exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.	
Template Description	
Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.	
M2 Parameter	
FlexrayTpConnectionControl.separationCycleExponent	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00020]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeBr	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines the time in seconds the FrTp requires to transmit a corresponding FlowControl Frame. According to ISO 10681-2 this parameter is a performance requirement.	
Template Description	
Time (in seconds) until transmission of the next FlowControl N-PDU.	
M2 Parameter	
FlexrayTpConnectionControl.timeBr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00047]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeCs	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines the time in seconds between the sending of two CFs or between the sending of a CF and LF or between the reception of a FC and sending of the next CF.	
Template Description	
Time (in seconds) until transmission of the next ConsecutiveFrame NPdu / LastFrame NPdu.	
M2 Parameter	
FlexrayTpConnectionControl.timeCs	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00056]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutAr	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
Template Description	
This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.	
M2 Parameter	
FlexrayTpConnectionControl.timeoutAr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00032]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutAs	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter specifies the timeout in seconds the FrIf shall confirm a transmitted Pdu to the FrTp via the LSduR.	
Template Description	
This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.	
M2 Parameter	
FlexrayTpConnectionControl.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00033]

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutBs	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	

▽

△

Template Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
M2 Parameter	
FlexrayTpConnectionControl.timeoutBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00034]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutCr	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines the timeout value in seconds a receiver is waiting for a CF or a LF.	
Template Description	
This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.	
M2 Parameter	
FlexrayTpConnectionControl.timeoutCr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00035]

BSW Module	BSW Context
FrTp	FrTp/FrTpMultipleConfig
BSW Parameter	**BSW Type**
FrTpRxPduPool	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains all Pdus that are assigned to that Pdu Pool.	
Template Description	
FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.	
M2 Parameter	
FlexrayTpPduPool	
Mapping Rule	**Mapping Type**
Create container if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the rxPduPool or txPduPool reference. If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus. If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00024]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool	
BSW Parameter	**BSW Type**	
FrTpRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
Container to hold the PDU parameters. ImplementationType: PduInfoType		
Template Description		
M2 Parameter		
FlexrayTpPduPool.nPdu		
Mapping Rule		**Mapping Type**
Create container for each NPdu that is referenced by the regarded FlexrayTpPduPool.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTp_00022]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter	**BSW Type**	
FrTpTxPduPool	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
This container contains all Pdus that are assigned to that Pdu Pool.		
Template Description		
FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.		
M2 Parameter		
FlexrayTpPduPool		
Mapping Rule		**Mapping Type**
Create container if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the rxPduPool or txPduPool reference. If the regarded ECU is the transmitter then the txPduPool holds the sent NPdus and the rxPduPool holds the received NPdus. If the ECU is the receiver then the txPduPool holds the received NPdus and the rxPduPool holds the sent NPdus.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_FrTp_00038]

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool	
BSW Parameter	**BSW Type**	
FrTpTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
Container to hold the PDU parameters. ImplementationType: PduInfoType		
Template Description		
M2 Parameter		
FlexrayTpPduPool.nPdu		
Mapping Rule		**Mapping Type**

▽

△

Create container for each NPdu that is referenced by the regarded FlexrayTpPduPool.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTp_00037]

C.31 FrTrcv

BSW Module	**BSW Context**
FrTrcv	FrTrcv/FrTrcvChannel
BSW Parameter	**BSW Type**
FrTrcvWakeupByBusUsed	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Is wake up by node supported? If FlexRay transceiver hardware does not support wake up by node value is always FALSE. If FlexRay transceiver hardware supports wake up by node value is TRUE or FALSE depending whether it is used or not.	
Template Description	
Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
`CommunicationController.wakeUpByControllerSupported`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_FrTrcv_00350]

C.32 IEEE1722Tp

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig
BSW Parameter	**BSW Type**
IEEE1722TpStream	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722Tp stream.	
Template Description	
Definition of the IEEE1722Tp protocol.	
M2 Parameter	
`IEEE1722TpConnection`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**

▽

△

draft	[ECUC_IEEE1722Tp_00006]

BSW Module	BSW Context	
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream	
BSW Parameter		**BSW Type**
IEEE1722TpStbMSynchronizedTimeBaseRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the StbMSynchronizedTimeBase from which the current synchronized time could be retrieved (e.g. determine avtp timestamp for transmission).		
Template Description		
Reference to the GlobalTimeDomain this IEEE1722TpConnection shall be synchronized with.		
M2 Parameter		
IEEE1722TpConnection.globalTimeDomain		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_IEEE1722Tp_00100]

BSW Module	BSW Context	
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream	
BSW Parameter		**BSW Type**
IEEE1722TpStreamIdMacAddress		ECUC-STRING-PARAM-DEF
BSW Description		
Definition of the MAC address part of the Stream Id.		
Template Description		
MAC Address part of the Stream Id.		
M2 Parameter		
IEEE1722TpConnection.macAddressStreamId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_IEEE1722Tp_00008]

BSW Module	BSW Context	
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream	
BSW Parameter		**BSW Type**
IEEE1722TpStreamIdUniquePart		ECUC-INTEGER-PARAM-DEF
BSW Description		
Definition of the unique ID part of the Stream Id.		
Template Description		
Unique Id part of the Stream Id.		
M2 Parameter		

▽

△

IEEE1722TpConnection.uniqueStreamId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00007]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream
BSW Parameter	**BSW Type**
IEEE1722TpStreamMaxTransitTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Definition of the max transit time for the stream. Value in seconds.	
Template Description	
IEEE1722TpAcfConnection.acfMaxTransitTime: Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers. **IEEE1722TpAvConnection.maxTransitTime:** Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.	
M2 Parameter	
IEEE1722TpAcfConnection.acfMaxTransitTime, IEEE1722TpAvConnection.maxTransitTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00033]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype
BSW Parameter	**BSW Type**
IEEE1722TpStreamAAF	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722 AVTP Audio Format (AAF) stream.	
Template Description	
AV IEEE1722Tp AAF connection.	
M2 Parameter	
IEEE1722TpAafConnection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00011]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafAes3DataTypeH	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the AAF AES3 aes3_data_type_h default value.	
Template Description	
Definition of the AAF AES3 aes3_data_type_h default value.	
M2 Parameter	
IEEE1722TpAafConnection.aes3DataTypeH	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00049]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafAes3DataTypeL	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the AAF AES3 aes3_data_type_l default value.	
Template Description	
Definition of the AAF AES3 aes3_data_type_l default value.	
M2 Parameter	
IEEE1722TpAafConnection.aes3DataTypeL	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00063]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafAes3DataTypeRef	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the AAF AES3 stream aes3_data_type reference (aes3_dt_ref).	
Template Description	
Definition of the AAF AES3 stream aes3_data_type reference.	
M2 Parameter	
IEEE1722TpAafConnection.aafAes3DataType	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00062]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafAes3NominalFrameRate	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the AAF AES3 stream nominal frame rate (nfr).	
Template Description	
Definition of the AAF stream nominal sample / frame rate. For an AAF PCM stream this is the nominal sample rate. For an AAF AES3 stream this is the nominal frame rate.	
M2 Parameter	
IEEE1722TpAafConnection.aafNominalRate	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00060]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafAes3StreamsPerFrame	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the AAF AES3 stream streams_per_frame.	
Template Description	
AAF AES3 stream streams_per_frame.	
M2 Parameter	
IEEE1722TpAafConnection.streamsPerFrame	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00061]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafEventDefault	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Definition of the AAF stream event (evt) default value.	
Template Description	
Definition of a value to be used for the 4-bit "evt" field.	
M2 Parameter	
IEEE1722TpAafConnection.eventDefaultValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00048]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafFormat	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the AAF stream format.	
Template Description	
Definition of the AAF stream format.	
M2 Parameter	
IEEE1722TpAafConnection.aafFormat	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00043]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAafPcmBitDepth	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the AAF PCM stream bit_depth. e.g. 16, 24, 32	
Template Description	
Definition of the AAF PCM stream bit_depth. e.g. 16, 24, 32.	
M2 Parameter	
IEEE1722TpAafConnection.pcmBitDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00047]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF

BSW Parameter	BSW Type
IEEE1722TpStreamAafPcmChannelsPerFrame	ECUC-INTEGER-PARAM-DEF

BSW Description

Definition of the AAF PCM stream channels_per_frame.

e.g. 1: mono, 2: stereo, 8: 7.1 multicannel

Template Description

Definition of the AAF PCM stream channels_per_frame.

e.g. 1: mono, 2: stereo, 8: 7.1 multicannel

M2 Parameter

IEEE1722TpAafConnection.channelsPerFrame

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00045]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF

BSW Parameter	BSW Type
IEEE1722TpStreamAafPcmNominalSampleRate	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Definition of the AAF PCM stream nominal sample rate (nsr).

Template Description

Definition of the AAF stream nominal sample / frame rate.

For an AAF PCM stream this is the nominal sample rate.

For an AAF AES3 stream this is the nominal frame rate.

M2 Parameter

IEEE1722TpAafConnection.aafNominalRate

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00044]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamAAF

BSW Parameter	BSW Type
IEEE1722TpStreamAafSparseTimestamp	ECUC-BOOLEAN-PARAM-DEF

BSW Description

▽

△

Defines the CRF stream sparce timestamp (sp). true = 1 false = 0	
Template Description	
Defines whether the "sp" (sparse timestamp) shall be enabled. false: Normal operation, timestamp in every AAF AVTPDU true: Sparse mode, timestamp in every eighth AAF AVTPDU	
M2 Parameter	
IEEE1722TpAafConnection.sparseTimestampEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00046]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype
BSW Parameter	**BSW Type**
IEEE1722TpStreamACF	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722 AVTP Control Format (ACF) stream.	
Template Description	
ACF IEEE1722Tp connection.	
M2 Parameter	
IEEE1722TpAcfConnection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00014]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF
BSW Parameter	**BSW Type**
IEEE1722TpAcfCollectionThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the size threshold in bytes which, when exceeded, triggers the sending of the ACF message.	
Template Description	
Defines the size threshold in bytes which, when exceeded, triggers the sending of the IEEE1722Tp ACF message, even when the maxium IEEE1722Tp ACF message size has not been reached yet.	
M2 Parameter	
IEEE1722TpAcfConnection.collectionThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_IEEE1722Tp_00094]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF
BSW Parameter	**BSW Type**
IEEE1722TpAcfCollectionTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines a timeout which, when exceeded, triggers the sending of the ACF message. Defined in seconds.	
Template Description	
When this timeout expires the IEEE1722Tp ACF message is triggered for sending. The respective timer is started when the first Pdu is put into the IEEE1722Tp ACF message. Defined in seconds.	
M2 Parameter	
IEEE1722TpAcfConnection.collectionTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00093]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfHeaderType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the ACF stream header format. Depending on this selection the AVTP stream data subtype will be defined.	
Template Description	
Defines the time offset that is added to the current time at the producer in order to get the "presentation time" (in seconds) when content shall be presented at the consumers.	
M2 Parameter	
IEEE1722TpAcfConnection.acfMaxTransitTime	
Mapping Rule	**Mapping Type**
If IEEE1722TpAcfConnection.acfMaxTransitTime is defined, then IEEE1722TpStreamAcfHeader Type shall be set to TIME_SYNCHRONOUS, otherwise IEEE1722TpStreamAcfHeaderType shall be set to NON_TIME_SYNCHRONOUS.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00072]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfMixedBusTypeCollection	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines if this ACF-stream is allowed to collect ACF-messages of different bus kinds (i.e. whether it is allowed to collect CAN and LIN ACF-messages in one ACF-stream message). For the ACF-stream producer this configures the collection behavior. For an ACF-stream consumer this configures that this ACF-stream was produced with this assumption. true = 1: mixed collection is allowed false = 0 : mixed collection is not allowed	
Template Description	
Defines if this ACF-stream is allowed to collect ACF-messages of different bus kinds (i.e. whether it is allowed to collect CAN and LIN ACF-messages in one ACF-stream message).	
M2 Parameter	
IEEE1722TpAcfConnection.mixedBusTypeCollection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00110]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCan	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for CAN and CAN_BRIEF.	
Template Description	
ACF IEEE1722Tp bus used for CAN transport.	
M2 Parameter	
IEEE1722TpAcfCan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00074]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfBusId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the ACF stream bus_id.	

▽

△

Template Description	
Id of the transported bus over the ACF connection.	
M2 Parameter	
IEEE1722TpAcfBus.busId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00078]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanMessageType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the ACF CAN stream message type. Depending on this selection the ACF stream acf_msg_type will be defined.	
Template Description	
Definition of the ACF CAN stream message type.	
M2 Parameter	
IEEE1722TpAcfCan.messageType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00081]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of a CAN Pdu transported on this ACF stream. Identification can be done by either explicit CAN Id or via meta-data.	
Template Description	
Definition of one CAN part (frame or frame range) transported over the IEEE1722Tp channel.	
M2 Parameter	
IEEE1722TpAcfCanPart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00082]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanExtendedFrameFormat	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the ACF CAN stream extended_frame_format (eff). true = 1 false = 0	
Template Description	
Defines whether standard or extended address format shall be used.	
M2 Parameter	
IEEE1722TpAcfCanPart.canAddressingMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00077]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanFDBitDataRateFormat	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Represents the CAN-FD Data-rate (FD) format (fdf) switch. This is used, when a PDU is produced by an upper layer and transferred as encapsulated ACF_CAN via an Ethernet network.	
Template Description	
Defines which CAN protocol shall be used for frame transmission.	
M2 Parameter	
IEEE1722TpAcfCanPart.canFrameTxBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00099]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanFDBitRateSwitch	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Represents the CAN-FD bit rate switch (brs). This is used, if a PDU is produced by an upper layer and transferred as encapsulated ACF_CAN via an Ethernet network.	



△

Template Description	
Defines whether the bit rate switch bit shall be set.	
M2 Parameter	
IEEE1722TpAcfCanPart.canBitRateSwitch	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00098]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the ACF stream CAN Id in case CAN Id is not taken from meta-data.	
Template Description	
Optional Can Id defined in case the Can Id can not be determined during runtime.	
M2 Parameter	
IEEE1722TpAcfCanPart.canIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00079]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfCanPduFilter
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter defines a CAN Identifier range in an alternative way to IEEE1722TpStreamAcfCanIdRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier. Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.	
Template Description	
CAN identifier mask which denotes relevant bits in the CAN Identifier. This attribute defines a CAN Identifier range in an alternative way to canIdentifierRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier.	
M2 Parameter	
IEEE1722TpAcfCanPart.canIdentifierMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00083]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfCanPduFilter

BSW Parameter	BSW Type
IEEE1722TpStreamAcfCanIdRange	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Identifier range definition. This parameter defines a CAN Identifier range in an alternative way to IEEE1722TpStreamAcfCanIdMask.

Template Description
Definition of the identifier range for IEEE1722Tp ACF Can messages.

M2 Parameter
IEEE1722TpAcfCanPart.canIdentifierRange

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IEEE1722Tp_00088]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfCanPduFilter/IEEE1722TpStreamAcfCanIdRange

BSW Parameter	BSW Type
IEEE1722TpStreamAcfCanIdRangeLowerCanId	ECUC-INTEGER-PARAM-DEF

BSW Description
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.

Template Description
This attribute can be used together with the upperCanId attribute to define a range of CanIds.

M2 Parameter
RxIdentifierRange.lowerCanId

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IEEE1722Tp_00090]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfCanPduFilter/IEEE1722TpStreamAcfCanIdRange

BSW Parameter	BSW Type
IEEE1722TpStreamAcfCanIdRangeUpperCanId	ECUC-INTEGER-PARAM-DEF

BSW Description
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.

Template Description
This attribute can be used together with the lowerCanId attribute to define a range of CanIds.



△

M2 Parameter	
RxIdentifierRange.upperCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IEEE1722Tp_00089]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfRxPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the EcuC Pdu.	
Template Description	
IEEE1722TpAcfCanPart.sdu: Reference to the Pdu transported in the IEEE1722Tp channel. **IEEE1722TpAcfLinPart.sdu:** Reference to the Pdu transported in the IEEE1722Tp channel.	
M2 Parameter	
IEEE1722TpAcfCanPart.sdu, IEEE1722TpAcfLinPart.sdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00086]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfTriggerMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines whether putting this Can/Lin Pdu to the ACF message • always or • never triggers immediate sending of the ACF message.	
Template Description	
Defines whether putting this AcfPart to the IEEE1722Tp ACF message triggers immediate sending of the IEEE1722Tp ACF message.	
M2 Parameter	
IEEE1722TpAcfBusPart.collectionTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_IEEE1722Tp_00095]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfCan/IEEE1722TpStreamAcfCanPdu/IEEE1722TpStreamAcfTxPdu

BSW Parameter	BSW Type
IEEE1722TpStreamAcfPduRef	ECUC-REFERENCE-DEF

BSW Description

Reference to the EcuC Pdu.

Template Description

IEEE1722TpAcfCanPart.sdu:
Reference to the Pdu transported in the IEEE1722Tp channel.

IEEE1722TpAcfLinPart.sdu:
Reference to the Pdu transported in the IEEE1722Tp channel.

M2 Parameter

IEEE1722TpAcfCanPart.sdu, IEEE1722TpAcfLinPart.sdu

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00086]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload

BSW Parameter	BSW Type
IEEE1722TpStreamAcfLin	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Definition of an IEEE1722Tp AVTP Control Format (ACF) stream payload for LIN.

ACF message type = 0x03.

Template Description

ACF IEEE1722Tp bus used for LIN transport.

M2 Parameter

IEEE1722TpAcfLin

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00076]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin

▽

△

BSW Parameter	BSW Type
IEEE1722TpStreamAcfBusId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the ACF stream bus_id.	
Template Description	
Id of the transported bus over the ACF connection.	
M2 Parameter	
IEEE1722TpAcfBus.busId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00078]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfLinPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of a LIN Pdu transported on this ACF stream. Identification can be done by either explicit LIN Id or via meta-data.	
Template Description	
Definition of one LIN part transported over the IEEE1722Tp channel.	
M2 Parameter	
IEEE1722TpAcfLinPart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00096]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin/IEEE1722TpStreamAcfLinPdu
BSW Parameter	**BSW Type**
IEEE1722TpStreamAcfLinId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the ACF stream Lin Id.	
Template Description	
Optional Lin Id defined in case the Lin Id can not be determined during runtime.	
M2 Parameter	
IEEE1722TpAcfLinPart.linIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00080]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin/IEEE1722TpStreamAcfLinPdu/IEEE1722TpStreamAcfRxPdu

BSW Parameter	BSW Type
IEEE1722TpStreamAcfPduRef	ECUC-REFERENCE-DEF

BSW Description

Reference to the EcuC Pdu.

Template Description

IEEE1722TpAcfCanPart.sdu:
Reference to the Pdu transported in the IEEE1722Tp channel.

IEEE1722TpAcfLinPart.sdu:
Reference to the Pdu transported in the IEEE1722Tp channel.

M2 Parameter

IEEE1722TpAcfCanPart.sdu, IEEE1722TpAcfLinPart.sdu

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00086]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin/IEEE1722TpStreamAcfLinPdu

BSW Parameter	BSW Type
IEEE1722TpStreamAcfTriggerMode	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Defines whether putting this Can/Lin Pdu to the ACF message

- always or
- never

triggers immediate sending of the ACF message.

Template Description

Defines whether putting this AcfPart to the IEEE1722Tp ACF message triggers immediate sending of the IEEE1722Tp ACF message.

M2 Parameter

IEEE1722TpAcfBusPart.collectionTrigger

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00095]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamACF/IEEE1722TpStreamAcfPayload/IEEE1722TpStreamAcfLin/IEEE1722TpStreamAcfLinPdu/IEEE1722TpStreamAcfTxPdu

BSW Parameter	BSW Type



△

IEEE1722TpStreamAcfPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the EcuC Pdu.	
Template Description	
IEEE1722TpAcfCanPart.sdu: Reference to the Pdu transported in the IEEE1722Tp channel. **IEEE1722TpAcfLinPart.sdu:** Reference to the Pdu transported in the IEEE1722Tp channel.	
M2 Parameter	
IEEE1722TpAcfCanPart.sdu, IEEE1722TpAcfLinPart.sdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00086]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype
BSW Parameter	**BSW Type**
IEEE1722TpStreamCRF	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722 Clock Reference Format (CRF) stream.	
Template Description	
AV IEEE1722Tp CRF connection.	
M2 Parameter	
IEEE1722TpCrfConnection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00010]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamCRF
BSW Parameter	**BSW Type**
IEEE1722TpStreamCrfBaseFrequency	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the CRF stream base frequency. This is defined in Hz.	
Template Description	
CRF base frequency in Hz.	
M2 Parameter	
IEEE1722TpCrfConnection.baseFrequency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00042]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamCRF

BSW Parameter	BSW Type
IEEE1722TpStreamCrfFrameSync	ECUC-BOOLEAN-PARAM-DEF

BSW Description

Defines the CRF stream frame sync (fs).

true = 1

false = 0

Template Description

Defines whether the "fs" (frame sync) shall be enabled.

M2 Parameter

`IEEE1722TpCrfConnection.frameSyncEnabled`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00038]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamCRF

BSW Parameter	BSW Type
IEEE1722TpStreamCrfPull	ECUC-ENUMERATION-PARAM-DEF

BSW Description

Definition of the CRF stream pull value.

Template Description

Definition of the CRF stream pull value.

M2 Parameter

`IEEE1722TpCrfConnection.crfPull`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00041]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamCRF

BSW Parameter	BSW Type
IEEE1722TpStreamCrfTimestampInterval	ECUC-INTEGER-PARAM-DEF

BSW Description

Definition of the CRF stream timestamp interval. This is defined as multiple of base frequency.

Template Description

CRF timestamp interval as multiple of the baseFrequency.

M2 Parameter

`IEEE1722TpCrfConnection.timestampInterval`



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00058]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamCRF
BSW Parameter	**BSW Type**
IEEE1722TpStreamCrfType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the CRF stream type.	
Template Description	
Definition of the CRF stream type.	
M2 Parameter	
IEEE1722TpCrfConnection.crfType	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00040]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype
BSW Parameter	**BSW Type**
IEEE1722TpStreamIIDC	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Definition of an IEEE1722 61883_IIDC (IEC 61883/IIDC over AVTP) stream.	
Template Description	
AV IEEE1722Tp IIDC and IEC61883 connection.	
M2 Parameter	
IEEE1722TpIidcConnection	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00012]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcChannel	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Definition of the IIDC and IEC61883 channel.	
Template Description	
Definition of the IIDC and IEC61883 channel.	
M2 Parameter	
IEEE1722TpIidcConnection.iidcChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00051]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcDataBlockSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IEC61883 data block size (DBS).	
Template Description	
Definition of the IEC61883 data block size (DBS).	
M2 Parameter	
IEEE1722TpIidcConnection.iidcDataBlockSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00053]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcFractionNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IEC61883 fractionNumber (FN).	
Template Description	
Definition of the IEC61883 fractionNumber (FN).	
M2 Parameter	
IEEE1722TpIidcConnection.iidcFractionNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00054]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcSourcePacketHeader	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the IEC61883 source packet header (SPH). true = 1 false = 0	
Template Description	
Defines the IEC61883 source packet header (SPH) existence.	
M2 Parameter	
IEEE1722TpIidcConnection.iidcSourcePacketHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00055]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcStreamFormat	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IEC61883 stream format (FMT).	
Template Description	
Definition of the IEC61883 stream format (FMT).	
M2 Parameter	
IEEE1722TpIidcConnection.iidcStreamFormat	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00056]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcSy	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IIDC and IEC61883 sy.	
Template Description	
Definition of the IIDC and IEC61883 sy.	
M2 Parameter	
IEEE1722TpIidcConnection.iidcSy	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00059]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcTCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IIDC and IEC61883 tcode.	
Template Description	
Definition of the IIDC and IEC61883 tcode.	
M2 Parameter	
IEEE1722TpIidcConnection.iidcTCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00052]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamIIDC
BSW Parameter	**BSW Type**
IEEE1722TpStreamIidcTag	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the IIDC and IEC61883 tag.	
Template Description	
Definition of the IIDC and IEC61883 tag.	
M2 Parameter	
IEEE1722TpIidcConnection.iidcTag	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00050]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype
BSW Parameter	**BSW Type**
IEEE1722TpStreamRVF	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Definition of an IEEE1722 Raw Video Format (RVF) stream.	
Template Description	
AV IEEE1722Tp RVF connection.	
M2 Parameter	
IEEE1722TpRvfConnection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00013]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfActivePixels	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the RVF stream active_pixels.	
Template Description	
Definition of the RVF stream active_pixels.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfActivePixels	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00064]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfColorSpace	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the RVF stream colorspace.	
Template Description	
Definition of the RVF stream colorspace.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfColorSpace	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00071]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfEventDefault	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the RVF stream event (evt) default value.	
Template Description	
Definition of the RVF stream event (evt) default value.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfEventDefault	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00066]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfFrameRate	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the RVF stream frame_rate.	
Template Description	
Definition of the RVF stream frame_rate.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfFrameRate	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00070]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfInterlaced	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines the RVF stream interlaced (i). true = 1 false = 0	
Template Description	
Defines the RVF stream interlaced (i).	
M2 Parameter	
IEEE1722TpRvfConnection.rvfInterlaced	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00067]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfPixelDepth	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the RVF stream pixel_depth.	
Template Description	
Definition of the RVF stream pixel_depth.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfPixelDepth	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00068]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfPixelFormat	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the RVF stream pixel_format.	
Template Description	
Definition of the RVF stream pixel_format.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfPixelFormat	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00069]

BSW Module	BSW Context
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream/IEEE1722TpStreamSubtype/IEEE1722TpStreamRVF
BSW Parameter	**BSW Type**
IEEE1722TpStreamRvfTotalLines	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Definition of the RVF stream total_lines.	
Template Description	
Definition of the RVF stream total_lines.	
M2 Parameter	
IEEE1722TpRvfConnection.rvfTotalLines	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00065]

BSW Module	**BSW Context**
IEEE1722Tp	IEEE1722Tp/IEEE1722TpConfig/IEEE1722TpStream
BSW Parameter	**BSW Type**
IEEE1722TpStreamVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Definition of the stream version.	
Template Description	
Version of the IEEE1722TP stream.	
M2 Parameter	
IEEE1722TpConnection.version	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IEEE1722Tp_00039]

C.33 IKE

BSW Module	**BSW Context**
IKE	IKE/IKEConnections/IKEConnection
BSW Parameter	**BSW Type**
IKEPqPresharedKey	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container with the post-quantum preshared key configuration.	
Template Description	
Post-quantum pre-shared key configuration to support IETF RFC 8784 to ensure resistance against post-quantum attacks by combining a post-quantum pre-shared key (PPK) with the exchanged information (via e.g., DH/ECDH) during key derivation.	
M2 Parameter	
IPSecRule.localIkePqPresharedKeySupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**

▽

△

valid	[ECUC_IKE_00090]

BSW Module	**BSW Context**
IKE	IKE/IKEConnections/IKEConnection/IKEPqPresharedKey
BSW Parameter	**BSW Type**
IKEPpkListElem	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
List element with the PPK and the corresponding PPK_ID.	
Template Description	
List of post-quantum pre-shared keys (PPK) and their post-quantum pre-shared key identifiers (PKK_IDs).	
M2 Parameter	
`LocalPqPresharedKeySupport.ikePqPresharedKey`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IKE_00092]

BSW Module	**BSW Context**
IKE	IKE/IKEConnections/IKEConnection/IKEPqPresharedKey/IKEPpkListElem
BSW Parameter	**BSW Type**
IKEPpkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
PPK_ID post-quantum preshared key identifier.	
Template Description	
This attribute defines the post-quantum pre-shared key identifier for the referenced pqPresharedKey.	
M2 Parameter	
`IkePqPresharedKey.pqPresharedKeyId`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IKE_00093]

BSW Module	**BSW Context**
IKE	IKE/IKEConnections/IKEConnection/IKEPqPresharedKey/IKEPpkListElem
BSW Parameter	**BSW Type**
IKEPpkRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to PPK post-quantum preshared key.	
Template Description	
Reference to PPK post-quantum pre-shared key.	
M2 Parameter	
`IkePqPresharedKey.pqPresharedKey`	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IKE_00094]

BSW Module	**BSW Context**
IKE	IKE/IKEConnections/IKEConnection/IKEPqPresharedKey
BSW Parameter	**BSW Type**
IKEPpkMandatory	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Boolean defines whether the usage of the PPK is mandatory or not. RFC 8784 parameter mandatory_or_not	
Template Description	
This boolean attribute defines whether the usage of the post-quantum pre-shared keys is mandatory or not for communication with the local node.	
M2 Parameter	
LocalPqPresharedKeySupport.remotePqPresharedKeyMandatorySupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IKE_00091]

C.34 ISO15118Chrg

BSW Module	**BSW Context**
ISO15118Chrg	ISO15118Chrg/ISO15118ChrgV2GTP
BSW Parameter	**BSW Type**
ISO15118ChrgV2GTPPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters of the ISO15118 module's Pdus that are exchanged between ISO15118 and the PduR.	
Template Description	
Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed.	
M2 Parameter	
ISignalIPdu	
Mapping Rule	**Mapping Type**
create container for each SignalIPdu that is transmitted by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_ISO15118 Chrg_00035]

C.35 IdsM

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMPdus/IdsMIfTxPdu
BSW Parameter	**BSW Type**
IdsMIfTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the IF PDU used for transmission of the QSEvs.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**
IdsMIfTxPduRef shall reference the only one existing (M2) GeneralPurposeIPdu of category "IDS" which is not mapped by (M2) IPduMapping and not referenced by any (M2) TP configuration element.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00042]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMPdus/IdsMTpTxPdu
BSW Parameter	**BSW Type**
IdsMTpTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the TP PDU used for transmission of the QSEvs.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**
IdsMTpTxPduRef shall reference the only one existing (M2) GeneralPurposeIPdu of category "IDS" which is not mapped by (M2) IPduMapping but referenced by the (M2) TP configuration element corresponding to the bus on which the IdsM shall send the IDS messages.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00045]

C.36 IpduM

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	**BSW Type**
IpduMContainedPduOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Static offset (in bytes) of the ContainedPdu.	
Template Description	
Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.	
M2 Parameter	
ContainedIPduProps.offset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IpduM_00206]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	**BSW Type**
IpduMContainedRxInContainerPduRef	ECUC-REFERENCE-DEF
BSW Description	
Optional reference to an IpduMContainerRxPdu this IpduMContainedRxPdu may be received in. If this IpduMContainedRxPdu shall be received in exactly one IpduMContainerRxPdu with IpduMContainerRxAccept ContainedPdu=IPDUM_ACCEPT_CONFIGURED then the IpduMContainedRxInContainerPduRef shall be defined. If this IpduMContainedRxPdu can be received in any IpduMContainerRxPdu with IpduMContainerRxAcceptContained Pdu=IPDUM_ACCEPT_ALL then the IpduMContainedRxInContainerPduRef shall NOT be defined.	
Template Description	
Defines properties for an IPdu that is part of the ContainerIPdu.	
M2 Parameter	
ContainerIPdu.containedIPduTriggeringProps	
Mapping Rule	**Mapping Type**
In the SysT the ContainerPdu references all PduTriggerings via ContainedIPduProps which can be put inside this Container. In the EcuC each Contained Pdu refers to the Containers it can be transported in. In case of IPDUM_ACCEPT_ALL reception strategy:a set of IpduMContainedRxPdu without an IpduMContainedRxInContainerPduRef is derived. An IpduMContainedRxPdu shall only be derived once in this set of IPDUM_ACCEPT_ALL reception Pdus. The identity of an IpduMContainedRx Pdu in the set of IPDUM_ACCEPT_ALL reception Pdus is defined by the IpduMContainedRxPdu ShortHeaderId and IpduMContainedRxPduLongHeaderId.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00173]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	**BSW Type**
IpduMContainedRxPduLongHeaderId	ECUC-INTEGER-PARAM-DEF
BSW Description	
LongHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.	
Template Description	
Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.	
M2 Parameter	
ContainedIPduProps.headerIdLongHeader	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00203]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	**BSW Type**
IpduMContainedRxPduShortHeaderId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ShortHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.	
Template Description	
Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.	
M2 Parameter	
ContainedIPduProps.headerIdShortHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00202]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedRxPdu
BSW Parameter	**BSW Type**
IpduMPduUpdateBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
This value specifies where the PDU's Update-Bit is stored in the Container PDU (bit location of PDU's Update-Bit in the Container PDU).	
Template Description	
The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.	
M2 Parameter	
ContainedIPduProps.updateIndicationBitPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IpduM_00207]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	**BSW Type**
IpduMContainedPduHeaderId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Header Id which is part of the ContainerPdu when this ContainedPdu is inside.	
Template Description	

▽

△

ContainedIPduProps.headerIdLongHeader: Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader. **ContainedIPduProps.headerIdShortHeader:** Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.
M2 Parameter
ContainedIPduProps.headerIdLongHeader, ContainedIPduProps.headerIdShortHeader

Mapping Rule	Mapping Type
If IpduMContainerHeaderSize = LONG the IPduMContainedPduHeaderId is taken from headerId LongHeader. If IpduMContainerHeaderSize = SHORT the IPduMContainedPduHeaderId is taken from headerIdShortHeader.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00172]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu

BSW Parameter	BSW Type
IpduMContainedPduOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
Static offset (in bytes) of the ContainedPdu.	
Template Description	
Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.	
M2 Parameter	
ContainedIPduProps.offset	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IpduM_00206]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu

BSW Parameter	BSW Type
IpduMContainedTxInContainerPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the container Pdu which this contained Pdu shall be collected in.	
Template Description	
Defines properties for an IPdu that is part of the ContainerIPdu.	
M2 Parameter	
ContainerIPdu.containedIPduTriggeringProps	

Mapping Rule	Mapping Type
In the SysT the ContainerPdu references all PduTriggerings (via ContainedIPduProps) which can be put inside this container. In the EcuC each Pdu refers to the containers it can be transported in.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00176]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	**BSW Type**
IpduMContainedTxPduPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.	
Template Description	
Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.	
M2 Parameter	
ContainedIPduProps.priority	
Mapping Rule	**Mapping Type**
Priority handling for a ContainerIPdu is enabled if at least one ContainedIPdu contains the attribute "priority" within its aggregated ContainerIPduProps, and there are different priorities configured within one ContainerIPdu (Reason: When all ContainedIPdus have the same priority, they cannot be prioritized).	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00210]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	**BSW Type**
IpduMContainedTxPduSendTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines a ContainedPdu specific sender timeout which adapts the ContainerPdu timer when this ContainedPdu is put inside the ContainerPdu: If the ContainerPdu timer is already running, it is reduced to this value, otherwise it is set to this value. Defined in seconds.	
Template Description	
Defines an IPdu specific sender timeout which adapts the ContainerIPdu transmission timer when this containedIPdu is put inside the ContainerIPdu: • f the ContainerIPdu transmission timer is already running, and the transmission timer value is greater than ContainedIPdu Props.timeout, then the transmission timer value is reduced to ContainedIPduProps.timeout. • If the ContainerIPdu transmission timer is already running, and the transmission timer value is less than or equal to ContainedIPduProps.timeout, then the transmission timer value is not changed. • If the ContainerIPdu transmission timer is not running, then the transmission timer value is set to ContainedIPdu Props.timeout. This attribute is ignored on receiver side.	
M2 Parameter	
ContainedIPduProps.timeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00181]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	**BSW Type**
IpduMContainedTxPduTrigger	ECUC-ENUMERATION-PARAM-DEF



△

BSW Description	
Defines whether this Pdu triggers the sending of the ContainerPdu.	
Template Description	
Defines whether this IPdu does trigger the sending of the ContainerIPdu. This attribute is ignored on receiver side.	
M2 Parameter	
ContainedIPduProps.trigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00182]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduTrigger	
BSW Parameter	**BSW Type**	
IPDUM_TRIGGER_ALWAYS	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This Pdu directly triggers the sending of the ContainerPdu.		
Template Description		
Pdu will trigger the transmission of the data.		
M2 Parameter		
PduCollectionTriggerEnum.always		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu/IpduMContainedTxPduTrigger	
BSW Parameter	**BSW Type**	
IPDUM_TRIGGER_NEVER	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
This Pdu does not triggers the sending of the ContainerPdu (other trigger criteria might still trigger sending of the Container Pdu).		
Template Description		
Pdu will be buffered and will not trigger the transmission of the data.		
M2 Parameter		
PduCollectionTriggerEnum.never		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainedTxPdu
BSW Parameter	**BSW Type**
IpduMPduUpdateBitPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
This value specifies where the PDU's Update-Bit is stored in the Container PDU (bit location of PDU's Update-Bit in the Container PDU).	
Template Description	
The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.	
M2 Parameter	
`ContainedIPduProps.updateIndicationBitPosition`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_IpduM_00207]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	**BSW Type**
IpduMContainerHeaderSize	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the layout of the header information (header id and length).	
Template Description	
Defines whether and which header type is used (header id and length).	
M2 Parameter	
`ContainerIPdu.headerType`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00183]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	**BSW Type**
IPDUM_HEADERTYPE_LONG	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit	
Template Description	
Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit	
M2 Parameter	

▽

△

ContainerIPduHeaderTypeEnum.longHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	**BSW Type**
IPDUM_HEADERTYPE_NONE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Static Container Layout	
Template Description	
No Header is used and the location of each containedPdu in the ContainerPdu is statically configured.	
M2 Parameter	
ContainerIPduHeaderTypeEnum.noHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerHeaderSize
BSW Parameter	**BSW Type**
IPDUM_HEADERTYPE_SHORT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit	
Template Description	
Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit.	
M2 Parameter	
ContainerIPduHeaderTypeEnum.shortHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu
BSW Parameter	**BSW Type**
IpduMContainerQueueSize	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.	
Template Description	
ContainerIPdu.minimumRxContainerQueueSize: This attribute defines the minimum queue size for received containers. **ContainerIPdu.minimumTxContainerQueueSize:** This attribute defines the minimum queue size for transmitted containers.	
M2 Parameter	
ContainerIPdu.minimumRxContainerQueueSize, ContainerIPdu.minimumTxContainerQueueSize	
Mapping Rule	**Mapping Type**
The value of this parameter can not be derived from the System Extract but the System Extract may define a minimum queue size. If this parameter is used in the context of the IpduMContainer TxPdu then the minimumTxContainerQueueSize attribute needs to be respected. If this parameter is used in the context of the IpduMContainerRxPdu then the minimumRxContainerQueueSize attribute needs to be respected.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00185]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu	
BSW Parameter		**BSW Type**
IpduMContainerRxAcceptContainedPdu		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Defines for the received IpduMContainerRxPdu whether the list of referencing IpduMContainedRxPdus (via the reference IpduMContainedPduContainerRefRx) is a closed set.		
Template Description		
Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.		
M2 Parameter		
ContainerIPdu.rxAcceptContainedIPdu		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IpduM_00186]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerRxAcceptContainedPdu	
BSW Parameter		**BSW Type**
IPDUM_ACCEPT_ALL		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
The IpduMContainedRxPdus which are referencing this IpduMContainerRxPdu are expected inside this IpduMContainerRx Pdu, but there may also occur other Pdus inside this IpduMContainerRxPdu as well. This also supports the case where no IpduMContainedRxPdu references the IpduMContainerRxPdu.		
Template Description		
No fixed set of containedIPdus is defined for reception, any known containedIPdu (based on headerId) shall be expected within this ContainerIPdu.		
M2 Parameter		
RxAcceptContainedIPduEnum.acceptAll		

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerRxPdu/IpduMContainerRxAcceptContainedPdu
BSW Parameter	**BSW Type**
IPDUM_ACCEPT_CONFIGURED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Only the IpduMContainedRxPdus which are referencing this IpduMContainerRxPdu are expected inside this IpduMContainer RxPdu.	
Template Description	
A fixed set of containedIPdus is defined for reception. Only these assigned containedIPdus (based on headerId) are expected in this ContainerIPdu. If a not assigned containedIPdu is received within this ContainerIPdu this containedIPdu is discarded.	
M2 Parameter	
RxAcceptContainedIPduEnum.acceptConfigured	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainedTxPduCollectionSemantics	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines whether this IpduMContainerTxPdu shall be collector using a last-is-best or queued semantics.	
Template Description	
Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.	
M2 Parameter	
ContainedIPduProps.collectionSemantics	
Mapping Rule	**Mapping Type**
If the ContainerIPdu references the PduTriggering via ContainerIPdu.containedIPduTriggering Props then the value of IpduMContainedTxPduCollectionSemantics shall be derived from collectionSemantics of one of the referenced ContainedIPduProps. It is possible to derive this parameter from ContainedIPduProps because the value of collectionSemantics is the same in all ContainedIPduProps that are aggregated by the ContainerIPdu in the role containedIPduTriggering Props as described by constr_3517.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00198]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainedTxPduCollectionSemantics
BSW Parameter	**BSW Type**
IPDUM_COLLECT_LAST_IS_BEST	ECUC-ENUMERATION-LITERAL-DEF

▽

△

BSW Description	
Collected IpduMContainedTxPdu data by this IpduMContainerTxPdu will be fetched via TriggerTransmit just before the transmission executes.	
Template Description	
The ContainedIPdu data will be fetched via TriggerTransmit just before the transmission executes.	
M2 Parameter	
ContainedIPduCollectionSemanticsEnum.lastIsBest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainedTxPduCollectionSemantics
BSW Parameter	**BSW Type**
IPDUM_COLLECT_QUEUED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Relevant IpduMContainedTxPdu data will instantly be stored to this IpduMContainerTxPdu in the context of the Transmit API.	
Template Description	
The ContainedIPdu data will instantly be stored to the ContainerIPdu in the context of the Transmit API.	
M2 Parameter	
ContainedIPduCollectionSemanticsEnum.queued	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainerHeaderSize	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the layout of the header information (header id and length).	
Template Description	
Defines whether and which header type is used (header id and length).	
M2 Parameter	
ContainerIPdu.headerType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00183]

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize	
BSW Parameter		**BSW Type**
IPDUM_HEADERTYPE_LONG		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit		
Template Description		
Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit		
M2 Parameter		
ContainerIPduHeaderTypeEnum.longHeader		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize	
BSW Parameter		**BSW Type**
IPDUM_HEADERTYPE_NONE		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Static Container Layout		
Template Description		
No Header is used and the location of each containedPdu in the ContainerPdu is statically configured.		
M2 Parameter		
ContainerIPduHeaderTypeEnum.noHeader		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu/IpduMContainerHeaderSize	
BSW Parameter		**BSW Type**
IPDUM_HEADERTYPE_SHORT		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit		
Template Description		

▽

△

Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit.	
M2 Parameter	
ContainerIPduHeaderTypeEnum.shortHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainerQueueSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.	
Template Description	
ContainerIPdu.minimumRxContainerQueueSize: This attribute defines the minimum queue size for received containers. **ContainerIPdu.minimumTxContainerQueueSize:** This attribute defines the minimum queue size for transmitted containers.	
M2 Parameter	
ContainerIPdu.minimumRxContainerQueueSize, ContainerIPdu.minimumTxContainerQueueSize	
Mapping Rule	**Mapping Type**
The value of this parameter can not be derived from the System Extract but the System Extract may define a minimum queue size. If this parameter is used in the context of the IpduMContainer TxPdu then the minimumTxContainerQueueSize attribute needs to be respected. If this parameter is used in the context of the IpduMContainerRxPdu then the minimumRxContainerQueueSize attribute needs to be respected.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00185]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainerTxFirstContainedPduTrigger	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines if the transmission of this IpduMContainerTxPdu shall be requested right after the first IpduMContainedTxPdu was put into it.	
Template Description	
Defines if the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into it. This attribute shall be ignored on receiver side.	
M2 Parameter	
ContainerIPdu.containerTrigger	
Mapping Rule	**Mapping Type**
TRUE if ContainerIPdu.containerTrigger = firstContainedTrigger, else FALSE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00199]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainerTxSendTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
When this timeout expires the ContainerPdu is triggered for sending. The respective timer is started when the first Pdu is put into the ContainerPdu. Defined in seconds.	
Template Description	
When this timeout expires the ContainerIPdu is sent out. The respective timer is started when the first Ipdu is put into the ContainerIPdu. This attribute is ignored on receiver side.	
M2 Parameter	
ContainerIPdu.containerTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00194]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMContainerTxSizeThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the size threshold in bytes which, when exceeded, triggers the sending of the ContainerPdu although the maxium Pdu size (PduLength parameter of Pdu object) has not been reached yet.	
Template Description	
Defines the size threshold which, when exceeded, triggers the sending of the ContainerIPdu although the maximum Pdu size has not been reached yet. Unit: byte.	
M2 Parameter	
ContainerIPdu.thresholdSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00195]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMContainerTxPdu
BSW Parameter	**BSW Type**
IpduMUnusedAreasDefault	ECUC-INTEGER-PARAM-DEF
BSW Description	
IpduM fills not updated areas of the Container PDU with this byte-pattern.	
Template Description	
IPduM fills not updated areas of the ContainerPdu with this byte-pattern.	
M2 Parameter	
ContainerIPdu.unusedBitPattern	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_IpduM_00208]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig
BSW Parameter	**BSW Type**
IpduMRxPathway	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters received I-PDUs by the IpduM module.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	
Mapping Rule	**Mapping Type**
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00071]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway
BSW Parameter	**BSW Type**
IpduMRxIndication	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration for incoming RxIndication calls.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	
Mapping Rule	**Mapping Type**
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00047]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	**BSW Type**

▽

△

IpduMByteOrder	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.	
Template Description	
MultiplexedIPdu.selectorFieldByteOrder: This attribute defines the order of the bytes of the selectorField and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1. **SegmentPosition.segmentByteOrder:** This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.	
M2 Parameter	
`MultiplexedIPdu.selectorFieldByteOrder`, `SegmentPosition.segmentByteOrder`	
Mapping Rule	**Mapping Type**
A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00162]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	**BSW Type**
IpduMRxDynamicPart	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration for the dynamic part of incoming RxIndication calls. When an incoming received I-PDU's selector field matches the IpduMRxSelectorValue, the new outgoing I-PDU for the dynamic part is constructed as defined by the segments (defined in the IpduMDynamicSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingDynamicPduRef. In case no dynamic part shall be extracted from this received I-PDU this container does not exist. This use-case can occur in case a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu.	
Template Description	
One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.	
M2 Parameter	
`DynamicPartAlternative`	
Mapping Rule	**Mapping Type**
Create container for each DynamicPartAlternative of the MultiplexedIPdu that is actually received by this ECU Instance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00048]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart
BSW Parameter	**BSW Type**

▽

△

IpduMOutgoingDynamicPduRef	ECUC-REFERENCE-DEF
BSW Description	
When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent PDU representation in the ECU Configuration Description exchange file.	
Template Description	
M2 Parameter	
DynamicPartAlternative.iPdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00112]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart
BSW Parameter	**BSW Type**
IpduMRxSelectorValue	ECUC-INTEGER-PARAM-DEF
BSW Description	
This is the selector value that this container refers to.	
Template Description	
The selector field is part of a multiplexed IPdu. It consists of contiguous bits. The value of the selector field selects the layout of the multiplexed part of the IPdu.	
M2 Parameter	
DynamicPartAlternative.selectorFieldCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00113]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	**BSW Type**
IpduMRxDynamicSegment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The dynamic part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxDynamic Part container and will be copied from the same location in the multiplexed incoming I-Pdu. The segment layout for all dynamic Parts is always identical.	
Template Description	
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.	
M2 Parameter	
SegmentPosition	
Mapping Rule	**Mapping Type**

▽

△

Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00170]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicSegment
BSW Parameter	**BSW Type**
IpduMSegmentLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	
SegmentPosition.segmentLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00114]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicSegment
BSW Parameter	**BSW Type**
IpduMSegmentPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	
Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.	
M2 Parameter	
SegmentPosition.segmentPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00159]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	**BSW Type**
IpduMRxSelectorField	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This contains the location and the length of the selector field.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	
Mapping Rule	**Mapping Type**
If this IpduMRxIndication defines at least one IpduMRxDynamicPart to be received by this ECU Instance, then derive the IpduMRxSelectorField as mandatory.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00222]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxSelectorField
BSW Parameter	**BSW Type**
IpduMSelectorFieldLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the selector field in bits.	
Template Description	
The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
MultiplexedIPdu.selectorFieldLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00160]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxSelectorField
BSW Parameter	**BSW Type**
IpduMSelectorFieldPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Selector field bit position in the multiplexed Pdu. Range: 0..63 for CAN/ LIN I-PDUs, 0..511 for CAN FD I-PDUs, 0..2031 for FlexRay I-PDUs.	
Template Description	

▽

△

This parameter is necessary to describe the position of the selector field within the IPdu. Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorField ByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
M2 Parameter
MultiplexedIPdu.selectorFieldStartPosition

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00161]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication

BSW Parameter	**BSW Type**
IpduMRxStaticPart	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration for the static part of incoming RxIndication calls. On reception, the new outgoing I-PDU for the static part is constructed as defined by the segments (defined in the IpduMStaticSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingStaticPduRef.	
Template Description	
Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.	
M2 Parameter	
StaticPart	

Mapping Rule	**Mapping Type**
Create container if StaticPart exists in the MultiplexedIPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00049]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart

BSW Parameter	**BSW Type**
IpduMOutgoingStaticPduRef	ECUC-REFERENCE-DEF
BSW Description	
When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent Pdu representation in the ECU Configuration Description exchange file.	
Template Description	
M2 Parameter	
StaticPart.iPdu	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00115]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication
BSW Parameter	**BSW Type**
IpduMRxStaticSegment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxStatic Part container and will be copied from the same location in the multiplexed incoming I-Pdu.	
Template Description	
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.	
M2 Parameter	
SegmentPosition	
Mapping Rule	**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00169]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment
BSW Parameter	**BSW Type**
IpduMSegmentLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	
SegmentPosition.segmentLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00114]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment
BSW Parameter	**BSW Type**
IpduMSegmentPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	

▽

△

Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.	
M2 Parameter	
SegmentPosition.segmentPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00159]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig
BSW Parameter	**BSW Type**
IpduMTxPathway	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration parameters transmitted I-PDUs by the IpduM module.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	
Mapping Rule	**Mapping Type**
Create container for each transmitted multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "Out" Pdu Port.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00070]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway
BSW Parameter	**BSW Type**
IpduMTxRequest	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is used to specify the configuration for Transmit requests. There will be one instance of this container for each I-PDU that can be requested for transmission (the outgoing I-PDUs) by the IpduM.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	

▽

△

Mapping Rule	Mapping Type
Create container for each transmitted multiplexed Ipdu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00052]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMByteOrder	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.	
Template Description	
MultiplexedIPdu.selectorFieldByteOrder: This attribute defines the order of the bytes of the selectorField and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1. **SegmentPosition.segmentByteOrder:** This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.	
M2 Parameter	
`MultiplexedIPdu.selectorFieldByteOrder`, `SegmentPosition.segmentByteOrder`	
Mapping Rule	**Mapping Type**
A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00162]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMIPduUnusedAreasDefault	ECUC-INTEGER-PARAM-DEF
BSW Description	
IpduM module fills not used areas of an I-PDU with this bit-pattern If this attribute is omitted the IpduM module does not fill the I-PDU.	
Template Description	
AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPdu with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
`MultiplexedIPdu.unusedBitPattern`	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00121]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMInitialDynamicPart	ECUC-REFERENCE-DEF
BSW Description	
Reference to the dynamic part that shall be used to initialize this multiplexed TX-I-PDU.	
Template Description	
Dynamic part that shall be used to initialize this multiplexed IPdu. Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.	
M2 Parameter	
`DynamicPartAlternative.initialDynamicPart`	
Mapping Rule	**Mapping Type**
If the attribute initialDynamicPart is set to true then create this reference.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00157]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxDynamicPart	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration parameters for an instance of a TxRequest call into the IpduM. When a Tx Request with the IpduMTxDynamic HandleId is received by the IpduM, all segments (defined in the IpduMDynamicSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used by the dynamic part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the dynamic part.	
Template Description	
One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.	
M2 Parameter	
`DynamicPartAlternative`	
Mapping Rule	**Mapping Type**
Create container for each DynamicPartAlternative of the MultiplexedIPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00056]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxDynamicSegment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

The dynamic part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied out of the source I-Pdu that is referenced in the IpduMTxDynamic Part container and will be copied to the same location in the multiplexed outgoing I-Pdu. The segment layout for all dynamic Parts is always identical.
Template Description
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.
M2 Parameter
SegmentPosition

Mapping Rule	**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00168]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicSegment

BSW Parameter	**BSW Type**
IpduMSegmentLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	
SegmentPosition.segmentLength	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00114]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicSegment

BSW Parameter	**BSW Type**
IpduMSegmentPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	
Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.	
M2 Parameter	
SegmentPosition.segmentPosition	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxSelectorField	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This contains the location and the length of the selector field.	
Template Description	
A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer. A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.	
M2 Parameter	
MultiplexedIPdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00221]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxSelectorField
BSW Parameter	**BSW Type**
IpduMSelectorFieldLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the selector field in bits.	
Template Description	
The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
MultiplexedIPdu.selectorFieldLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00160]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxSelectorField
BSW Parameter	**BSW Type**

▽

△

IpduMSelectorFieldPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Selector field bit position in the multiplexed Pdu. Range: 0..63 for CAN/ LIN I-PDUs, 0..511 for CAN FD I-PDUs, 0..2031 for FlexRay I-PDUs.	
Template Description	
This parameter is necessary to describe the position of the selector field within the IPdu. Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorField ByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
MultiplexedIPdu.selectorFieldStartPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00161]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxStaticPart	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Configuration parameters for an instance of a Tx_Request call into the IpduM. When a Tx Request with the IpduMTxStatic HandleId is received by the IpduM, all segments (defined in the IpduMStaticSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used for the static part of a Tx Request configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the static part if it exists.	
Template Description	
Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.	
M2 Parameter	
StaticPart	
Mapping Rule	**Mapping Type**
Create container if StaticPart exists in the MultiplexedIPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00082]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxStaticSegment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

The static part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each segment in the source I-Pdu that is referenced in the IpduMTxStaticPart container will be copied to the same location in the multiplexed outgoing I-Pdu.	
Template Description	
The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.	
M2 Parameter	
SegmentPosition	
Mapping Rule	**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00171]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSegment
BSW Parameter	**BSW Type**
IpduMSegmentLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Length of the segment in bits.	
Template Description	
Data Length of the segment in bits.	
M2 Parameter	
SegmentPosition.segmentLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00114]

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSegment
BSW Parameter	**BSW Type**
IpduMSegmentPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
Segments bit position in the multiplexed Pdu.	
Template Description	
Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.	
M2 Parameter	
SegmentPosition.segmentPosition	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00159]

BSW Module	BSW Context
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxTriggerMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Selects whether to send the multiplexed I-PDU immediately or at some later date.	
Template Description	
IPduM can be configured to send a transmission request for the new multiplexed IPdu to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.	
M2 Parameter	
MultiplexedIPdu.triggerMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00125]

BSW Module	BSW Context
IpduM	IpduM/IpduMGeneral
BSW Parameter	**BSW Type**
IpduMHeaderByteOrder	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the ByteOrder of the headers inside a Container I-PDU.	
Template Description	
Defines the byteOrder of the header in ContainerIPdus.	
M2 Parameter	
System.containerIPduHeaderByteOrder	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IpduM_00197]

BSW Module	BSW Context
IpduM	IpduM/IpduMGeneral/IpduMHeaderByteOrder
BSW Parameter	**BSW Type**
IPDUM_BIG_ENDIAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Headers inside a Container I-PDU shall be ordered big endian.	
Template Description	
Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format)	
M2 Parameter	
ByteOrderEnum.mostSignificantByteFirst	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IpduM	IpduM/IpduMGeneral/IpduMHeaderByteOrder
BSW Parameter	**BSW Type**
IPDUM_LITTLE_ENDIAN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Headers inside a Container I-PDU shall be ordered little endian.	
Template Description	
Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format)	
M2 Parameter	
ByteOrderEnum.mostSignificantByteLast	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

C.37 J1939Dcm

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel
BSW Parameter	**BSW Type**
J1939DcmBusType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Identifies the communication port	
Template Description	
Identification of a network to which the broadcast of messages is started by the DM13 command message.	
M2 Parameter	
J1939Cluster.dcmBusType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00039]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_ISO9141	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the ISO 9141 communications port.	
Template Description	
ISO 9141 network	
M2 Parameter	
J1939BusTypeEnum.iso_9141	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1587	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1587 communication port.	
Template Description	
SAE J1587 network	
M2 Parameter	
J1939BusTypeEnum.j1587	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1850	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1850 communication port.	
Template Description	
SAE J1850 network	
M2 Parameter	
J1939BusTypeEnum.j1850	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1922	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1922 communication port.	
Template Description	
SAE J1922 network	
M2 Parameter	
J1939BusTypeEnum.j1922	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1939_NETWORK_1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1939 Network #1, Primary Vehicle Network communication port.	
Template Description	
SAE J1939 network #1, primary vehicle network	
M2 Parameter	
J1939BusTypeEnum.j1939_1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1939_NETWORK_2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1939 Network #2 communication port.	
Template Description	
SAE J1939 network #2	
M2 Parameter	
J1939BusTypeEnum.j1939_2	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1939_NETWORK_3	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1939 Network #3 communication port.	
Template Description	
SAE J1939 network #3	
M2 Parameter	
J1939BusTypeEnum.j1939_3	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_J1939_NETWORK_4	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the J1939 Network #4 communication port.	
Template Description	
SAE J1939 network #4	
M2 Parameter	
J1939BusTypeEnum.j1939_4	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_OTHER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the "Other, Manufacture Specified Port" communication port.	
Template Description	
Manufacture Specified Port	
M2 Parameter	
J1939BusTypeEnum.other	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_PROPRIETARY_NETWORK_1	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the Proprietary Network #1 communication port.	
Template Description	
Proprietary network #1	
M2 Parameter	
J1939BusTypeEnum.proprietary_1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel/J1939DcmBusType
BSW Parameter	**BSW Type**
J1939DCM_PROPRIETARY_NETWORK_2	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Identifies the Proprietary Network #2 communication port.	
Template Description	
Proprietary network #2	
M2 Parameter	
J1939BusTypeEnum.proprietary_2	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode
BSW Parameter	**BSW Type**
J1939DcmDiagnosticMessageSupport	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains parameters to configure the diagnostic message support	
Template Description	
Represents the IPdus handled by J1939Dcm.	
M2 Parameter	
J1939DcmIPdu	
Mapping Rule	**Mapping Type**
The container shall be created for every J1939DcmIPdu that is transmitted oder received by the regarded Ecu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00014]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport
BSW Parameter	**BSW Type**
J1939DcmDmxSupport	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter is used to identify the actual DMx message.	
Template Description	
This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.	
M2 Parameter	
J1939DcmIPdu.diagnosticMessageType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00042]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport
BSW Parameter	**BSW Type**
J1939DcmRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains parameters to configure the J1939DcmRxPdu. This PDU consumes meta data items of type CAN_ID_32 for PDUs received from CanIf, and of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8 for PDUs received from J1939Tp.	
Template Description	
Represents the IPdus handled by J1939Dcm.	
M2 Parameter	
J1939DcmIPdu	
Mapping Rule	**Mapping Type**
The direction of the J1939DcmIPdu shall be derived from the PduTriggering and the references to IPduPorts.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00046]

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode/J1939DcmDiagnosticMessageSupport
BSW Parameter	**BSW Type**
J1939DcmTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains parameters to configure the J1939DcmTxPdu. This PDU produces meta data items of type CAN_ID_32 for PDUs transmitted via CanIf, and of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8 for PDUs transmitted via J1939Tp.	
Template Description	
Represents the IPdus handled by J1939Dcm.	
M2 Parameter	

▽

△

J1939DcmIPdu	
Mapping Rule	**Mapping Type**
The direction of the J1939DcmIPdu shall be derived from the PduTriggering and the references to IPduPorts.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Dcm_00045]

C.38 J1939Nm

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet
BSW Parameter	**BSW Type**
J1939NmChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Physical CAN channel handled by J1939Nm.	
Template Description	
J1939 specific NmCluster attributes	
M2 Parameter	
J1939NmCluster	
Mapping Rule	**Mapping Type**
Create Container for each existing J1939NmCluster.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00005]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel
BSW Parameter	**BSW Type**
J1939NmChannelUsesAddressArbitration	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes. True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not react to other address claims.	
Template Description	
Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).	
M2 Parameter	
J1939NmNode.addressConfigurationCapability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00035]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel

BSW Parameter	BSW Type
J1939NmChannelUsesDynamicAddressing	ECUC-BOOLEAN-PARAM-DEF

BSW Description

Defines whether fully dynamic address resolution according to SAE J1939-81 shall be supported on this channel.

- True: The dynamically allocated addresses on the bus are matched at runtime to the configured addresses (see J1939Nm Node.J1939NmNodePreferredAddress and J1939NmExternalNode.J1939NmExternalNodePreferredAddress). J1939Nm Nodes with J1939NmAddressConfigurationCapability set to J1939NM_AAC will change their addresses dynamically in case of an address conflict.
- False: The addresses on the bus resemble the configured addresses. J1939NmAddressConfigurationCapability shall not be set to J1939NM_AAC.

Template Description

Defines whether fully dynamic address resolution according to SAE J1939-81 shall be supported on this J1939NmCluster.

- True: The dynamically allocated addresses on the bus are matched at runtime to the configured addresses.
- False: The addresses on the bus resemble the configured addresses.

M2 Parameter

`J1939NmCluster.usesDynamicAddressing`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00054]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel

BSW Parameter	BSW Type
J1939NmRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Contains the configuration of the PDU used to receive the AddressClaimed PG. This PDU consumes a meta data item of type CAN_ID_32.

Template Description

M2 Parameter

`NmNode.rxNmPdu`

Mapping Rule	Mapping Type
Shall be derived from the NmPdu that is referenced by the NmNode.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00010]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel

BSW Parameter	BSW Type
J1939NmTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Contains the configuration of the PDU used to transmit the AddressClaimed PG. This PDU produces a meta data item of type CAN_ID_32.

▽

△

Template Description	
M2 Parameter	
NmNode.txNmPdu	
Mapping Rule	**Mapping Type**
Shall be derived from the NmPdu that is referenced by the NmNode.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00009]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet	
BSW Parameter		**BSW Type**
J1939NmExternalNode		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Logical node implemented in another ECU. Configures potential communication partners. If this container is connected to more than one channel, the external ECU is linked to the local ECU by each of these channels.		
Template Description		
J1939 specific NM Node attributes.		
M2 Parameter		
J1939NmNode		
Mapping Rule		**Mapping Type**
J1939NmNode shall be derived from existing J1939NmNodes that are available in the ExuExtract. Please note that J1939NmNodes that have the same shortName and nmNodeId that are located on different NmClusters shall be combined to one J1939NmNode.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00039]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter		**BSW Type**
J1939NmExternalNodeNameArbitraryAddressCapable		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Arbitrary Address Capable field of the NAME of this external node.		
Template Description		
Arbitrary Address Capable field of the NAME of this node.		
M2 Parameter		
J1939NodeName.arbitraryAddressCapable		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00041]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameECUInstance	ECUC-INTEGER-PARAM-DEF	
BSW Description		
ECU Instance field of the NAME of this external node.		
Template Description		
ECU Instance field of the NAME of this node.		
M2 Parameter		
J1939NodeName.ecuInstance		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00042]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameFunction	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Function field of the NAME of this external node.		
Template Description		
Function field of the NAME of this node.		
M2 Parameter		
J1939NodeName.function		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00043]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameFunctionInstance	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Function Instance field of the NAME of this external node.		
Template Description		
Function Instance field of the NAME of this node.		
M2 Parameter		
J1939NodeName.functionInstance		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00044]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameIdentityNumber	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Identity Number field of the NAME of this external node.		
Template Description		
Identity Number field of the NAME of this node.		
M2 Parameter		
J1939NodeName.identitiyNumber		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00045]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameIndustryGroup	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Industry Group field of the NAME of this external node.		
Template Description		
Industry Group field of the NAME of this node.		
M2 Parameter		
J1939NodeName.industryGroup		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00046]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode	
BSW Parameter	**BSW Type**	
J1939NmExternalNodeNameManufacturerCode	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Manufacturer Code field of the NAME of this external node.		
Template Description		
Manufacturer Code field of the NAME of this node.		
M2 Parameter		
J1939NodeName.manufacturerCode		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00047]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	**BSW Type**
J1939NmExternalNodeNameVehicleSystem	ECUC-INTEGER-PARAM-DEF
BSW Description	
Vehicle System field of the NAME of this external node.	
Template Description	
Vehicle System field of the NAME of this node.	
M2 Parameter	
J1939NodeName.vehicleSystem	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00048]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	**BSW Type**
J1939NmExternalNodeNameVehicleSystemInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
Vehicle System Instance field of the NAME of this external node.	
Template Description	
Vehicle System Instance field of the NAME of this node.	
M2 Parameter	
J1939NodeName.vehicleSystemInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00050]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmExternalNode
BSW Parameter	**BSW Type**
J1939NmExternalNodePreferredAddress	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source address of this external node.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
NmNode.nmNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00049]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet	
BSW Parameter		**BSW Type**
J1939NmNode		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Logical node representing one function handled by J1939Nm.		
Template Description		
J1939 specific NM Node attributes.		
M2 Parameter		
J1939NmNode		
Mapping Rule		**Mapping Type**
J1939NmNode shall be derived from existing J1939NmNodes that are available in the ExuExtract. Please note that J1939NmNodes that have the same shortName and nmNodeId that are located on different NmClusters shall be combined to one J1939NmNode.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00015]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmAddressConfigurationCapability		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).		
Template Description		
Defines the Address Configuration Capability of the J1939NmNode (corresponding to an SAE J1939 Controller Application, CA).		
M2 Parameter		
J1939NmNode.addressConfigurationCapability		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_00055]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeChannelRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the channels this node has access to.		
Template Description		
M2 Parameter		
NmCluster.nmNode		
Mapping Rule		**Mapping Type**
This reference shall be derived from NmClusters that aggregate the nmNode in the Ecu Extract.		full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_J1939Nm_00029]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameArbitraryAddressCapable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Arbitrary Address Capable field of the NAME of this node.	
Template Description	
Arbitrary Address Capable field of the NAME of this node.	
M2 Parameter	
J1939NodeName.arbitraryAddressCapable	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00018]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameECUInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
ECU Instance field of the NAME of this node.	
Template Description	
ECU Instance field of the NAME of this node.	
M2 Parameter	
J1939NodeName.ecuInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00024]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameFunction	ECUC-INTEGER-PARAM-DEF
BSW Description	
Function field of the NAME of this node.	
Template Description	
Function field of the NAME of this node.	
M2 Parameter	

▽

△

J1939NodeName.function	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00022]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameFunctionInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
Function Instance field of the NAME of this node.	
Template Description	
Function Instance field of the NAME of this node.	
M2 Parameter	
J1939NodeName.functionInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00023]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameIdentityNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identity Number field of the NAME of this node.	
Template Description	
Identity Number field of the NAME of this node.	
M2 Parameter	
J1939NodeName.identitiyNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00026]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameIndustryGroup	ECUC-INTEGER-PARAM-DEF
BSW Description	
Industry Group field of the NAME of this node.	

▽

△

Template Description	
Industry Group field of the NAME of this node.	
M2 Parameter	
J1939NodeName.industryGroup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00019]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameManufacturerCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Manufacturer Code field of the NAME of this node.	
Template Description	
Manufacturer Code field of the NAME of this node.	
M2 Parameter	
J1939NodeName.manufacturerCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00025]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodeNameVehicleSystem	ECUC-INTEGER-PARAM-DEF
BSW Description	
Vehicle System field of the NAME of this node.	
Template Description	
Vehicle System field of the NAME of this node.	
M2 Parameter	
J1939NodeName.vehicleSystem	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00021]

BSW Module	BSW Context
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**

▽

△

J1939NmNodeNameVehicleSystemInstance	ECUC-INTEGER-PARAM-DEF
BSW Description	
Vehicle System Instance field of the NAME of this node.	
Template Description	
Vehicle System Instance field of the NAME of this node.	
M2 Parameter	
J1939NodeName.vehicleSystemInstance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00020]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode
BSW Parameter	**BSW Type**
J1939NmNodePreferredAddress	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source address of this node used for address claiming.	
Template Description	
Node identifier of local NmNode. Shall be unique in the NmCluster.	
M2 Parameter	
NmNode.nmNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00016]

BSW Module	**BSW Context**
J1939Nm	J1939Nm/J1939NmConfigSet
BSW Parameter	**BSW Type**
J1939NmSharedAddressSpace	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Set of J1939NmChannels that share a common address space. Address claims will be routed between these channels.	
Template Description	
This meta-class represents the ability to identify several J1939Clusters that share a common address space for the routing of messages	
M2 Parameter	
J1939SharedAddressCluster	
Mapping Rule	**Mapping Type**
Container shall be created for each exisiting J1939SharedAddressCluster	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Nm_00037]

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmSharedAddressSpace	
BSW Parameter		**BSW Type**
J1939NmSharedChannelRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to a channel that belongs to the shared address space.		
Template Description		
This identifies the J1939Clusters that share a common address space		
M2 Parameter		
J1939SharedAddressCluster.participatingJ1939Cluster		
Mapping Rule		**Mapping Type**
Reference shall be created for each J1939 cluster that is referenced by J1939SharedAddress Cluster in the role participating1939Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Nm_ 00038]

C.39 J1939Rm

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet	
BSW Parameter		**BSW Type**
J1939RmChannel		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Contains the parameters for a CAN channel supported by the J1939 Request Manager.		
Template Description		
J1939 specific NmCluster attributes		
M2 Parameter		
J1939NmCluster		
Mapping Rule		**Mapping Type**
Container shall be created for each J1939NmCluster that is available in the EcuExtract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_J1939Rm_ 00009]

BSW Module	BSW Context	
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmChannel	
BSW Parameter		**BSW Type**
J1939RmRqst2RxPdu		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Contains the configuration of the I-PDU used to receive the Request2 PG. This PDU consumes a meta data item of type CAN_ID_32.		
Template Description		
Enables support for the Request2 PGN (RQST2).		

▽

△

M2 Parameter	
J1939Cluster.request2Support	
Mapping Rule	**Mapping Type**
Create container if J1939Cluster.request2Support is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00075]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmChannel
BSW Parameter	**BSW Type**
J1939RmRqst2TxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the configuration of the I-PDU used to transmit the Request2 PG. This PDU produces a meta data item of type CAN_ID_32.	
Template Description	
Enables support for the Request2 PGN (RQST2).	
M2 Parameter	
J1939Cluster.request2Support	
Mapping Rule	**Mapping Type**
Create container if J1939Cluster.request2Support is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00076]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet
BSW Parameter	**BSW Type**
J1939RmNode	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the parameters for the support of a logical J1939 node (identified by an ECU address).	
Template Description	
J1939 specific NM Node attributes.	
M2 Parameter	
J1939NmNode	
Mapping Rule	**Mapping Type**
J1939RmNode shall be derived from existing J1939NmNodes that are available in the ExuExtract. Please note that J1939NmNodes that have the same shortName and nmNodeId that are located on different NmClusters shall be combined to one J1939RmNode.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00049]

BSW Module	BSW Context
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode
BSW Parameter	**BSW Type**

▽

△

J1939RmNodeChannelRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the channels this node has access to.	
Template Description	
M2 Parameter	
NmCluster.nmNode	
Mapping Rule	**Mapping Type**
This reference shall be derived from NmClusters that aggregate the nmNode in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00052]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode
BSW Parameter	**BSW Type**
J1939RmUser	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
Contains the configuration of a module that uses the request and acknowledgement interfaces of J1939Rm.	
Template Description	
DcmIPdu: Represents the IPdus handled by Dcm. **ISignalIPdu:** Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed. **NmPdu:** Network Management Pdu **CanFrameTriggering.identifier:** This attribute is used to define the identifier this frame shall use on the CAN network. **CanFrameTriggering.j1939requestable:** Frame can be triggered by the J1939 request message.	
M2 Parameter	
DcmIPdu, ISignalIPdu, NmPdu, CanFrameTriggering.identifier, CanFrameTriggering.j1939requestable	
Mapping Rule	**Mapping Type**
+ J1939NM user exists always, UserPGN has 0x0ee00 as solitary value + J1939DCM user exists if transmitted J1939DcmIPdus exist which are requestable + COM user exists if transmitted ISignal IPdus exist which are requestable + CDD user exists if transmitted UserDefinedPdus or User DefinedIPdus exist which are requestable + RTE users cannot be derived	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00010]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmCddUser
BSW Parameter	**BSW Type**
J1939RmUserRequestPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

PGN supported to be requested from this module. The PGNs supported by different modules should usually be disjunctive.	
Template Description	
Pdu: Collection of all Pdus that can be routed through a bus interface. **CanFrameTriggering.j1939requestable:** Frame can be triggered by the J1939 request message. **CanFrameTriggering.identifier:** This attribute is used to define the identifier this frame shall use on the CAN network. **J1939TpPg.requestable:** Parameter Group can be triggered by the J1939 request message. **J1939TpPg.pgn:** Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
Pdu, CanFrameTriggering.j1939requestable, CanFrameTriggering.identifier, J1939TpPg.requestable, J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
This parameter can be derived from the Pdu and a combination of CanFrame Triggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00026]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmComUser/J1939RmComIPdu
BSW Parameter	**BSW Type**
J1939RmComIPduPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	
PGN of the COM I-PDU.	
Template Description	
ISignalIPdu: Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed. **CanFrameTriggering.j1939requestable:** Frame can be triggered by the J1939 request message. **CanFrameTriggering.identifier:** This attribute is used to define the identifier this frame shall use on the CAN network. **J1939TpPg.requestable:** Parameter Group can be triggered by the J1939 request message. **J1939TpPg.pgn:** Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
ISignalIPdu, CanFrameTriggering.j1939requestable, CanFrameTriggering.identifier, J1939TpPg.requestable, J1939TpPg.pgn	
Mapping Rule	**Mapping Type**

▽

△

This parameter can be derived fromISignalIPdu and a combination of CanFrame Triggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00033]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmDcmUser
BSW Parameter	**BSW Type**
J1939RmUserRequestPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	
PGN of DMx PG supported by J1939Dcm.	
Template Description	
Pdu: Collection of all Pdus that can be routed through a bus interface. **CanFrameTriggering.j1939requestable:** Frame can be triggered by the J1939 request message. **CanFrameTriggering.identifier:** This attribute is used to define the identifier this frame shall use on the CAN network. **J1939TpPg.requestable:** Parameter Group can be triggered by the J1939 request message. **J1939TpPg.pgn:** Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
Pdu, CanFrameTriggering.j1939requestable, CanFrameTriggering.identifier, J1939TpPg.requestable, J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
This parameter can be derived from the Pdu and a combination of CanFrame Triggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_00070]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmConfigSet/J1939RmNode/J1939RmUser/J1939RmRteUser
BSW Parameter	**BSW Type**
J1939RmUserRequestPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	
PGN supported to be requested from this module. The PGNs supported by different modules should usually be disjunctive.	
Template Description	

▽

△

Pdu: Collection of all Pdus that can be routed through a bus interface. **CanFrameTriggering.j1939requestable:** Frame can be triggered by the J1939 request message. **CanFrameTriggering.identifier:** This attribute is used to define the identifier this frame shall use on the CAN network. **J1939TpPg.requestable:** Parameter Group can be triggered by the J1939 request message. **J1939TpPg.pgn:** Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.

M2 Parameter	
Pdu, CanFrameTriggering.j1939requestable, CanFrameTriggering.identifier, J1939TpPg.requestable, J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
This parameter can be derived from the Pdu and a combination of CanFrame Triggering.j1939requestable and CanFrameTriggering.identifier or J1939TpPg.requestable and J1939TpPg.pgn	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_ 00026]

BSW Module	**BSW Context**
J1939Rm	J1939Rm/J1939RmGeneral
BSW Parameter	**BSW Type**
J1939RmSupportRequest2	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling support of the Request2 PG. Please note: Transfer is not supported.	
Template Description	
Enables support for the Request2 PGN (RQST2).	
M2 Parameter	
J1939Cluster.request2Support	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Rm_ 00073]

C.40 J1939Tp

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration
BSW Parameter	**BSW Type**
J1939TpRxChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This container describes a reception channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.	
Template Description	
A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.	
M2 Parameter	
J1939TpConnection	
Mapping Rule	**Mapping Type**
Create container for each existing J1939TpConnection that is used to receive a NSdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00053]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxCancellationSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable receive cancellation using the API J1939Tp_CancelReceive() for this channel.	
Template Description	
Enable support for Tx/Rx cancellation.	
M2 Parameter	
J1939TpConnection.cancellation	
Mapping Rule	**Mapping Type**
Please note that in the System Template the cancellation support is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00186]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxChannelComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the channel defined by the ComMChannel providing access to the unique channel index ComMChannelId.	
Template Description	
M2 Parameter	
J1939TpNode.connector	
Mapping Rule	**Mapping Type**
Create this reference for each CommunicationConnector that is referenced by a J1939TpNode that is available in the EcuExtract. The CommunicationConnector is referenced by a PhysicalChannel that is aggregated by the CommunicationCluster that corresponds to the ComMChannel in the Ecuc.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00194]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxCmNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection. This N-PDU consumes a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
J1939TpConnection.flowControlPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a received directINPdu that is referenced by the J1939Tp Connection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00128]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxDa	ECUC-INTEGER-PARAM-DEF
BSW Description	
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.	
Template Description	
The TP message is accepted independently of the actually used destination address (DA). Otherwise, only the destination address configured as receiver.tpAddress is accepted. Only derived for the receiving ECU.	
M2 Parameter	
J1939TpNode.tpAddress, J1939TpConnection.acceptVariableDA	
Mapping Rule	**Mapping Type**
If J1939TpConnection.acceptVariableDA is not set to a value or set to false then the this parameter shall be derived from J1939TpConnection.receiver..tpAddress. If J1939TpConnection.accept VariableDA is set to true then this parameter shall not be set to a value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00178]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxDtNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU. This N-PDU consumes a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	



△

J1939TpConnection.dataPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a received NPdu that is referenced by the J1939TpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00117]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxDynamicBlockCalculation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable dynamic calculation of "number of packets that can be sent" value in TP.CM_CTS, based on the size of buffers in upper layers reported via StartOfReception and PduR_J1939TpCopyRxData.	
Template Description	
Enable support for dynamic block size calculation.	
M2 Parameter	
J1939TpConnection.dynamicBs	
Mapping Rule	**Mapping Type**
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00187]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxDynamicBufferRatio	ECUC-INTEGER-PARAM-DEF
BSW Description	
Percentage of available buffer that shall be used for retry. This parameter is only applicable when "J1939TpRxRetrySupport" and "J1939TpRxDynamicBlockCalculation" are enabled.	
Template Description	
Defines usage of available data for dynamic block size calculation when protocol retry is enabled. This attribute describes in percent of available buffer that shall be used for retry.	
M2 Parameter	
J1939TpConnection.bufferRatio	
Mapping Rule	**Mapping Type**
Please note that in the System Template this attribute is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00188]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel

BSW Parameter	BSW Type
J1939TpRxPacketsPerBlock	ECUC-INTEGER-PARAM-DEF

BSW Description

Number of TP.DT frames the receiving J1939Tp module allows the sender to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_CTS frame, and is thus only relevant for reception of messages via CMDT. When J1939TpRxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.

Template Description

Set maximum block size (number of packets in TP.CM_CTS).

M2 Parameter

J1939TpConnection.maxBs

Mapping Rule	Mapping Type
Please note that in the System Template the maximum block size is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00189]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel

BSW Parameter	BSW Type
J1939TpRxPg	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Parameter group received by the J1939 transport layer.

Template Description

A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.

M2 Parameter

J1939TpPg

Mapping Rule	Mapping Type
Create container for each Rx J1939TpPg that is available in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00050]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg

BSW Parameter	BSW Type
J1939TpRxDirectNPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less that 8 bytes. This N-PDU consumes a meta data item of type CAN_ID_32.

Please note: This sub container is only necessary when J1939TpRxPgDynLength is TRUE.

Template Description

▽

△

M2 Parameter	
J1939TpPg.directPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a received directINPdu that is referenced by the J1939TpPg.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00130]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg
BSW Parameter	**BSW Type**
J1939TpRxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the parameters that are relevant for the reception of a specific N-SDU. This N-SDU produces meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8.	
Template Description	

M2 Parameter	
J1939TpPg.sdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00063]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg
BSW Parameter	**BSW Type**
J1939TpRxPgDynLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpRxDirectNPdu is required.	
Template Description	
The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).	
M2 Parameter	
SystemSignal.dynamicLength	
Mapping Rule	**Mapping Type**
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00066]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg
BSW Parameter	**BSW Type**
J1939TpRxPgPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	
PGN of the referenced N-SDUs.	
Template Description	
Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00065]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxProtocolType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.	
Template Description	
Protocol type used by the J1939TpConnection	
M2 Parameter	
J1939TpConnection.tpProtocolType	
Mapping Rule	**Mapping Type**
• J1939TP_PROTOCOL_BAM if tpProtocolType is bam. • J1939TP_PROTOCOL_CMDT if tpProtocolType is cmdt.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00029]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxRetrySupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable support for triggering repetition of failed transmission using TP.CM_CTS with a packet number that has already been sent. Retransmission is triggered when a sequence number is missing or a timeout occurs during reception.	
Template Description	
Enable support for protocol retry.	
M2 Parameter	
J1939TpConnection.retry	
Mapping Rule	**Mapping Type**



△

Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00185]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpRxSa	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with Meta Data containing the SA.	
Template Description	
The TP message is accepted independently of the actually used source address (SA). Otherwise, only the source address configured as transmitter.tpAddress is accepted. Only derived for the receiving ECU.	
M2 Parameter	
J1939TpNode.tpAddress, J1939TpConnection.acceptVariableSA	
Mapping Rule	**Mapping Type**
If J1939TpConnection.acceptVariableSA is not set to a value or set to false then the this parameter shall be derived from J1939TpConnection.transmitter.tpAddress. If J1939TpConnection.accept VariableSA is set to true then this parameter shall not be set to a value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00179]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel
BSW Parameter	**BSW Type**
J1939TpTxFcNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. This N-PDU produces a meta data item of type CAN_ID_32. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.	
Template Description	
M2 Parameter	
J1939TpConnection.flowControlPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a received FlowControlNPdu that is referenced by the J1939Tp Connection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00135]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration

BSW Parameter	BSW Type
J1939TpTxChannel	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This container describes a transmission channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.

Template Description
A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.

M2 Parameter
J1939TpConnection

Mapping Rule	Mapping Type
Create container for each existing J1939TpConnection that is used to transmit a NSdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00059]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel

BSW Parameter	BSW Type
J1939TpRxFcNPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. This N-PDU consumes a meta data item of type CAN_ID_32. Please note: This sub container is only required when J1939TpTxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.

Template Description

M2 Parameter
J1939TpConnection.flowControlPdu

Mapping Rule	Mapping Type
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939Tp Connection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00144]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel

BSW Parameter	BSW Type
J1939TpTxBamTiming	ECUC-FLOAT-PARAM-DEF

BSW Description
If configured, TP.DT frames of a BAM transmission will be sent with the configured delay in seconds. If not configured, 0,05 seconds will be used.

Template Description

▽

△

Timing of TP.DT frames of a BAM transmission. Range 10ms to 50ms, if not configured 50ms are assumed. Relevant for tp ProtocolTypes bam, bam_cmdt, fd_bam, and fd_bam_cmdt.	
M2 Parameter	
J1939TpConnection.bamTiming	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00197]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxCancellationSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable transmit cancellation using the API J1939Tp_CancelTransmit() for this channel.	
Template Description	
Enable support for Tx/Rx cancellation.	
M2 Parameter	
J1939TpConnection.cancellation	
Mapping Rule	**Mapping Type**
Please note that in the System Template the cancellation support is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00192]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxChannelComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the channel defined by the ComMChannel providing access to the unique channel index ComMChannelId.	
Template Description	
M2 Parameter	
J1939TpNode.connector	
Mapping Rule	**Mapping Type**
Create this reference for each CommunicationConnector that is referenced by a J1939TpNode that is available in the EcuExtract. The CommunicationConnector is referenced by a PhysicalChannel that is aggregated by the CommunicationCluster that corresponds to the ComMChannel in the Ecuc.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00195]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxCmNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	
J1939TpConnection.flowControlPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939Tp Connection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00138]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxDa	ECUC-INTEGER-PARAM-DEF
BSW Description	
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.	
Template Description	
The TP message is sent with variable destination address (DA). Otherwise, the destination address configured as receiver.tp Address is always used. Only derived for the transmitting ECU.	
M2 Parameter	
J1939TpConnection.useVariableDA, J1939TpNode.tpAddress	
Mapping Rule	**Mapping Type**
If J1939TpConnection.acceptVariableDA is not set to a value or set to false then the this parameter shall be derived from J1939TpConnection.receiver..tpAddress. If J1939TpConnection.useVariable DA is set to true then this parameter shall not be set to a value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00180]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxDtNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU. This N-PDU produces a meta data item of type CAN_ID_32.	
Template Description	
M2 Parameter	



△

J1939TpConnection.dataPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a transmitted NPdu that is referenced by the J1939TpConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00142]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxDynamicBlockCalculation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable dynamic calculation of "maximum number of packets that can be sent" value in TP.CM_RTS, based on the available amount of data in upper layers reported via PduR_J1939TpCopyTxData.	
Template Description	
Enable support for dynamic block size calculation.	
M2 Parameter	
J1939TpConnection.dynamicBs	
Mapping Rule	**Mapping Type**
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00191]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxMaxPacketsPerBlock	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of TP.DT frames the transmitting J1939Tp module is ready to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_RTS frame, and is thus only relevant for transmission of messages via CMDT. When J1939TpTxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.	
Template Description	
Set maximum for expected block size (maximum number of packets in TP.CM_RTS).	
M2 Parameter	
J1939TpConnection.maxExpBs	
Mapping Rule	**Mapping Type**
Please note that in the System Template the maximum for expected block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00190]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxPg	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Parameter group transmitted by the J1939 transport layer.	
Template Description	
A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.	
M2 Parameter	
J1939TpPg	
Mapping Rule	**Mapping Type**
Create container for each Tx J1939TpPg that is available in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00070]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	**BSW Type**
J1939TpTxDirectNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less that 8 bytes. This N-PDU produces a meta data item of type CAN_ID_32. Please note: This sub container is only necessary when J1939TpTxPgDynLength is TRUE.	
Template Description	
M2 Parameter	
J1939TpPg.directPdu	
Mapping Rule	**Mapping Type**
Information can be derived from a transmitted directlNPdu that is referenced by the J1939TpPg.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00140]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	**BSW Type**
J1939TpTxDirectPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
If configured, and J1939TpTxUseNonDefaultTpPriority is enabled for the enclosing J1939TpTxChannel, this value will be used as priority of transmitted direct frames, otherwise the default priority 6 will be used.	
Template Description	
If configured, direct frames shall be sent with this priority instead of the priority configured in the CAN ID.	
M2 Parameter	
J1939TpPg.txDirectPduPriority	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00200]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	**BSW Type**
J1939TpTxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the parameters that are relevant for the transmission of a specific N-SDU. This N-SDU consumes meta data items of type SOURCE_ADDRESS_16, TARGET_ADDRESS_16, and PRIORITY_8.	
Template Description	
M2 Parameter	
J1939TpPg.sdu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00147]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	**BSW Type**
J1939TpTxPgDynLength	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpTxDirectNPdu is required.	
Template Description	
The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).	
M2 Parameter	
SystemSignal.dynamicLength	
Mapping Rule	**Mapping Type**
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00148]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg
BSW Parameter	**BSW Type**
J1939TpTxPgPGN	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

PGN of the referenced N-SDUs.	
Template Description	
Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a CanFrameTriggering with an identifier.	
M2 Parameter	
J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00150]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
If configured, and J1939TpTxUseNonDefaultTpPriority is enabled, this value will be used as priority of transmitted TP.CM and TP.DT frames, otherwise the default priority 7 will be used.	
Template Description	
If configured, TP.CM and TP.DT frames shall be sent with this priority instead of the default priority 7. Range 0 to 7.	
M2 Parameter	
J1939TpConnection.txPduPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00199]

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxProtocolType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.	
Template Description	
Protocol type used by the J1939TpConnection	
M2 Parameter	
J1939TpConnection.tpProtocolType	
Mapping Rule	**Mapping Type**
• J1939TP_PROTOCOL_BAM if tpProtocolType is bam. • J1939TP_PROTOCOL_CMDT if tpProtocolType is cmdt.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00137]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxRetrySupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable support for repetition of failed transmission using TP.CM_CTS with a packet number that has already been sent. Retransmission is handled via the retry feature of PduR_J1939TpCopyTxData.	
Template Description	
Enable support for protocol retry.	
M2 Parameter	
J1939TpConnection.retry	
Mapping Rule	**Mapping Type**
Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00193]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxSa	ECUC-INTEGER-PARAM-DEF
BSW Description	
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with Meta Data containing the SA.	
Template Description	
The TP message is sent with variable source address (SA). Otherwise, the source address configured as transmitter.tp Address is always used. Only derived for the transmitting ECU.	
M2 Parameter	
J1939TpConnection.useVariableSA, J1939TpNode.tpAddress	
Mapping Rule	**Mapping Type**
If J1939TpConnection.useVariableSA is not set to a value or set to false then the this parameter shall be derived from J1939TpConnection.transmitter.tpAddress. If J1939TpConnection.use VariableSA is set to true then this parameter shall not be set to a value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00181]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxUseNonDefaultTpPriority	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables/disables non-default priority of transmitted TP.CM and TP.DT frames. If enabled, the priority will be set according to J1939TpTxPriority or to the priority provided by the upper layer via meta data.	
Template Description	
If configured, TP.CM and TP.DT frames shall be sent with this priority instead of the default priority 7. Range 0 to 7.	



△

M2 Parameter	
J1939TpConnection.txPduPriority	
Mapping Rule	**Mapping Type**
If txPduPriority exists, J1939TpTxChannel.J1939TpTxUseNonDefaultTpPriority is created and set to True.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00198]

BSW Module	BSW Context
J1939Tp	J1939Tp/J1939TpGeneral
BSW Parameter	**BSW Type**
J1939TpCancellationSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable transmit and receive cancellation. The APIs J1939Tp_CancelTransmit() and J1939Tp_CancelReceive() will only be available when this parameter is enabled.	
Template Description	
Enable support for Tx/Rx cancellation.	
M2 Parameter	
J1939TpConnection.cancellation	
Mapping Rule	**Mapping Type**
Please note that in the System Template the cancellation support is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_J1939Tp_00174]

C.41 Lin

BSW Module	BSW Context
Lin	Lin/LinGlobalConfig
BSW Parameter	**BSW Type**
LinChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration (parameters) of the LIN Controller(s).	
Template Description	
A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one physical channel. Bus systems like CAN and LIN only have exactly one PhysicalChannel. A FlexRay cluster may have more than one PhysicalChannels that may be used in parallel for redundant communication. An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.#	
M2 Parameter	
PhysicalChannel	
Mapping Rule	**Mapping Type**

▽

△

A LinChannel container is constructed per CommunicationConnector belonging to the CommunicationController associated with the owning Lin Module container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Lin_00069]

BSW Module	**BSW Context**
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	**BSW Type**
LinChannelBaudRate	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the baud rate of the LIN channel	
Template Description	
Channels speed in bits/s.	
M2 Parameter	
`CommunicationCluster.baudrate`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Lin_00180]

BSW Module	**BSW Context**
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	**BSW Type**
LinChannelWakeupSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if the LIN hardware channel supports wake up functionality	
Template Description	
Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.	
M2 Parameter	
`CommunicationController.wakeUpByControllerSupported`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Lin_00182]

BSW Module	**BSW Context**
Lin	Lin/LinGlobalConfig/LinChannel
BSW Parameter	**BSW Type**
LinNodeType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies the LIN node type of this channel.	

▽

△

Template Description	
LinMaster: Describing the properties of the refering ecu as a LIN master. **LinSlave:** Describing the properties of the referring ecu as a LIN slave.	
M2 Parameter	
LinMaster, LinSlave	
Mapping Rule	**Mapping Type**
Set to MASTER if the CommunicationController is defined as a LinMaster. Set to SLAVE if the CommunicationController is defined as a LinSlave. In the System Template the LinMaster/LinSlave is connected to the LinChannel via a CommunicationConnector.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Lin_00191]

C.42 LinIf

BSW Module	BSW Context	
LinIf	LinIf	
BSW Parameter		**BSW Type**
LinIfGeneral		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the general parameters of LIN Interface module.		
Template Description		
LIN specific attributes		
M2 Parameter		
LinCluster		
Mapping Rule		**Mapping Type**
Container shall be created if the ECU is connected to a LIN Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00019]

BSW Module	BSW Context	
LinIf	LinIf	
BSW Parameter		**BSW Type**
LinIfGlobalConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the global configuration parameters of the LinIf.		
Template Description		
LIN specific attributes		
M2 Parameter		
LinCluster		
Mapping Rule		**Mapping Type**
Container shall be created if the ECU is connected to a LIN Cluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00020]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig	
BSW Parameter		**BSW Type**
LinIfChannel		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes each LIN channel the LinIf is connected to.		
Template Description		
The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.		
M2 Parameter		
`CommunicationConnector`		
Mapping Rule		**Mapping Type**
Container shall be created if the CommunicationConnector belonging to the ECU is connected to a LinChannel.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00364]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfBusIdleTimeoutPeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
Bus idle timeout in seconds. According to the LIN protocol specification, the bus idle timeout period shall be in range [4, 10] seconds.		
Template Description		
This attribute shall be used to set an idle timeout period for the enclosing LinPhysicalChannel.		
M2 Parameter		
`LinPhysicalChannel.busIdleTimeoutPeriod`		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00655]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfFrame		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Generic container for all types of LIN frames.		
Template Description		
LIN specific attributes to the FrameTriggering		
M2 Parameter		
`LinFrameTriggering`		
Mapping Rule		**Mapping Type**



△

Create container for each LinFrameTriggering aggregated by the PhysicalChannel representing the regarded LIN channel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00367]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	**BSW Type**
LinIfChecksumType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Type of checksum that the frame is using. This parameter is optional because in case of sporadic frames it should not be set.	
Template Description	
Type of checksum that the frame is using. This attribute is optional because in case of sporadic frames it should not be set.	
M2 Parameter	
`LinFrameTriggering.linChecksum`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00005]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu
BSW Parameter	**BSW Type**
LinIfFixedFrameSduByte	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents a byte within the 8 byte array.	
Template Description	
LinSlaveConfig: Node attributes of LIN slaves that are handled by the LinMaster. In the System Description LIN slaves may be described in the context of the Lin Master. In an ECU Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the LinMaster. The values of attributes of LinSlaveConfig and the corresponding LinSlave shall be identical (if both are defined in a System Description). **LinSlave:** Describing the properties of the referring ecu as a LIN slave.	
M2 Parameter	
`LinSlaveConfig`, `LinSlave`	
Mapping Rule	**Mapping Type**

▽

△

For LinConfigurationEntry elements the byte array depends on the LinIfFrameType in the LinIf Entry. The parameters ConfiguredNad, SupplierId, and Function can be derived from the LinSlave Config that is referenced from the LinConfigurationEntry in the assignedLinSlaveConfig role or from the LinSlave referenced via assignedController. AssignFrameId/UnassignFrameId : Byte Array: ConfiguredNad, 0x06, 0xb1, LSB of SupplierId, MSB of SupplierId, LSB of MessageId, MSB of MessageId, protected identifier of LinFrame Triggering. The LinFrameTriggering is referenced by AssignFrameId/UnassignFrameId directly in the role assignedFrameTrriggering/unassignedFrameTriggering. The MessageId shall be derived in the following way: in case that the AssignFrameId/UnassignFrameId refers to a LinSlave in the role assigned Controller the messageId of the AssignFrameId/UnassignFrameId is located in the LinConfigurable Frame that references the same LinFrame as the LinFrameTriggering that is referenced by the AssignFrameId/UnassignFrameId and that is aggregated by the LinCommunicationConnector in role linConfigurableFrame that points to this LinSlave in the role commController. in case that the AssignFrameId/UnassignFrameId refers to a LinSlaveConfigIdent in the role assignedLinSlaveConfig the messageId is located in the LinConfigurableFrame that references the same LinFrame as the LinFrameTriggering that is referenced by the AssignFrameId/Unassign FrameId and that is aggregated by the referenced LinSlaveConfig. AssignFrameIdRange: Byte Array for Request: ConfiguredNad, 0x06, 0xB7, start index PID (index), PID (index+1), PID (index+2), PID (index+3), The four PID values shall be derived from AssignFrameIdRange.framePid element. The Frame Pid.index describes the Position of the pid value in the AssignFrameIdRange command. AssignNad: Byte Array for Request: initialNad, 0x06, 0xB0, Supplier ID LSB, Supplier ID MSB, Function ID LSB, Function ID MSB, newNad Byte Array for Response: initialNad, 0x01, 0xF0, 0x FF, 0xFF, 0xFF, 0xFF, 0xFF The intialNad shall be derived either: from the LinCommunicationConnector that references the LinSlave that in turn is referenced by the AssignNad in the role assignedController. from the LinSlaveConfig that is referenced by the Assign Nad LinConfigurationEntry in the role assignedLinSlaveConfig The newNad shall be derived from the AssignedNad LinConfigurationEntry element itself. ConditionalChangeNad: Byte Array for Request: ConfiguredNad, 0x06, 0xb3, Id, Byte, Mask, Invert, newNad The id, byte, mask, invert, newNad shall be derived from the ConditionalChangeNad Lin ConfigurationEntry element itself. DataDumpEntry: Byte Array for Request: ConfiguredNad, 0x06, 0xB4, Byte Value, Byte Value, Byte Value, Byte Value, Byte Value Byte Array for Response: ConfiguredNad, 0x06, 0xF4, Byte Value, Byte Value, Byte Value, Byte Value, Byte Value SaveConfigurationEntry: Byte Array for Request: ConfiguredNad, 0x01, 0xb6, 0xff, 0xff, 0xff, 0xff, 0xff Byte Array for Response: ConfiguredNad, 0x01, 0xF6, 0xff, 0xff, 0xff, 0xff, 0xff	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00013]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	**BSW Type**
LinIfFrameId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID of the LIN frame. The Protected ID including parity is calculated by the generation tool.	
Template Description	
To describe a frames identifier on the communication system, usually with a fixed identifierValue. For LinSporadicFrames the attribute shall be ignored.	

▽

△

M2 Parameter	
LinFrameTriggering.identifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00638]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	**BSW Type**
LinIfFrameIndex	ECUC-INTEGER-PARAM-DEF
BSW Description	
PID index of the frame. This index is used in the AssignFrameIdentifierRange node configuration service to identify the frame(s) to which a new PID shall be assigned. It corresponds to the order of the frames in the configurable frames list in the node attributes section of the LDF / NCF of the slave node. Only relevant for LIN slave nodes.	
Template Description	
This attribute is used to order the elements and allows an assignment of Pids to ConfigurableFrames that are defined in the slave.	
M2 Parameter	
LinOrderedConfigurableFrame.index	
Mapping Rule	**Mapping Type**
LinIfFrameIndex shall be derived from the list of LinOrderedConfigurableFrame elements that are ordered according to the index attribute. The LinIfFrameIndex = 0 shall be derived from the first element in the list. The LinIfFrameIndex = 1 shall be derived from the second element in the list and so on. Please note that the index settings in the System Description may contain gaps. In other words, if for example the System Description defines LinCommunicationConnector.linOrderedConfigurable Frame with index = 2 and a second linOrderedConfigurableFrame with index = 4 then the LinIf FrameIndex in the Ecuc needs to be set to 0 for the first Frame and to 1 for the second Frame.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00653]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	**BSW Type**
LinIfFrameType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the type of frame (e.g. sporadic frame). For master nodes, all frame types are permitted. A sporadic slot may be used by a set of unconditional frames in the role of substitution frames. For slave nodes, only following types are permitted: Uncondtional, MRF, SRF, Event-triggered. An event-triggered slot may be used by a set of unconditional frames in the role of substitution frames.	
Template Description	

▽

△

FrameTriggering:
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent.

For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.

LinUnconditionalFrame:
Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data.

LinEventTriggeredFrame:
An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response.

The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable.

The event controlled frame shall not contain any Pdus.

LinSporadicFrame:
A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus.

AssignFrameId:
Schedule entry for an Assign Frame Id master request.

UnassignFrameId:
Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier.

AssignNad:
Schedule entry for an Assign NAD master request.

FreeFormat:
Representing freely defined data.

ConditionalChangeNad:
Generates an conditional change NAD request. See ISO 17987 protocol specification for more information.

SaveConfigurationEntry:
This service is used to notify a slave node to store its configuration.

AssignFrameIdRange:
AssignFrameIdRange generates an assign frame PID range request.

M2 Parameter	
FrameTriggering, LinUnconditionalFrame, LinEventTriggeredFrame, LinSporadicFrame, AssignFrameId, UnassignFrameId, AssignNad, FreeFormat, ConditionalChangeNad, SaveConfigurationEntry, AssignFrameIdRange	
Mapping Rule	**Mapping Type**
see details in EnumerationLiteralDef descritpions	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00017]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
ASSIGN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
AssignFrameId	
Template Description	
Schedule entry for an Assign Frame Id master request.	
M2 Parameter	
AssignFrameId	

▽

△

Mapping Rule	Mapping Type
Use FrameType "Assign" if ScheduleEntry is an "AssignFrameId".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
ASSIGN_FRAME_ID_RANGE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
AssignFrameIdRange	
Template Description	
AssignFrameIdRange generates an assign frame PID range request.	
M2 Parameter	
AssignFrameIdRange	
Mapping Rule	**Mapping Type**
Use FrameType "Assign_Frame_Id_Range" if ScheduleEntry is an "AssignFrameIdRange".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
ASSIGN_NAD	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
AssignNAD	
Template Description	
Schedule entry for an Assign NAD master request.	
M2 Parameter	
AssignNad	
Mapping Rule	**Mapping Type**
Use FrameType "Assign_NAD" if ScheduleEntry is an "AssignNad".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
CONDITIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Conditional Change NAD	
Template Description	
Generates an conditional change NAD request. See ISO 17987 protocol specification for more information.	
M2 Parameter	

▽

△

ConditionalChangeNad	
Mapping Rule	**Mapping Type**
Use FrameType "CONDITIONAL" if ScheduleEntry is an "ConditionalChangeNad".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
EVENT_TRIGGERED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Event triggered frame	
Template Description	
An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response. The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable. The event controlled frame shall not contain any Pdus.	
M2 Parameter	
LinEventTriggeredFrame	
Mapping Rule	**Mapping Type**
Derive the type from System Description	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
FREE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
FreeFormat	
Template Description	
Representing freely defined data.	
M2 Parameter	
FreeFormat	
Mapping Rule	**Mapping Type**
Use FrameType "Free Format" if ScheduleEntry is a "FreeFormatEntry".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
MRF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	

▽

△

Master Request Frame	
Template Description	
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.	
M2 Parameter	
FrameTriggering	
Mapping Rule	**Mapping Type**
Use common Frame for Master Request.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
SAVE_CONFIGURATION	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SaveConfiguration	
Template Description	
This service is used to notify a slave node to store its configuration.	
M2 Parameter	
SaveConfigurationEntry	
Mapping Rule	**Mapping Type**
Use FrameType "Save_Configuration" is an "SaveConfiguration".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
SPORADIC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Sporadic slot	
Template Description	
A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus.	
M2 Parameter	
LinSporadicFrame	
Mapping Rule	**Mapping Type**
Derive the type from System Description	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
SRF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Slave Response Frame	
Template Description	
The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.	
M2 Parameter	
FrameTriggering	
Mapping Rule	**Mapping Type**
Use common Frame for Slave Response.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
UNASSIGN	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
UnassignFrameId	
Template Description	
Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier.	
M2 Parameter	
UnassignFrameId	
Mapping Rule	**Mapping Type**
Use FrameType "Unassign" if ScheduleEntry is an "UnassignFrameId".	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFrameType
BSW Parameter	**BSW Type**
UNCONDITIONAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Unconditional Frame	
Template Description	
Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data.	
M2 Parameter	
LinUnconditionalFrame	
Mapping Rule	**Mapping Type**
Derive the type from System Description	full



△

Mapping Status	ECUC Parameter ID
valid	

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter	**BSW Type**	
LinIfPduDirection	ECUC-CHOICE-CONTAINER-DEF	
BSW Description		
Direction of the frame		
Template Description		
LIN specific attributes to the FrameTriggering		
M2 Parameter		
LinFrameTriggering		
Mapping Rule		**Mapping Type**
Create container for each existing LinFrame.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00027]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter	**BSW Type**	
LinIfRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
represents a received PDU/frame		
Template Description		
M2 Parameter		
LinFrameTriggering.framePort		
Mapping Rule		**Mapping Type**
Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "in" FramePort		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00035]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter	**BSW Type**	
LinIfTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF	
BSW Description		
represents a transmitted PDU/frame		
Template Description		
M2 Parameter		
LinFrameTriggering.framePort		
Mapping Rule		**Mapping Type**

▽

△

Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "out" FramePort	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00049]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame
BSW Parameter	**BSW Type**
LinIfSubstitutionFrames	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
List of sporadic frames that can be sent in a sporadic frame slot (master node) or list of unconditional frames that can be sent in an event-triggered frame slot (slave node).	
Template Description	
M2 Parameter	
LinSporadicFrame.substitutedFrame	
Mapping Rule	**Mapping Type**
emulate reference from System Description	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00042]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfSubstitutionFrames
BSW Parameter	**BSW Type**
LinIfFramePriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Priority of sporadic frame in a master node or of event-triggered frame in slave node.	
Template Description	
M2 Parameter	
LinSporadicFrame.substitutedFrame	
Mapping Rule	**Mapping Type**
In the System Description the priority is described by the Order of the UnconditionalFrames	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00513]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	**BSW Type**
LinIfMainFunctionPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the interval of calls to main functions per channel in seconds.	
Template Description	

▽

△

Time base is mandatory for the master. It is not used for slaves. LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time." The time base shall be specified AUTOSAR conform in seconds.
M2 Parameter
LinMaster.timeBase

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00639]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel

BSW Parameter	**BSW Type**
LinIfNodeType	ECUC-CHOICE-CONTAINER-DEF

BSW Description
This container defines the LIN node type of this channel.
Template Description
LinMaster: Describing the properties of the refering ecu as a LIN master. **LinSlave:** Describing the properties of the referring ecu as a LIN slave.
M2 Parameter
LinMaster, LinSlave

Mapping Rule	**Mapping Type**
Set to MASTER if the CommunicationController is defined as a LinMaster. Set to SLAVE if the CommunicationController is defined as a LinSlave. In the System Template the LinMaster/LinSlave is connected to the LinChannel via a CommunicationConnector.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00654]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType

BSW Parameter	**BSW Type**
LinIfMaster	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Each Master can only be connected to one physical channel. This could be compared to the Node parameter in a LDF file.
Template Description
Describing the properties of the refering ecu as a LIN master.
M2 Parameter
LinMaster

Mapping Rule	**Mapping Type**
Create container if the regarded ECU contains a CommunicationController that is defined as a Lin Master. In the System Template the LinMaster is connected to the LinChannel via a CommunicationConnector.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00512]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfMaster
BSW Parameter	**BSW Type**
LinIfJitter	ECUC-FLOAT-PARAM-DEF
BSW Description	
The jitter specifies the differences between the maximum and minimum delay from time base tick to the header sending start point in seconds.	
Template Description	
The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves. LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)." The jitter shall be specified AUTOSAR conform in seconds.	
M2 Parameter	
LinMaster.timeBaseJitter	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00629]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType
BSW Parameter	**BSW Type**
LinIfSlave	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes all parameters which are only relevant for a LIN Slave node.	
Template Description	
Describing the properties of the referring ecu as a LIN slave.	
M2 Parameter	
LinSlave	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00649]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave
BSW Parameter	**BSW Type**
LinIfLinProtocolVersion	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the LIN protocol version of the slave node. This information is relevant for the LIN conformance test execution.	
Template Description	
LinSlaveConfig.protocolVersion: Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different. **LinCommunicationController.protocolVersion:** Version specifier for a communication protocol.	
M2 Parameter	

▽

△

LinSlaveConfig.protocolVersion, LinCommunicationController.protocolVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00647]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	**BSW Type**
LinIfConfiguredNAD	ECUC-INTEGER-PARAM-DEF
BSW Description	
Slave node configured NAD.	
Template Description	
LinSlave.configuredNad: To distinguish LIN slaves that are used twice or more within the same cluster. **LinSlaveConfig.configuredNad:** To distinguish LIN slaves that are used twice or more within the same cluster.	
M2 Parameter	
LinSlave.configuredNad, LinSlaveConfig.configuredNad	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00643]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	**BSW Type**
LinIfFunctionId	ECUC-INTEGER-PARAM-DEF
BSW Description	
LIN function Id.	
Template Description	
LinSlave.functionId: LIN function ID **LinSlaveConfig.functionId:** LIN function ID.	
M2 Parameter	
LinSlave.functionId, LinSlaveConfig.functionId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00646]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	**BSW Type**

▽

△

LinIfInitialNAD	ECUC-INTEGER-PARAM-DEF
BSW Description	
Slave node initial NAD.	
Template Description	
LinSlave.initialNad: This attribute represents the initial NAD. **LinSlaveConfig.initialNad:** Initial NAD of the LIN slave.	
M2 Parameter	
LinSlave.initialNad, LinSlaveConfig.initialNad	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00642]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification

BSW Parameter	**BSW Type**
LinIfNasTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
N_As timeout in seconds.	
Template Description	
Value of the N_AS timeout. Unit: seconds.	
M2 Parameter	
LinSlave.nasTimeout	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00644]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification

BSW Parameter	**BSW Type**
LinIfSupplierId	ECUC-INTEGER-PARAM-DEF
BSW Description	
LIN consortium or ISO LIN supplier Id.	
Template Description	
LinSlave.supplierId: LIN Supplier ID **LinSlaveConfig.supplierId:** LIN Supplier ID.	
M2 Parameter	
LinSlave.supplierId, LinSlaveConfig.supplierId	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00645]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave/LinIfNodeConfigurationIdentification
BSW Parameter	**BSW Type**
LinIfVariantId	ECUC-INTEGER-PARAM-DEF
BSW Description	
LIN variant Id.	
Template Description	
LinSlave.variantId: Specifies the Variant ID **LinSlaveConfig.variantId:** Specifies the Variant ID.	
M2 Parameter	
LinSlave.variantId, LinSlaveConfig.variantId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00641]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfNodeType/LinIfSlave
BSW Parameter	**BSW Type**
LinIfResponseErrorSignal	ECUC-REFERENCE-DEF
BSW Description	
Reference to the response_error signal. Mandatory for all LIN 2.x and ISO LIN slave nodes, not relevant for LIN 1.3 slave nodes.	
Template Description	
This ISignal shall be taken to transport the responseError bit.	
M2 Parameter	
LinErrorResponse.responseError	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00648]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	**BSW Type**
LinIfScheduleChangeNextTimeBase	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables/disables the switch to a new schedule table at the start of the next time base after status check. True: LinIf selects a new schedule table in next main function. Only applicable for LIN Master nodes.	
Template Description	
This attribute defines the point in time where a schedule table switch is performed. If this attribute is set to false or not present, the schedule table shall be switched after the current entry of the active schedule table is ended. If this attribute is enabled, the schedule table shall be switched when message transmission or reception within an entry has been completed, ensured by status checks for transmission and reception.	
M2 Parameter	



△

LinCommunicationConnector.scheduleChangeNextTimeBase	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00640]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel
BSW Parameter	**BSW Type**
LinIfScheduleTable	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes a schedule table. Each LinIfChannel may have several schedule tables. Each schedule table can only be connected to one channel. Mandatory for LIN Master nodes. The SHORT-NAME of the LinIfScheduleTable container represents the symbolic name of the schedule table.	
Template Description	
The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame.	
M2 Parameter	
LinScheduleTable	
Mapping Rule	**Mapping Type**
Create container for each ScheduleTable that is defined for this channel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00365]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable
BSW Parameter	**BSW Type**
LinIfEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes an entry in the schedule table (also known as Frame Slot).	
Template Description	
Table entry in a LinScheduleTable. Specifies what will be done in the frame slot.	
M2 Parameter	
ScheduleTableEntry	
Mapping Rule	**Mapping Type**
Each RelativelyScheduledTiming element in the System Description requires the creation of a LinIf Entry. RelativelyScheduledTiming.scheduleTablle decides to which schedule table the LinIfEntry belongs.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00366]

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry
BSW Parameter	**BSW Type**
LinIfCollisionResolvingRef	ECUC-REFERENCE-DEF

▽

△

BSW Description	
Reference to the schedule table, which resolves the collision. This parameter is only used if the referenced frames are event triggered frames.	
Template Description	
M2 Parameter	
LinEventTriggeredFrame.collisionResolvingSchedule	
Mapping Rule	**Mapping Type**
Emulate the reference from the System Description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00007]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	**BSW Type**	
LinIfDelay	ECUC-FLOAT-PARAM-DEF	
BSW Description		
Delay to next entry in schedule table in seconds.		
Template Description		
Relative delay between this tableEntry and the start of the successor in the schedule table in seconds.		
M2 Parameter		
ScheduleTableEntry.delay		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00009]

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter	**BSW Type**	
LinIfEntryIndex	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Position of the Frame Entry in the Schedule Table. The first entry index in the schedule table is 0.		
Template Description		
Relative position in the schedule table. The first entry index in the schedule table is 0.		
M2 Parameter		
ScheduleTableEntry.positionInTable		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinIf_00011]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry
BSW Parameter	**BSW Type**
LinIfFrameRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the frames that belong to this schedule table entry.	
Template Description	
M2 Parameter	
ApplicationEntry.frameTriggering	
Mapping Rule	**Mapping Type**
Emulate reference from the System Description	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00016]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable
BSW Parameter	**BSW Type**
LinIfResumePosition	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines where a RUN_CONTINUOUS schedule table shall proceed in case it has been interrupted by a RUN_ONCE table.	
Template Description	
Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.	
M2 Parameter	
LinScheduleTable.resumePosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00033]

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable
BSW Parameter	**BSW Type**
LinIfRunMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The schedule table can be executed in two different modes.	
Template Description	
The schedule table can be executed in two different modes.	
M2 Parameter	
LinScheduleTable.runMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinIf_00034]

C.43 LinTp

BSW Module	BSW Context
LinTp	LinTp
BSW Parameter	**BSW Type**
LinTpGlobalConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the global configuration parameters of the LinTp.	
Template Description	
TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.	
M2 Parameter	
LinTpNode	
Mapping Rule	**Mapping Type**
Create container if the regarded ECU is a LinTpNode.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00056]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	**BSW Type**
LinTpChannelConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the channel specific configuration parameters of LinTp.	
Template Description	
M2 Parameter	
LinTpNode.connector	
Mapping Rule	**Mapping Type**
Create this container for each CommunicationConnector that is referenced by a LinTpNode that is available in the EcuExtract. The CommunicationConnector is referenced by a PhysicalChannel that is aggregated by the CommunicationCluster that corresponds to the ComMChannel in the Ecuc that is referenced by the LinTpChannelRef in the LinTpChannelConfig.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00071]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	**BSW Type**
LinTpDropNotRequestedNad	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Configures if TP Frames of not requested LIN-Slaves are dropped or not. TRUE: Drop TP Frames of not requested LIN-Slaves FALSE: Keep TP Frames of not requested LIN-Slaves Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
Configures if TP Frames of not requested LIN-Slaves are dropped or not.	
M2 Parameter	

▽

△

LinTpNode.dropNotRequestedNad	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00072]

BSW Module	**BSW Context**
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	**BSW Type**
LinTpMaxNumberOfRespPendingFrames	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configures the maximum number of allowed response pending frames. Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
Configures the maximum number of allowed response pending frames.	
M2 Parameter	
LinTpNode.maxNumberOfRespPendingFrames	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00624]

BSW Module	**BSW Context**
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	**BSW Type**
LinTpP2Max	ECUC-FLOAT-PARAM-DEF
BSW Description	
P2*max timeout when a response pending frame is expected in seconds. Note that the minimum value of LinTpP2Max shall be more than or equal to the value of LinTpP2Timing. Only used for LIN Master nodes, ignored for slave nodes.	
Template Description	
After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.	
M2 Parameter	
LinTpNode.p2Max	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00622]

BSW Module	**BSW Context**
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig
BSW Parameter	**BSW Type**
LinTpP2Timing	ECUC-FLOAT-PARAM-DEF
BSW Description	
Definition of the P2max timeout observation parameter in seconds. Only used for LIN Master nodes, ignored for slave nodes.	

▽

△

Template Description	
P2 timeout observation parameter.	
M2 Parameter	
LinTpNode.p2Timing	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00625]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	**BSW Type**
LinTpRxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container exists once for each received N-SDU on any channel the node is connected to. This N-SDU produces meta data items of type LIN_NAD_8.	
Template Description	
M2 Parameter	
LinTpConnection.linTpNSdu	
Mapping Rule	**Mapping Type**
Create container for each NSdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00428]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu
BSW Parameter	**BSW Type**
LinTpNcr	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.	
Template Description	
This attribute defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.	
M2 Parameter	
LinTpConnection.timeoutCr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00632]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu
BSW Parameter	**BSW Type**
LinTpRxNSduNad	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
A N-SDU transported on LIN is identified using the NAD for the specific slave.	
Template Description	
To distinguish LIN slaves that are used twice or more within the same cluster.	
M2 Parameter	
LinSlave.configuredNad	
Mapping Rule	**Mapping Type**
Find connection from NSdu to CommunicationController	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00062]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig
BSW Parameter	**BSW Type**
LinTpTxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container exists once for each transmitted N-SDU on any channel the node is connected to. This N-SDU consumes meta data items of type LIN_NAD_8.	
Template Description	
M2 Parameter	
LinTpConnection.linTpNSdu	
Mapping Rule	**Mapping Type**
Create container for each NSdu that is received by the regarded ECU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00511]

BSW Module	BSW Context
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu
BSW Parameter	**BSW Type**
LinTpNas	ECUC-FLOAT-PARAM-DEF
BSW Description	
Value in seconds of the N_As timeout. N_As is the time for transmission of a LIN frame (any N_PDU) on the part of the sender.	
Template Description	
Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.	
M2 Parameter	
LinTpConnection.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTp_00633]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpNcs		ECUC-FLOAT-PARAM-DEF
BSW Description		
Value in seconds of the performance requirement of N_Cs. N_Cs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.		
Template Description		
The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.		
M2 Parameter		
LinTpConnection.timeoutCs		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinTp_00634]

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpTxNSduNad		ECUC-INTEGER-PARAM-DEF
BSW Description		
A N-SDU transported on LIN is identified using the NAD for the specific slave.		
Template Description		
To distinguish LIN slaves that are used twice or more within the same cluster.		
M2 Parameter		
LinSlave.configuredNad		
Mapping Rule		**Mapping Type**
Find connection from NSdu to CommunicationController		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_LinTp_00066]

C.44 LinTrcv

BSW Module	BSW Context	
LinTrcv	LinTrcv/LinTrcvChannel	
BSW Parameter		**BSW Type**
LinTrcvWakeupByBusUsed		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Is wake up by bus supported? If LIN transceiver hardware does not support wake up by bus value is always FALSE. If LIN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not. TRUE = Is used. FALSE = Is not used.		
Template Description		

▽

△

Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.
M2 Parameter
CommunicationController.wakeUpByControllerSupported

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTrcv_00006]

BSW Module	**BSW Context**
LinTrcv	LinTrcv/LinTrcvGeneral

BSW Parameter	**BSW Type**
LinTrcvWakeUpSupport	ECUC-BOOLEAN-PARAM-DEF

BSW Description
Informs whether wake up is supported or not. In case wake up is not supported by LIN transceiver hardware the setting shall be false. The wake up ability may be switched on or off for each channel of one LIN transceiver by LinTrcvWakeupSourceRef. True: Is used False: Is not used
Template Description
Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.
M2 Parameter
CommunicationController.wakeUpByControllerSupported

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_LinTrcv_00107]

C.45 Mirror

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet

BSW Parameter	**BSW Type**
MirrorDestNetwork	ECUC-CHOICE-CONTAINER-DEF

BSW Description
Destination bus to which frames are sent by the Bus Mirroring module.
Template Description

▽

△

This element defines a bus mirroring in which the traffic from one communication bus (sourceChannel) is forwarded to another one (targetChannel).	
M2 Parameter	
BusMirrorChannelMapping	
Mapping Rule	**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a concrete subclass of BusMirrorChannelMapping in role targetChannel which is available in the System Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00051]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork
BSW Parameter	**BSW Type**
MirrorDestNetworkCan	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Destination bus representing a CAN network.	
Template Description	
This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel.	
M2 Parameter	
BusMirrorChannelMappingCan	
Mapping Rule	**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirro ChannelMappingCan in the role targetChannel which is available in the SystemExtract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00052]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	**BSW Type**
MirrorDestPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
`BusMirrorChannelMapping.targetPduTriggering`	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00055]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
`BusMirrorChannel.busMirrorNetworkId`	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCan
BSW Parameter	**BSW Type**
MirrorStatusCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
Template Description	
CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
M2 Parameter	
BusMirrorChannelMappingCan.mirrorStatusCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00061]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanFD
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanFD
BSW Parameter	**BSW Type**
MirrorDestPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus.	
Template Description	

▽

△

Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00055]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanFD
BSW Parameter	**BSW Type**
MirrorDestTransmissionDeadline	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655 ms.	
Template Description	
BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingIp.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingUserDefined.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.	
M2 Parameter	
BusMirrorChannelMappingFlexray.transmissionDeadline, BusMirrorChannelMappingIp.transmissionDeadline, BusMirrorChannelMappingUserDefined.transmissionDeadline	
Mapping Rule	**Mapping Type**
Please note that this parameter is aggregated in different containers: - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMapping Flexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmission Deadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUser Defined.transmissionDeadline	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00059]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanFD
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanFD
BSW Parameter	**BSW Type**
MirrorStatusCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
Template Description	
CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.	
M2 Parameter	
BusMirrorChannelMappingCan.mirrorStatusCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00061]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanXL
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanXL
BSW Parameter	**BSW Type**
MirrorDestPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00055]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanXL
BSW Parameter	**BSW Type**
MirrorDestTransmissionDeadline	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655 ms.	



△

Template Description
BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingIp.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingUserDefined.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.
M2 Parameter
BusMirrorChannelMappingFlexray.transmissionDeadline, BusMirrorChannelMappingIp.transmissionDeadline, BusMirrorChannelMappingUserDefined.transmissionDeadline

Mapping Rule	Mapping Type
Please note that this parameter is aggregated in different containers: - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMapping Flexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmission Deadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUser Defined.transmissionDeadline	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00059]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCanXL

BSW Parameter	BSW Type
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	

Mapping Rule	Mapping Type
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping.	full

▽

▽

△

△ If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork
BSW Parameter	**BSW Type**
MirrorDestNetworkCdd	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Destination bus representing a user defined network.	
Template Description	
This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a UserDefined targetChannel.	
M2 Parameter	
BusMirrorChannelMappingUserDefined	
Mapping Rule	**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirro ChannelMappingUserDefined in the role targetChannel which is available in the SystemExtract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00062]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	**BSW Type**
MirrorDestPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00055]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	**BSW Type**
MirrorDestTransmissionDeadline	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655 ms.	
Template Description	
BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingIp.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingUserDefined.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.	
M2 Parameter	
BusMirrorChannelMappingFlexray.transmissionDeadline, BusMirrorChannelMappingIp.transmissionDeadline, BusMirrorChannelMappingUserDefined.transmissionDeadline	
Mapping Rule	**Mapping Type**
Please note that this parameter is aggregated in different containers: - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMapping Flexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmission Deadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUser Defined.transmissionDeadline	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00059]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkCdd
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork
BSW Parameter	**BSW Type**
MirrorDestNetworkFlexRay	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Destination bus representing a FlexRay network.	
Template Description	
This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel.	
M2 Parameter	
BusMirrorChannelMappingFlexray	
Mapping Rule	**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirro ChannelMappingFlexray in the role targetChannel which is available in the SystemExtract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00058]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	**BSW Type**
MirrorDestPduFlexRay	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus. For FlexRay, an arbitrary number of I-PDUs can be configured.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00066]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay
BSW Parameter	**BSW Type**
MirrorDestTransmissionDeadline	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655 ms.
Template Description
BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingIp.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingUserDefined.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.
M2 Parameter
BusMirrorChannelMappingFlexray.transmissionDeadline, BusMirrorChannelMappingIp.transmissionDeadline, BusMirrorChannelMappingUserDefined.transmissionDeadline

Mapping Rule	**Mapping Type**
Please note that this parameter is aggregated in different containers: - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMapping Flexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmission Deadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUser Defined.transmissionDeadline	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00059]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkFlexRay

BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	

Mapping Rule	**Mapping Type**

▽

△

If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork
BSW Parameter	**BSW Type**
MirrorDestNetworkIp	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Destination bus representing an IP network.	
Template Description	
This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel.	
M2 Parameter	
BusMirrorChannelMappingIp	
Mapping Rule	**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a BusMirro ChannelMappingIp in the role targetChannel which is available in the SystemExtract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00060]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	

▽

△

BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp
BSW Parameter	**BSW Type**
MirrorDestPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
I-PDU used for transmission of the mirrored frames on the destination bus.	
Template Description	
Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPdu Triggering is supported.	
M2 Parameter	
BusMirrorChannelMapping.targetPduTriggering	
Mapping Rule	**Mapping Type**
Create container for the Pdu that is referenced by the PduTriggering that is referenced by the instance of the BusMirrorChannelMapping in the targetPduTriggering role.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00055]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp
BSW Parameter	**BSW Type**
MirrorDestTransmissionDeadline	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds after which the collection of source frames into the destination frame stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows after 655 ms.	
Template Description	
BusMirrorChannelMappingFlexray.transmissionDeadline: Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingIp.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows. **BusMirrorChannelMappingUserDefined.transmissionDeadline:** Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.	
M2 Parameter	

▽

△

BusMirrorChannelMappingFlexray.transmissionDeadline, BusMirrorChannelMappingIp.transmissionDeadline, BusMirrorChannelMappingUserDefined.transmissionDeadline	
Mapping Rule	**Mapping Type**
Please note that this parameter is aggregated in different containers: - if aggregated by MirrorDestNetworkFlexRay take value from BusMirrorChannelMapping Flexray.transmissionDeadline - if aggregated by MirrorDestNetworkIp take value from BusMirrorChannelMappingIp.transmission Deadline - if aggregated by MirrorDestNetworkCdd take value from BusMirrorChannelMappingUser Defined.transmissionDeadline	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00059]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorDestNetwork/MirrorDestNetworkIp
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet	
BSW Parameter		**BSW Type**
MirrorSourceNetwork		ECUC-CHOICE-CONTAINER-DEF
BSW Description		
Source bus from which frames are received by the Bus Mirroring module.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
BusMirrorChannel.channel		
Mapping Rule		**Mapping Type**
Create a container for each BusMirrorChannel that is composed by an instance of a concrete subclass of BusMirrorChannelMapping in role sourceChannel which is available in the System Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Mirror_00009]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork	
BSW Parameter		**BSW Type**
MirrorSourceNetworkCan		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Source bus representing a CAN network.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
BusMirrorChannel.channel		
Mapping Rule		**Mapping Type**
Create a container for each CanPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Mirror_00010]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan	
BSW Parameter		**BSW Type**
MirrorComMNetworkHandleRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the ComMChannel that represents the bus.		
Template Description		
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.		
M2 Parameter		
BusMirrorChannel.channel		
Mapping Rule		**Mapping Type**



△

This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Rule for remapping a set of CAN IDs.	
Template Description	
This element defines a rule for remapping a set of CAN IDs.	
M2 Parameter	
BusMirrorCanIdRangeMapping	
Mapping Rule	**Mapping Type**
Create Container in case that BusMirrorCanIdRangeMapping is aggregated by BusMirrorChannel MappingCan in the role canIdRangeMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00025]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingDestBaseId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.	
Template Description	
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.	
M2 Parameter	
BusMirrorCanIdRangeMapping.destinationBaseId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00028]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingSourceCanIdCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value to match masked original CAN IDs.	
Template Description	
Value to match masked original CAN IDs.	
M2 Parameter	
BusMirrorCanIdRangeMapping.sourceCanIdCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00026]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingSourceCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Mask applied to original CAN IDs before comparison.	
Template Description	
Mask applied to original CAN IDs before comparison.	
M2 Parameter	
BusMirrorCanIdRangeMapping.sourceCanIdMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00027]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan
BSW Parameter	**BSW Type**
MirrorSourceCanSingleIdMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Rule for remapping a single CAN ID.	
Template Description	
This element defines a rule for remapping a single CAN ID.	
M2 Parameter	
BusMirrorCanIdToCanIdMapping	
Mapping Rule	**Mapping Type**
Create container in case that the BusMirrorCanIdToCanIdMapping is aggregated by BusMirrorChannelMappingCan in the role canIdToCanIdMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00022]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanSingleIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanSingleIdMappingDestCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Mapped CAN ID.	
Template Description	
This attribute defines the CanId on the targetChannel.	
M2 Parameter	
BusMirrorCanIdToCanIdMapping.remappedCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00024]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCan/MirrorSourceCanSingleId Mapping

BSW Parameter	BSW Type
MirrorSourceCanSingleIdMappingSourceCanId	ECUC-INTEGER-PARAM-DEF

BSW Description
Original CAN ID.

Template Description
This reference points to the sourceFrame with sourceCanId on the sourceChannel.

M2 Parameter
BusMirrorCanIdToCanIdMapping.sourceCanId

Mapping Rule	Mapping Type
Take the value from the identifier attribute of the referenced CanFrameTriggering.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00023]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork

BSW Parameter	BSW Type
MirrorSourceNetworkCanFD	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Source bus representing a CAN FD network.

Template Description
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.

M2 Parameter
BusMirrorChannel.channel

Mapping Rule	Mapping Type
Create a container for each CanPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00072]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD

BSW Parameter	BSW Type
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF

BSW Description
Reference to the ComMChannel that represents the bus.

Template Description
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.

M2 Parameter

▽

△

BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	
BusMirrorChannel.busMirrorNetworkId	

Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD
BSW Parameter	**BSW Type**

▽

△

MirrorSourceCanMaskBasedIdMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Rule for remapping a set of CAN IDs.	
Template Description	
This element defines a rule for remapping a set of CAN IDs.	
M2 Parameter	
BusMirrorCanIdRangeMapping	
Mapping Rule	**Mapping Type**
Create Container in case that BusMirrorCanIdRangeMapping is aggregated by BusMirrorChannel MappingCan in the role canIdRangeMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00025]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingDestBaseId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.	
Template Description	
Base ID merged with the masked parts of the original CAN ID to form the mapped CAN ID.	
M2 Parameter	
BusMirrorCanIdRangeMapping.destinationBaseId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00028]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingSourceCanIdCode	ECUC-INTEGER-PARAM-DEF
BSW Description	
Value to match masked original CAN IDs.	
Template Description	
Value to match masked original CAN IDs.	
M2 Parameter	
BusMirrorCanIdRangeMapping.sourceCanIdCode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00026]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD/MirrorSourceCanMaskBasedIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanMaskBasedIdMappingSourceCanIdMask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Mask applied to original CAN IDs before comparison.	
Template Description	
Mask applied to original CAN IDs before comparison.	
M2 Parameter	
BusMirrorCanIdRangeMapping.sourceCanIdMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00027]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD
BSW Parameter	**BSW Type**
MirrorSourceCanSingleIdMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Rule for remapping a single CAN ID.	
Template Description	
This element defines a rule for remapping a single CAN ID.	
M2 Parameter	
BusMirrorCanIdToCanIdMapping	
Mapping Rule	**Mapping Type**
Create container in case that the BusMirrorCanIdToCanIdMapping is aggregated by BusMirrorChannelMappingCan in the role canIdToCanIdMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00022]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD/MirrorSourceCanSingleIdMapping
BSW Parameter	**BSW Type**
MirrorSourceCanSingleIdMappingDestCanId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Mapped CAN ID.	
Template Description	
This attribute defines the CanId on the targetChannel.	
M2 Parameter	
BusMirrorCanIdToCanIdMapping.remappedCanId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_Mirror_00024]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkCanFD/MirrorSourceCanSingleIdMapping

BSW Parameter	BSW Type
MirrorSourceCanSingleIdMappingSourceCanId	ECUC-INTEGER-PARAM-DEF

BSW Description
Original CAN ID.

Template Description
This reference points to the sourceFrame with sourceCanId on the sourceChannel.

M2 Parameter
BusMirrorCanIdToCanIdMapping.sourceCanId

Mapping Rule	Mapping Type
Take the value from the identifier attribute of the referenced CanFrameTriggering.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00023]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork

BSW Parameter	BSW Type
MirrorSourceNetworkFlexRay	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description
Source bus representing a FlexRay network.

Template Description
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.

M2 Parameter
BusMirrorChannel.channel

Mapping Rule	Mapping Type
Create a container for each FlexrayPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00042]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay

BSW Parameter	BSW Type
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF

BSW Description
Reference to the ComMChannel that represents the bus.

Template Description
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.

▽

△

M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context	
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkFlexRay	
BSW Parameter		**BSW Type**
MirrorNetworkId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Network ID of the bus.		
Template Description		
This attribute defines the networkId of the communication channel.		
M2 Parameter		
BusMirrorChannel.busMirrorNetworkId		
Mapping Rule		**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Mirror_00012]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork
BSW Parameter	**BSW Type**
MirrorSourceNetworkLin	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Source bus representing a LIN network.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
Create a container for each LinPhysicalChannel which is available in the SystemExtract and is referenced by BusMirrorChannel that is composed by an instance of a BusMirrorChannelMapping in the role sourceChannel.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00029]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	**BSW Type**
MirrorComMNetworkHandleRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the ComMChannel that represents the bus.	
Template Description	
Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel.	
M2 Parameter	
BusMirrorChannel.channel	
Mapping Rule	**Mapping Type**
This reference shall be derived from the: - CanCluster that aggregates the AbstractCanPhysical Channel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCan container - FlexrayCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetwork Flexray container - EthernetPhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkIp container - UserDefinedCluster that aggregates the PhysicalChannel that is referenced by the BusMirrorChannel in case that this reference is used in the MirrorDestNetworkCdd container	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00064]

BSW Module	BSW Context
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	**BSW Type**
MirrorNetworkId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network ID of the bus.	
Template Description	
This attribute defines the networkId of the communication channel.	
M2 Parameter	

▽

△

BusMirrorChannel.busMirrorNetworkId	
Mapping Rule	**Mapping Type**
If this parameter is aggregated by MirrorDestNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingCan. If this parameter is aggregated by MirrorDestNetworkFlexray container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingFlexray. If this parameter is aggregated by MirrorDestNetworkIp container take the value from the BusMirror Channel that is aggregated in the role targetChannel by the instance of BusMirrorChannelMapping Ip. If this parameter is aggregated by MirrorDestNetworkCdd container take the value from the Bus MirrorChannel that is aggregated in the role targetChannel by the instance of BusMirrorChannel MappingUserDefined. If this parameter is aggregated by MirrorSourceNetworkCan container take the value from the Bus MirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirrorChannel Mapping. If this parameter is aggregated by MirrorSourceNetworkFlexray container take the value from the BusMirrorChannel that is aggregated in the role sourceChannel by the instance of BusMirror ChannelMapping. If this parameter is aggregated by MirrorSourceNetworkLin container take the value from the Bus MirrorChannel that is aggregated in the role sourcetChannel by the instance of BusMirrorChannel Mapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00012]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	**BSW Type**
MirrorSourceLinToCanBaseId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Base ID merged with the LIN frame ID to form the CAN ID.	
Template Description	
Base ID merged with the LIN frame ID to form the CAN ID. Only required when a BusMirrorChannel that refers to a LinPhysicalChannel in the role channel is referenced in the role sourceChannel.	
M2 Parameter	
BusMirrorChannelMappingCan.mirrorSourceLinToCanRangeBaseId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00041]

BSW Module	**BSW Context**
Mirror	Mirror/MirrorConfigSet/MirrorSourceNetwork/MirrorSourceNetworkLin
BSW Parameter	**BSW Type**
MirrorSourceLinToCanIdMapping	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Rule for mapping a LIN frame ID to a special CAN ID.	

▽

△

Template Description	
This element defines a rule for remapping a single LIN Frame.	
M2 Parameter	
BusMirrorLinPidToCanIdMapping	
Mapping Rule	**Mapping Type**
Create container in case that the BusMirrorLinPidToCanIdMapping is aggregated by BusMirror ChannelMappingCan in the role linPidToCanIdMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mirror_00038]

C.46 Mka

BSW Module	BSW Context	
Mka	Mka	
BSW Parameter		**BSW Type**
MkaCryptoAlgoConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Cryptography configuration for MACsec.		
Template Description		
This meta-class defines the cryptography configuration for MACsec.		
M2 Parameter		
MacSecCryptoAlgoConfig		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Mka_00021]

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig	
BSW Parameter		**BSW Type**
MkaCipherSuites		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Cipher Suite configuration to use with MACsec. MkaCipherSuitePrio is present in case the MKA instance acts as a Key Server to select the cipher suite to use for MACsec.		
Template Description		
Cipher suite configuration to use with MACsec.		
M2 Parameter		
MacSecCryptoAlgoConfig.cipherSuiteConfig		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Mka_00050]

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig/MkaCipherSuites	
BSW Parameter		**BSW Type**
MkaMacSecCipherSuite		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Cipher Suite to use for MACsec.		
Template Description		
Cipher Suite to use for MACsec.		
M2 Parameter		
MacSecCipherSuiteConfig.cipherSuite		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Mka_00052]

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig/MkaCipherSuites	
BSW Parameter		**BSW Type**
MkaMacSecCipherSuitePrio		ECUC-INTEGER-PARAM-DEF
BSW Description		
In case the MKA instance acts as a Key Server, the priority is used to select the Cipher Suite to use with MACsec from the common supported Ciphers (with the client in the link). Value of 1 means the highest priority. Value of 4 means the lowest priority.		
Template Description		
In case the MKA instance acts as a Key Server, the priority is used to select the Cipher Suite to use with MACsec from the supported Ciphers.		
M2 Parameter		
MacSecCipherSuiteConfig.cipherSuitePriority		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_Mka_00051]

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig	
BSW Parameter		**BSW Type**
MkaMacSecCapability		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
MACsec capability to use for MACsec.		
Template Description		
This attribute defines the MACsec capability.		
M2 Parameter		
MacSecCryptoAlgoConfig.capability		
Mapping Rule		**Mapping Type**
1:1 mapping		full

▽

△

Mapping Status	ECUC Parameter ID
draft	[ECUC_Mka_00025]

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig/MkaMacSecCapability	
BSW Parameter		**BSW Type**
INTEGRITY_AND_CONFIDENTIALITY		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Protect integrity and confidentiality		
Template Description		
Option that ensures confidentiality and integrity		
M2 Parameter		
MacSecCapabilityEnum.intergrityAndConfidentiality		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig/MkaMacSecCapability	
BSW Parameter		**BSW Type**
INTEGRITY_WITHOUT_CONFIDENTIALITY		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Only protect integrity, do not protect confidentiality		
Template Description		
Option that ensures integrity without confidentiality		
M2 Parameter		
MacSecCapabilityEnum.intergrityWithoutConfidentiality		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		

BSW Module	BSW Context	
Mka	Mka/MkaCryptoAlgoConfig	
BSW Parameter		**BSW Type**
MkaMacSecConfidentialityOffset		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
The confidentiality Offset is only applicable if “Integrity and confidentiality” with a non-XPN cipher suite is selected.		
Template Description		
The MACsec confidentiality offset specifies the number of bytes starting from the frame header. MACsec encrypts only the bytes after the offset in a frame.		
M2 Parameter		
MacSecCryptoAlgoConfig.confidentialityOffset		

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00026]

BSW Module	BSW Context
Mka	Mka/MkaCryptoAlgoConfig/MkaMacSecConfidentialityOffset
BSW Parameter	**BSW Type**
CONFIDENTIALITY_OFFSET_0	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Start confidentiality protection at offset 0 (complete user data is encrypted)	
Template Description	
confidentiality offset of 0.	
M2 Parameter	
MacSecConfidentialityOffsetEnum.ConfidentialityOffset_0	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaCryptoAlgoConfig/MkaMacSecConfidentialityOffset
BSW Parameter	**BSW Type**
CONFIDENTIALITY_OFFSET_50	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Start confidentiality protection at octet 50 (first 49 octets are not encrypted)	
Template Description	
confidentiality offset of 50.	
M2 Parameter	
MacSecConfidentialityOffsetEnum.ConfidentialityOffset_50	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaCryptoAlgoConfig
BSW Parameter	**BSW Type**
MkaMacSecReplayProtection	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
MACsec replay protection parameter for MACsec.\newline The Replay Protection parameter is defined in the IEEE 802.1 AE-2018 document, on chapter 10.4. It enables the replay protection if a packet is received with PacketNumber outside of the Window = PN - ProtectionWindow.	
Template Description	

▽

△

This attribute is used to configure the MACsec replay protection.	
M2 Parameter	
MacSecCryptoAlgoConfig.replayProtection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00027]

BSW Module	**BSW Context**
Mka	Mka/MkaCryptoAlgoConfig
BSW Parameter	**BSW Type**
MkaMacSecReplayProtectionWindow	ECUC-INTEGER-PARAM-DEF
BSW Description	
In case replay protection is active, replay protection window.The Protection Window is a positive integer between 0 and 2^32-1 (No XPN) or 2^30-1 (XPN).	
Template Description	
In case replay protection is active, this attribute defines the replay protection window.	
M2 Parameter	
MacSecCryptoAlgoConfig.replayProtectionWindow	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00028]

BSW Module	**BSW Context**
Mka	Mka/MkaPaeConfiguration
BSW Parameter	**BSW Type**
MkaAutoStart	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Autostart or manual start of the PAE Instance. True := Autostart False := Manual Start If Autostart = False, the method Mka_StartPae is used to start the PAE instance.	
Template Description	
This attribute defines how the Port Access Entity (PAE) is started: • true := Autostart • false := Manual Start	
M2 Parameter	
MacSecProps.autoStart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00012]

BSW Module	BSW Context
Mka	Mka/MkaPaeConfiguration
BSW Parameter	**BSW Type**
MkaCalculateSakHashKey	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Indicates if Mka shall calculate and provide SAK hash key. • True = Mka shall calculate SAK hash key and provide it with HashKeyPtr of Mka_SakKeyPtrType. • False = Mka shall not calculate SAK hash key, HashKeyPtr of Mka_SakKeyPtrType shall be set to NULL pointer. (In this case SAK hash key is calculated by SecY itself.)	
Template Description	
This attribute defines how the Port Access Entity (PAE) is started: • true := Autostart • false := Manual Start	
M2 Parameter	
MacSecProps.autoStart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Mka_00077]

BSW Module	BSW Context
Mka	Mka/MkaPaeConfiguration
BSW Parameter	**BSW Type**
MkaSakRekeyTimeSpan	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time [s] to trigger the rekey of an in use SAK. If set to 0, the rekey will not be triggered after a time span.	
Template Description	
Time in seconds to trigger the rekey of an in use SAK (Static Secure Association key). If set to 0, the rekey will not be triggered after a time span.	
M2 Parameter	
MacSecProps.sakRekeyTimeSpan	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00024]

BSW Module	BSW Context
Mka	Mka
BSW Parameter	**BSW Type**
MkaPaeInstance	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
MKA configuration of a controlled port.	
Template Description	
This meta-class allows to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY).	
M2 Parameter	

▽

△

MacSecProps	
Mapping Rule	**Mapping Type**
Create container for each existing MacSecProps element.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00003]

BSW Module	**BSW Context**
Mka	Mka/MkaPaeInstance
BSW Parameter	**BSW Type**
MkaEthIfControllerRef	ECUC-REFERENCE-DEF
BSW Description	
A reference to the EthIfController is used to identify the MACsec protected Ethernet hardware entity either as PHY or as an Ethernet controller.	
Template Description	
M2 Parameter	
EthernetCommunicationController.couplingPort	
Mapping Rule	**Mapping Type**
Derive container from EthernetCommunicationController that aggregates the CouplingPort that in turn aggregates MkaProps that corresponds to MkaPaeInstance.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00013]

BSW Module	**BSW Context**
Mka	Mka/MkaPaeInstance
BSW Parameter	**BSW Type**
MkaKay	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
MKA instance (KaY) for a controlled port (PaE).	
Template Description	
Properties to configure the MKA instance (KaY) for a controlled CouplingPort (PaE).	
M2 Parameter	
MacSecProps.macSecKayConfig	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00017]

BSW Module	**BSW Context**
Mka	Mka/MkaPaeInstance/MkaKay
BSW Parameter	**BSW Type**
MkaDstMacAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Destination MAC address to use by the MKA instance. The destination MAC addresses to use are defined in the IEEE 802.1 X-2020 chapter 11.1.1 (Table 11-1).	

▽

△

Template Description	
This attribute defines the destination MAC Address that is used to calculate the ICV (Integrity Check Value).	
M2 Parameter	
MacSecLocalKayProps.destinationMacAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00032]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay
BSW Parameter	**BSW Type**
MkaKayParticipant	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
MKA participant configuration.	
Template Description	
Reference to MKA participant settings supported on the CouplingPort.	
M2 Parameter	
MacSecLocalKayProps.mkaParticipant	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00038]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay/MkaKayParticipant
BSW Parameter	**BSW Type**
MkaCryptoAlgoRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the cryptography to use (MkaAlgoConfiguration Container).	
Template Description	
Cryptography that is used by the MKA Participant.	
M2 Parameter	
MacSecKayParticipant.cryptoAlgoConfig	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00048]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay
BSW Parameter	**BSW Type**
MkaKeyServerPriority	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Key Server Priority of the MKA participants. In case it is not provided, the default value is 0 for an MKA_KEY_SERVER and 255 for an MKA_PEER.	
Template Description	
This attribute defines the key-server priority.	
M2 Parameter	
MacSecLocalKayProps.keyServerPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00022]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay
BSW Parameter	**BSW Type**
MkaRole	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Role of the MKA instance.	
Template Description	
Role of the MAC Security Key Agreement Entity	
M2 Parameter	
MacSecLocalKayProps.role	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00029]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay/MkaRole
BSW Parameter	**BSW Type**
MKA_KEY_SERVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Act as an MKA key server	
Template Description	
Port acts in the KeyServer role	
M2 Parameter	
MacSecRoleEnum.keyServer	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaKay/MkaRole
BSW Parameter	**BSW Type**
MKA_PEER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Act as an MKA peer	
Template Description	
Port acts in the peer role	
M2 Parameter	
MacSecRoleEnum.peer	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance
BSW Parameter	**BSW Type**
MkaOnFailPermissiveMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Sets the behavior of the PAE in case MKA does not succeed when MKA is enabled.	
Template Description	
This attribute sets the behavior of the Port Access Entity in case MACsec does not succeed.	
M2 Parameter	
MacSecProps.onFailPermissiveMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00018]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaOnFailPermissiveMode
BSW Parameter	**BSW Type**
MKA_PERMISSIVE_MODE_NEVER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The controlled port will never be set to enabled if the participants cannot establish and successfully use a MACsec Secure Channel.	
Template Description	
The controlled port will never be set to enabled if the participants cannot establish and successfully use a MACsec Secure Channel.	
M2 Parameter	
MacSecFailPermissiveModeEnum.never	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance/MkaOnFailPermissiveMode
BSW Parameter	**BSW Type**
MKA_PERMISSIVE_MODE_TIMEOUT	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The controlled port will be set to enabled and MACsec will not be used in the referred port if the timeout value (MkaOnFail PermissiveModeTimeout) is reached and none MKA instance under the PAE instance could success the following conditions: - A participant belonging to the same CA was recognized and authenticated. - A secure channel could be established. - Both participants can transmit and receive MACsec protected traffic through the SC.	
Template Description	
The controlled port will be set to enabled and MACsec will not be used in the port if the timeout value (onFailPermissiveMode Timeout) is reached and the following conditions apply: - A participant belonging to the same CA was recognized and authenticated. - A secure channel could be established. - Both participants can transmit and receive MACsec protected traffic through the SC.	
M2 Parameter	
MacSecFailPermissiveModeEnum.timeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance
BSW Parameter	**BSW Type**
MkaOnFailPermissiveModeTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in seconds to enable the controlled port in case MkaOnFailPermissiveMode is set to Timeout.	
Template Description	
Timeout in seconds to enable the controlled port in case onFailPermissiveMode is set to Timeout.	
M2 Parameter	
MacSecProps.onFailPermissiveModeTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00019]

BSW Module	BSW Context
Mka	Mka/MkaPaeInstance
BSW Parameter	**BSW Type**
MkaSwitchPortRef	ECUC-REFERENCE-DEF
BSW Description	
A reference to the EthSwtPort is used to identity the MACsec protected Ethernet hardware entity as Ethernet switch port.	
Template Description	
M2 Parameter	
CouplingElement.couplingPort	

▽

△

Mapping Rule	Mapping Type
Derive container from CouplingElement (SWITCH) that aggregates the CouplingPort that in turn aggregates MkaProps that corresponds to MkaPaeInstance.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Mka_00014]

C.47 Nm

BSW Module	BSW Context	
Nm	Nm	
BSW Parameter		**BSW Type**
NmChannelConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration (parameters) of the bus channel(s). The channel parameter shall be harmonized within the whole communication stack.		
Template Description		
Set of NM nodes coordinated with use of the NM algorithm.		
M2 Parameter		
NmCluster		
Mapping Rule		**Mapping Type**
Create Container for each existing NmCluster.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00197]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmActiveCoordinator		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This parameter indicates whether a NM channel - part of a Nm Coordination cluster - will be coordinated actively (NmActive Coordinator = TRUE) or passively (NmActiveCoordinator = FALSE).		
Template Description		
This attribute indicates the role the NM Coordinator will have on this channel.		
M2 Parameter		
NmNode.nmCoordinatorRole		
Mapping Rule		**Mapping Type**
If nmCoordinatorRole is set to Active then NmActiveCoordinator shall be present and set to true. If nmCoordinatorRole is set to Passive then NmActiveCoordinator shall be present and set to false.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00236]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmBusType		ECUC-CHOICE-CONTAINER-DEF
BSW Description		
Choose the NmBusType		
Template Description		
CanNmCluster: Can specific NmCluster attributes **FlexrayNmCluster:** FlexRay specific NM cluster attributes. **UdpNmCluster:** Udp specific NmCluster attributes		
M2 Parameter		
CanNmCluster, FlexrayNmCluster, UdpNmCluster		
Mapping Rule		**Mapping Type**
Bus Type can be derived from the BusNm Configuration in the System Description.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00218]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmChannelSleepMaster		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision. If this parameter is set to TRUE, the Nm shall assume that the channel is always ready to go to sleep and that no calls to Nm_RemoteSleepIndication or Nm_RemoteSleepCancellation will be made from the <Bus>Nm representing this channel. If this parameter is set to FALSE, the Nm shall not assume that the network is ready to sleep until a call has been made to Nm_RemoteSleepCancellation.		
Template Description		
This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision.		
M2 Parameter		
NmCluster.nmChannelSleepMaster		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00227]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmComUserDataSupport		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This parameter indicates whether on a NM channel user data is accessed via Com signals or by SetUserData API.		



△

Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
If an NmPdu is transmitted by the affected Ecu and if the NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping (and is consequently handled via the PduR and Com) then NmComUserDataSupport shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00241]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig
BSW Parameter	**BSW Type**
NmCoordClusterIndex	ECUC-INTEGER-PARAM-DEF
BSW Description	
If this parameter is undefined for a channel, the corresponding bus does not belong to an NM coordination cluster.	
Template Description	
NmCoordinationCluster identification number.	
M2 Parameter	
NmNode.nmCoordCluster	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00221]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig
BSW Parameter	**BSW Type**
NmDynamicPncToChannelMappingEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
1:1 Mapping. If M2 Parameter not defined then do not create NmDynamicPncToChannelMapping Enabled.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Nm_00248]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmPassiveModeEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This parameter indicates whether a NM channel is active,e.g. can request communication and keep the bus awake, or passive, e.g. can just be woken up and kept awake by other ECUs.		
Template Description		
Enables support of the Passive Mode. The passive mode is configurable per channel.		
M2 Parameter		
NmNode.nmPassiveModeEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00242]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmPnEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
If this parameter is true, then this NM channel supports Partial Networking.		
Template Description		
Defines whether this NmCluster contributes to the partial network mechanism.		
M2 Parameter		
NmCluster.nmPncParticipation		
Mapping Rule		**Mapping Type**
Set to true if the NmCluster has nmPncParticipation set to true.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00254]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmPnEraCalcEnabled		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Specifies if NmIf calculates the PN request information for external requests. (ERA)		
Template Description		
Defines whether this NmCluster contributes to the partial network mechanism.		
M2 Parameter		
NmCluster.nmPncParticipation		
Mapping Rule		**Mapping Type**
Set to true if at least one NmCluster has nmPncParticipation set to true and pncGatewayType is not NONE. Otherwise set to FALSE.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00259]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmPnFilterMaskByte		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Information for the filter of the PNC Request Bit Vector.		
Template Description		
Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.		
M2 Parameter		
CommunicationConnector.pncFilterArrayMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00255]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmPnFilterMaskByte	
BSW Parameter		**BSW Type**
NmPnFilterMaskByteIndex		ECUC-INTEGER-PARAM-DEF
BSW Description		
Index of the filter mask byte. Specifies the position within the filter mask byte array.		
Template Description		
Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.		
M2 Parameter		
CommunicationConnector.pncFilterArrayMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00256]

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmPnFilterMaskByte	
BSW Parameter		**BSW Type**
NmPnFilterMaskByteValue		ECUC-INTEGER-PARAM-DEF
BSW Description		
Parameter to configure the filter mask byte.		
Template Description		
Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.		
M2 Parameter		
CommunicationConnector.pncFilterArrayMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Nm_00257]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig

BSW Parameter	BSW Type
NmPncBitVectorLength	ECUC-INTEGER-PARAM-DEF

BSW Description

Parameter to configure the length of the PNC bit request information in bytes, which is transmitted within NM PDU by the corresponding <Bus>Nm.

Template Description

System.pncVectorLength:
Length of the partial networking request release information vector (in bytes).

NmCluster.pncClusterVectorLength:
Optionally defines the length of the PNC Request Bit Vector per CommunicationCluster (and VLAN in case of UdpNm). If not defined then System.pncVectorLength applies.

Should only make the PNC Request Bit Vector shorter (or same length as defined in System.pncVectorLength).

M2 Parameter

`System.pncVectorLength`, `NmCluster.pncClusterVectorLength`

Mapping Rule	Mapping Type
If NmCluster.pncClusterVectorLength is defined then the value is taken from NmCluster.pncClusterVectorLength, otherwise the value is taken from System.pncVectorLength.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00258]

BSW Module	BSW Context
Nm	Nm/NmChannelConfig

BSW Parameter	BSW Type
NmSynchronizingNetwork	ECUC-BOOLEAN-PARAM-DEF

BSW Description

If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.

Template Description

If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_SynchronizationPoint() at regular intervals.

M2 Parameter

`NmCluster.nmSynchronizingNetwork`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00223]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures

BSW Parameter	BSW Type
NmBusSynchronizationEnabled	ECUC-BOOLEAN-PARAM-DEF

BSW Description

Pre-processor switch for enabling bus synchronization support of the <Bus>Nms. This feature is required for NM Coordinator nodes only.

Template Description

▽

△

Enables bus synchronization support.	
M2 Parameter	
NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00208]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmComControlEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
NmEcu.nmComControlEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00210]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmCoordinatorSupportEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling NM Coordinator support.	
Template Description	
A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously.	
M2 Parameter	
NmCoordinator	
Mapping Rule	**Mapping Type**
If NmCoordinators are defined set this parameter to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00206]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmCoordinatorSyncSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	

▽

△

Enables/disables the coordinator synchronisation support.	
Template Description	
Switch for enabling NmCoordinatorSync (coordination of nested busses) support.	
M2 Parameter	
NmCoordinator.nmCoordSyncSupport	
Mapping Rule	**Mapping Type**
If NmCoordinator is present then the value of NmCoordinatorSyncSupport shall be set to the value of nmCoordSyncSupport.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00240]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmDynamicPncToChannelMappingSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
If at least one dynamicPncToChannelMappingEnabled attribute is defined and if at least one CommunicationConnector of the EcuInstance has dynamicPncToChannelMappingEnabled set to true, then NmDynamicPncToChannelMappingSupport shall be set to true. Otherwise NmDynamic PncToChannelMappingSupport shall be set to false	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Nm_00246]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmGlobalCoordinatorTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter defines the maximum shutdown time of a connected and coordinated NM-Cluster. Note:This includes nested connections.	
Template Description	
This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.	
M2 Parameter	
NmCoordinator.nmGlobalCoordinatorTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00237]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmPartialNetworkSupportEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the Nm Partial Network support.	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
NmCluster.nmPncParticipation	
Mapping Rule	**Mapping Type**
Set to true if at least one NmCluster has nmPncParticipation set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00253]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmPduRxIndicationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the PDU Rx Indication.	
Template Description	
Switch for enabling the PDU Rx Indication.	
M2 Parameter	
NmEcu.nmPduRxIndicationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00214]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmRemoteSleepIndEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling Remote Sleep Indication support. This feature is required for a Gateway or Nm Coordinator functionality. Note that this feature should not be used if all NM channels have Passive Mode enabled.	
Template Description	
Switch for enabling remote sleep indication support.	
M2 Parameter	
NmEcu.nmRemoteSleepIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00207]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmStateChangeIndEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the Network Management state change notification.	
Template Description	
Enables the CAN Network Management state change notification.	
M2 Parameter	
NmEcu.nmStateChangeIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00215]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalFeatures
BSW Parameter	**BSW Type**
NmUserDataEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling User Data support.	
Template Description	
Switch for enabling user data support.	
M2 Parameter	
NmEcu.nmUserDataEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00211]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalProperties
BSW Parameter	**BSW Type**
NmCycletimeMainFunction	ECUC-FLOAT-PARAM-DEF
BSW Description	
The period between successive calls to the Main Function of the NM Interface in seconds.	
Template Description	
The period between successive calls to the Main Function of the NM Interface in seconds.	
M2 Parameter	
NmEcu.nmCycletimeMainFunction	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00205]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalProperties
BSW Parameter	**BSW Type**
NmPnEiraCalcEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if NmIf calculates the PNC request information for internal and external requests (EIRA) true: PN request are calculated false: PN request are not calculated Note: A PNC coordinator (NmPnEraCalcEnabled set to TRUE) has always set NmPnEiraCalcEnabled to TRUE.	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
NmCluster.nmPncParticipation	
Mapping Rule	**Mapping Type**
Set to true if at least one NmCluster has nmPncParticipation set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00251]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalProperties
BSW Parameter	**BSW Type**
NmPnResetTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the runtime of the reset time in seconds. This reset time is valid for the reset of PNC requests in the EIRA and in the ERA. The value shall be the same for every channel. Thus it is a global config parameter.	
Template Description	
Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.	
M2 Parameter	
EcuInstance.pnResetTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00250]

BSW Module	BSW Context
Nm	Nm/NmGlobalConfig/NmGlobalProperties
BSW Parameter	**BSW Type**
NmPnShutdownMessageRetransmissionDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the duration in seconds of the retransmission phase of a PN shutdown message. A retransmission shall be performed per affected NM channel, as long as the PN shutdown message could not be successfully sent and the retransmission timer is running. The value shall be a multiple integral NmMainFunctionPeriod.	
Template Description	
The period between successive calls to the Main Function of the NM Interface in seconds.	
M2 Parameter	

△

NmEcu.nmCycletimeMainFunction	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00260]

BSW Module	**BSW Context**
Nm	Nm/NmGlobalConfig/NmGlobalProperties
BSW Parameter	**BSW Type**
NmPncBitVectorOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
Parameter to configure the offset in bytes of the PNC Request Bit Vector that contains the PNC requests, which is transmitted within NM PDU by the corresponding <Bus>Nm.	
Template Description	
Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.	
M2 Parameter	
System.pncVectorOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Nm_00252]

C.48 Os

BSW Module	**BSW Context**
Os	Os
BSW Parameter	**BSW Type**
OsTask	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents an ISO 17356 task.	
Template Description	
This meta-class represents a proxy for an OsTask in the System Description.	
M2 Parameter	
OsTaskProxy	
Mapping Rule	**Mapping Type**
The integrator is allowed to combine several OsTaskProxies into one OsTask.	partial
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Os_00073]

BSW Module	BSW Context
Os	Os/OsTask
BSW Parameter	**BSW Type**
OsTaskPeriod	ECUC-FLOAT-PARAM-DEF
BSW Description	
This parameter specifies the period in seconds of this task in case of a cyclically activated task. If this parameter is not given the task can be activated sporadicly or cyclically with a unknown period value. This value is information, e.g. for time base calculations in the RTE in case TimingEvents are mapped onto this OsTask.Be aware, that this parameter is not supposed to be relevant for the OS! This information is given as part of the OS configuration to support configuration work flows using a fixed set of OsTasks.	
Template Description	
This attribute specifies the period in seconds of this task in case of a cyclically activated task. Please note that this attribute is informative and not directly relevant for the AUTOSAR OS. But the attribute value can be mapped into the OS configuration to support configuration work flows using a fixed set of OsTasks.	
M2 Parameter	
`OsTaskProxy.period`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Os_00404]

BSW Module	BSW Context
Os	Os/OsTask
BSW Parameter	**BSW Type**
OsTaskPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
The priority of a task is defined by the value of this attribute. This value has to be understood as a relative value, i.e. the values show only the relative ordering of the tasks. ISO 17356-3 defines the lowest priority as zero (0); larger values correspond to higher priorities.	
Template Description	
This attribute defines the priority of a task as a relative value, i.e. the values show only the relative ordering of the tasks.	
M2 Parameter	
`OsTaskProxy.priority`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Os_00075]

BSW Module	BSW Context
Os	Os/OsTask
BSW Parameter	**BSW Type**
OsTaskSchedule	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The OsTaskSchedule attribute defines the preemptability of the task. If this attribute is set to NON, no internal resources may be assigned to this task.	
Template Description	



△

This attribute defines the preemptability of the task.	
M2 Parameter	
OsTaskProxy.preemptability	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Os_00076]

C.49 PduR

BSW Module	**BSW Context**
PduR	PduR/PduRRoutingPaths
BSW Parameter	**BSW Type**
PduRRoutingPath	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is a subcontainer of PduRRoutingTable and specifies the routing path of a PDU.	
Template Description	
IPduMapping: Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. **PduTriggering:** The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu. Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface. If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface. **TpConfig:** Contains all configuration elements for AUTOSAR TP.	
M2 Parameter	
IPduMapping, PduTriggering, TpConfig	
Mapping Rule	**Mapping Type**
For each MultiplatformGateway.pduMapping; for each SignalPdu-MultiplexedPdu Connection; for each IPduTriggering; for each TpConfig create one PduRRoutingPath.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00248]

BSW Module	**BSW Context**
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	**BSW Type**
PduRDefaultValue	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Specifies the default value of the I-PDU. Only required for gateway operation and if at least one PDU specified by PduRDest Pdu uses TriggerTransmit Data provision. Represented as an array of IntegerParamDef.	
Template Description	
M2 Parameter	
TargetIPduRef.defaultValue	
Mapping Rule	**Mapping Type**
Container should be created if PduMappingDefaulValue is described in the Sys-T	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00299]

BSW Module	**BSW Context**
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue
BSW Parameter	**BSW Type**
PduRDefaultValueElement	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Each value element is represented by the element and the position in an array.	
Template Description	
The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.	
M2 Parameter	
DefaultValueElement	
Mapping Rule	**Mapping Type**
Container shall be created for each DefaultValueElement that is aggregated by PduMapping DefaultValue.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00300]

BSW Module	**BSW Context**
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue/PduRDefaultValueElement
BSW Parameter	**BSW Type**
PduRDefaultValueElement	ECUC-INTEGER-PARAM-DEF
BSW Description	
The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength. The position of this parameter in the container is specified by the PduRElementBytePosition parameter.	
Template Description	
The integer value of a freely defined data byte.	
M2 Parameter	
DefaultValueElement.elementByteValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00290]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath/PduRDefaultValue/PduRDefaultValueElement
BSW Parameter	**BSW Type**
PduRDefaultValueElementBytePosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the byte position of the element within the default value	
Template Description	
This attribute specifies the byte position of the element within the default value	
M2 Parameter	
DefaultValueElement.elementPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00292]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	**BSW Type**
PduRRoutingPathGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to routing paths.	
Template Description	
M2 Parameter	
PdurIPduGroup.iPdu	
Mapping Rule	**Mapping Type**
If the PduTriggering this PduRRoutingPath is derived from is the target of a PdurIPduGroup.iPdu reference then a PduRRoutingPathGroupRef shall be created at the PduRRoutingPath and the respective PduRRoutingPathGroup shall be referenced from the PduRRoutingPathGroupRef.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00352]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths/PduRRoutingPath
BSW Parameter	**BSW Type**
PduRTpThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter is only relevant for TP routings. When configured, it enables on-the-fly routing and defines the number of bytes which must have been received before transmission on the destination bus may start. When omitted, direct TP routing is enforced. The PduRouter shall ensure that a buffer is allocated for this routing path which is at least as large as the threshold.	
Template Description	
Optionally defines the to be configured Pdu Router TpChunkSize for this routing relation.	
M2 Parameter	
IPduMapping.pdurTpChunkSize	

▽

△

Mapping Rule	Mapping Type
PduRTpThreshold shall only be configured by IpduMapping.pduRTpChunkSize if the following conditions hold: (1) The routing path uses the "Tp" API (2) The routing path only contains one single destination routing path.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00320]

BSW Module	BSW Context
PduR	PduR/PduRRoutingPaths
BSW Parameter	**BSW Type**
PduRRoutingPathGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container groups routing paths. By this grouping, it is possible to switch all routings related to one network, or to one kind of PDUs. PduRRoutingPaths link one source with one destination. Enabling and disabling of routing path groups is done using the PduR API.	
Template Description	
The AUTOSAR PduR will enable and disable the sending of configurable groups of IPdus during runtime according to the AUTOSAR PduR specification.	
M2 Parameter	
PdurIPduGroup	
Mapping Rule	**Mapping Type**
Create container for each existing PduRIPduGroup that is connected to the regarded Ecu	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_PduR_00308]

C.50 Rte

BSW Module	BSW Context
Rte	Rte/RteComDataMapping
BSW Parameter	**BSW Type**
RteComDataE2EXfNNDExtensionFactor	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configuring the jitter factor of no new data reception cycle time.	
Template Description	
This attribute defines the jitter factor of E2E no new data reception cycle time.	
M2 Parameter	
CyclicHandlingComDataToOsTaskProxyMapping.e2eNoNewDataExtensionFactor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Rte_09235]

BSW Module	BSW Context	
Rte	Rte/RteComDataMapping	
BSW Parameter		**BSW Type**
RteComDataE2EXfOverSamplingPeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
The Oversampling Cycle Time for the cyclic processing of the mapped VariableDataPrototypes.		
Template Description		
This attribute defines an optional E2E oversampling cycle time for the cyclic processing of the mapped VariableData Prototypes.		
M2 Parameter		
CyclicHandlingComDataToOsTaskProxyMapping.e2eOversamplingPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Rte_09234]

BSW Module	BSW Context	
Rte	Rte/RteComDataMapping	
BSW Parameter		**BSW Type**
RteComDataMainFunctionOsTaskRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to the OsTask in which the ComSignals or ComSignalGroups shall be read cyclically.		
Template Description		
Reference to the OsTaskProxy.		
M2 Parameter		
CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy		
Mapping Rule		**Mapping Type**
The OsTaskProxy might have been taken over 1:1 into the EcuC, then the reference can be defined based on the OsTask derivation.		partial
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Rte_09231]

BSW Module	BSW Context	
Rte	Rte/RteComDataMapping	
BSW Parameter		**BSW Type**
RteComDataMainFunctionPeriod		ECUC-FLOAT-PARAM-DEF
BSW Description		
Period of the Com_MainFunction		
Template Description		
This attribute defines the cyclic processing time within the OsTask.		
M2 Parameter		
CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Rte_09233]

BSW Module	BSW Context	
Rte	Rte/RteComDataMapping	
BSW Parameter		**BSW Type**
RteComDataMainFunctionPositionInTask		ECUC-INTEGER-PARAM-DEF
BSW Description		
Each set of VariableDataPrototypes mapped to an OsTask has a specific position within the task execution. If more than one VariableDataPrototype is mapped, then the handling of these VariableDataPrototypes is executed in an arbitrary order.		
Template Description		
This attribute is used to describe the position in the OsTask as a relative value.		
M2 Parameter		
CyclicHandlingComDataToOsTaskProxyMapping.offset		
Mapping Rule		**Mapping Type**
offset defines a relative position, there may be other activities mapped to the same OsTaks and thus the RteComDataMainFunctionPositionInTask needs to be adopted accordingly.		partial
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Rte_09232]

BSW Module	BSW Context	
Rte	Rte/RteComDataMapping	
BSW Parameter		**BSW Type**
RteComDataVariableDataPrototypeInstanceRef		ECUC-INSTANCE-REFERENCE-DEF
BSW Description		
Reference to the set of VariableDataPrototypes which are mapped to ComSignals or ComSignalGroups that will be read cyclically.		
Template Description		
This meta-class is used to map VariableDataPrototypes to an OsTaskProxy for the Cyclic Handling of Communication Data in the RTE.		
M2 Parameter		
CyclicHandlingComDataToOsTaskProxyMapping		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Rte_09230]

C.51 Sd

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdClientService		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container specifies all parameters used by Client services.		
Template Description		

▽

△

Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a Communication Connector.	
M2 Parameter	
ConsumedServiceInstance	
Mapping Rule	**Mapping Type**
Create container for each existing ConsumedServiceInstance that is available in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00005]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdBlocklistedVersions	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Collection of blocklisted versions.	
Template Description	
Collection of blocklisted versions	
M2 Parameter	
ConsumedServiceInstance.blocklistedVersion	
Mapping Rule	**Mapping Type**
If at least one ConsumedServiceInstance.blocklistedVersion exists the container shall be derived.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_SD_00141]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdBlocklistedVersions
BSW Parameter	**BSW Type**
SdBlocklistedMinorVersions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Blocklisted MinorVersions.	
Template Description	
Minor Version of the ServiceInterface.	
M2 Parameter	
SomeipServiceVersion.minorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_SD_00142]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientCapabilityRecord	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" -- password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")	
Template Description	
A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter	
`AbstractServiceInstance.capabilityRecord`	
Mapping Rule	**Mapping Type**
1:1 mapping to ConsumedServiceInstance.capabilityRecord.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00072]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord
BSW Parameter	**BSW Type**
SdClientServiceCapabilityRecordKey	ECUC-STRING-PARAM-DEF
BSW Description	
Defines a CapabilityRecord key.	
Template Description	
Defines a key.	
M2 Parameter	
`TagWithOptionalValue.key`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00073]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord
BSW Parameter	**BSW Type**
SdClientServiceCapabilityRecordValue	ECUC-STRING-PARAM-DEF
BSW Description	
Defines the corresponding CapabilityRecord value.	
Template Description	
Defines the corresponding value.	
M2 Parameter	
`TagWithOptionalValue.value`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00074]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceAllowedProvider	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The container defines the allowed providers for this ClientService.	
Template Description	
NetworkEndpoint on which the ProvidedServiceInstance that is communicating with this ConsumedServiceInstance is allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established.	
M2 Parameter	
ConsumedServiceInstance.allowedServiceProvider	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00147]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientServiceAllowedProvider
BSW Parameter	**BSW Type**
SdIpAddress	ECUC-STRING-PARAM-DEF
BSW Description	
This parameter defines the IP Address of the remote communication partner.	
Template Description	
M2 Parameter	
ApplicationEndpoint.networkEndpoint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00148]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientServiceAllowedProvider
BSW Parameter	**BSW Type**
SdIpAddressType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the IP version that is used for communication with the remote communication partner.	
Template Description	
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).	
M2 Parameter	
NetworkEndpoint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00149]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceAutoRequire	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If existing and set to true, this Service will be set to "required" on start.	
Template Description	
Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.	
M2 Parameter	
ConsumedServiceInstance.autoRequire	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00143]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Id to identify the service. This is unique for the service interface.	
Template Description	
This attribute represents the ability to describe the SOME/IP service ID that is searched.	
M2 Parameter	
ConsumedServiceInstance.serviceIdentifier	
Mapping Rule	**Mapping Type**
Shall be derived from ConsumedServiceInstance.serviceIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00020]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configuration parameter to specify Instance Id of the service as used in SD entries.	
Template Description	
This attribute represents the ability to describe the required service instance ID.	
M2 Parameter	
ConsumedServiceInstance.instanceIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00022]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceMajorVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Major version number of the Service as used in the SD entries.	
Template Description	
Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.	
M2 Parameter	
AbstractServiceInstance.majorVersion	
Mapping Rule	**Mapping Type**
Shall be derived from ConsumedServiceInstance.majorVersion	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00070]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceMinorVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Minor version number of the Service as used in the SD Service Entries. If configured to 0xffffffff (any), SD will accept all Minor Versions.	
Template Description	
Minor Version of the ServiceInterface. Value can be set to a number that represents the Minor Version of the searched service or to ANY.	
M2 Parameter	
ConsumedServiceInstance.minorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00071]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceMulticastRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SoAdSocketConnection representing the data path (UDP) for communication with the server. This element is also used to set the remote address of the server. This is used, if a ClientService subscribes with a Consumed Eventgroup multicast endpoint. This is an alternative to subscribe with a Consumed Eventgroup unicast endpoint (see SdClientServiceUdpRef). Please note: usage of this reference is mutually exclusive to SdClientServiceUdpRef.	
Template Description	
Multicast Address that is used by the client to subscribe to the server: This enables the multicast subscription feature.	
M2 Parameter	
ConsumedServiceInstance.eventMulticastSubscriptionAddress	



△

Mapping Rule	Mapping Type
This reference shall be derived from the ConsumedServiceInstance.eventMulticastSubscription Address reference to an ApplicationEndpoint that defines a multicast endpoint. If the Consumed ServiceInstance does not reference a multicast endpoint with the eventMulticastSubscription Address reference the SdClientServiceMulticastRef shall be skipped.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Sd_00145]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceTcpRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SoAdSocketConnection representing the data path (TCP) for communication with methods. This element is also used to set the remote address of the server and to open the TCP connection.	
Template Description	
The local address over which the CSI is consumed (udp, tcp or both).	
M2 Parameter	
ConsumedServiceInstance.localUnicastAddress	

Mapping Rule	Mapping Type
This reference shall be derived from the ConsumedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a TCP Port. If the ConsumedServiceInstance does not reference a TCP Port with the localUnicastAddress reference the SdServerServiceTcp Ref shall be skipped.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00100]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdClientServiceTimerRef	ECUC-REFERENCE-DEF
BSW Description	
The reference of the SdClientTimer container for this service.	
Template Description	
Client specific configuration settings relevant for the SOME/IP service discovery.	
M2 Parameter	
ConsumedServiceInstance.sdClientTimerConfig	

Mapping Rule	Mapping Type
The reference to the SdClientTimer shall be created pointing to the SdClientTimer container which was created based on the ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00103]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**

▽

△

SdClientServiceUdpRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SoAdSocketConnection representing the data path (UDP) for communication with methods. This element is also used to set the remote address of the server. This is used, if a ClientService subscribes with a Consumed Eventgroup unicast endpoint. This is an alternative to subscribe with a Consumed Eventgroup multicast endpoint. (see SdClientServiceMulticastRef). Please note: usage of this reference is mutually exclusive to SdClientServiceMulticastRef.	
Template Description	
The local address over which the CSI is consumed (udp, tcp or both).	
M2 Parameter	
ConsumedServiceInstance.localUnicastAddress	
Mapping Rule	**Mapping Type**
This reference shall be derived from the ConsumedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port. If the ConsumedServiceInstance does not reference a UDP Port with the localUnicastAddress reference the SdServerServiceUdp Ref shall be skipped.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00101]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdConsumedEventGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
A Service may have event groups which can be consumed. A service consumer has to subscribe to the corresponding event-group. After the subscription the event consumer takes the role of a server and the event provider that of a client.	
Template Description	
This element represents an event-group to which the service consumer wants to subscribe.	
M2 Parameter	
ConsumedEventGroup	
Mapping Rule	**Mapping Type**
Create container for every existing consumedEventGroup that is aggregated by the Consumed ServiceInstance	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00056]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupAutoRequire	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If existing and set to true, this EventGroup will be set to "required" on start.	
Template Description	
Defines that this ConsumedEventGroup shall be requested (subscribed) as soon as the corresponding ConsumedService Instance is requested. This could be at ECU start, if ConsumedServiceInstance.autoRequire is set to TRUE or as soon as the ConsumedServiceInstance is requested by the application, if ConsumedServiceInstance.autoRequire is set to FALSE.	
M2 Parameter	

▽

△

ConsumedEventGroup.autoRequire	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00144]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The Eventgroup Id of this eventGroup as a unique identifier of the eventgroup in this service. This identifier is used for Event Group entries as well. Please note, that the Eventgroup ID 0x0000 is reserved.	
Template Description	
EventGroup ID. Shall be unique within one system to allow service discovery.	
M2 Parameter	
ConsumedEventGroup.eventGroupIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00057]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupMulticastActivationRef	ECUC-REFERENCE-DEF
BSW Description	
The reference of a Routing Group in order to activate and setup the Socket Connection for Multicast Events of this Event Group. The Multicast address from the received Multicast Option is setup by SoAd_RequestIpAddrAssignment. The local address is the same as for the unicast events; thus, it was sent in the UDP Endpoint option of the Subscribe Event Group entry. This is usually equal to the SdConsumedEventGroupUdpActivationRef.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
This container shall be created if the EventHandler aggregates a PduActivationRoutingGroup with eventGroupControlType = activationMulticast or activationUnicastAndMulticast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00106]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**

▽

△

SdConsumedEventGroupMulticastGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SoAdSocketConnectionGroup representing the multicast data path (UDP).	
Template Description	
This reference defines the multicast address or a multicast address resource where the events of the event group are received. If the multicast address is determined via configuration and not at runtime via service discovery this reference points to the multicast address over which the events will be received. If the multicast address is determined at runtime via service discovery this reference shall be used to define the necessary local multicast address resources, i.e. RAM space in the TcpIp module in which the multicast address is stored at runtime. Please note that in this case the referenced address may be defined as ANY UDP port and ANY IP address since the multicast address will be received at runtime. If several multicast addresses are considered to be used the ConsumedEvent Group shall point to different ApplicationEndpoint objects to reserve the necessary resources in the configuration.	
M2 Parameter	
ConsumedEventGroup.eventMulticastAddress	
Mapping Rule	**Mapping Type**
If SD is used then the Client receives the Multicast Address at runtime. In this case the Application Endpoint and the corresponding NetworkEndpoint that are referenced by eventMulticastAddress from the ConsumedEventGroup define the resource where the IP Multicast address that will be determined at runtime will be stored. In case of a static configuration the eventMulticastAddress will point to a concrete IP Multicast address that can be used in the configuration.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00119]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupTcpActivationRef	ECUC-REFERENCE-DEF
BSW Description	
The reference of the Routing Group for activation of the data path for receiving TCP events. This element is also being used for getting the IP address and port number for building the TCP endpoint option for the Subscribe EventGroup entry. If no TCP methods are used in the service, this element is also being used for setting the remote address (TCP Endpoint option referenced by the Offer Service entry) and opening the TCP connection to the server before sending the Subscribe EventGroup entry. If multiple EventGroups of the same Service Instance are subscribed the TCP connection will be shared and must be opened only once.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
This container shall be created if the ConsumedEventGroup aggregates a PduActiationRouting Group with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTrigger Unicast and this PduActiationRoutingGroup points with the iPduIdentifierTcp reference to a collection of SoConIPduIdentifiers.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00105]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupTimerRef	ECUC-REFERENCE-DEF
BSW Description	
The reference of the SdClientTimer container for this eventGroup.	
Template Description	
Client Timing configuration settings that are EventGroup specific.	
M2 Parameter	
ConsumedEventGroup.sdClientTimerConfig	
Mapping Rule	**Mapping Type**
The reference to the SdClientTimer shall be created based on System Extract's SomeipSdClient EventGroupTimingConfig element that is referenced from the ConsumedEventGroup in the role sd ClientTimerConfig.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00107]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup
BSW Parameter	**BSW Type**
SdConsumedEventGroupUdpActivationRef	ECUC-REFERENCE-DEF
BSW Description	
The reference of the Routing Group for activation of the data path for receiving UDP events. This element is also being used for getting the IP address and port number for building the UDP Endpoint option or Consumed Multicast option for the Subscribe EventGroup entry. If no UDP methods are used in the service, this element is also being used for setting the remote address (UDP Endpoint option referenced by the Offer Service entry). If multiple EventGroups of the same Service Instance are subscribed the UDP Socket Connection will be shared and must be set only once.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.	
M2 Parameter	
ConsumedEventGroup.pduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
This container shall be created if the ConsumedEventGroup aggregates a PduActiationRouting Group with eventGroupControlType = activationUnicast or activationUnicastAndMulticast or trigger Unicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPdu IdentifierUdp reference to a collection of SoConIPduIdentifiers.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00104]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdConsumedMethods	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container element for representing the data path for accessing the server methods.	
Template Description	

▽

△

The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	
AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	**Mapping Type**
If the ConsumedServiceInstance contains a methodActivationRoutingGroup then this container shall be created	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00099]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedMethods
BSW Parameter	**BSW Type**
SdClientServiceActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroupRef to activate/deactivate the data path for the methods.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	
AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	**Mapping Type**
If the ConsumedServiceInstance contains a methodActivationRoutingGroup then this reference shall be created	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00102]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientService
BSW Parameter	**BSW Type**
SdServiceGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SdServiceGroupS this SdClientService belongs to.	
Template Description	
This reference assigns a set of ProvidedServiceInstances to the ConsumedProvidedServiceInstanceGroup.	
M2 Parameter	
ConsumedProvidedServiceInstanceGroup.consumedServiceInstance	
Mapping Rule	**Mapping Type**
Find ConsumedProvidedServiceInstanceGroup that points to this ConsumedServiceInstance and create the reference.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00137]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdVersionDrivenFindBehavior		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Defined the possible acceptance kinds for required service instances.		
Template Description		
Defines the service discovery find behavior.		
M2 Parameter		
ConsumedServiceInstance.versionDrivenFindBehavior		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
draft		[ECUC_SD_00140]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdClientTimer		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container specifies all timers used by the Service Discovery module for Client Services.		
Template Description		
Client specific configuration settings relevant for the SOME/IP service discovery.		
M2 Parameter		
ConsumedServiceInstance.sdClientTimerConfig		
Mapping Rule		**Mapping Type**
The Timing parameters can be derived from the SomeipSdClientServiceInstanceConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig. If multiple ConsumedServiceInstances reference the same SomeipSdClientServiceInstanceConfig, then one SdClientTimer shall be derived for those ConsumedServiceInstances which are referenced by the same ConsumedProvidedServiceInstanceGroup and for those which have ConsumedService Instance.autoRequire set to true. For all other ConsumedServiceInstances an own SdClientTimer shall be derived and referenced by the corresponding SdServerService.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SD_00043]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter		**BSW Type**
SdClientTimerInitialFindDelayMax		ECUC-FLOAT-PARAM-DEF
BSW Description		
Max value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.		
Template Description		
Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).		
M2 Parameter		
InitialSdDelayConfig.initialDelayMaxValue		

▽

△

Mapping Rule	Mapping Type
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00063]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerInitialFindDelayMin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Min value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.	
Template Description	
Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig).	
M2 Parameter	
`InitialSdDelayConfig.initialDelayMinValue`	
Mapping Rule	**Mapping Type**
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00044]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerInitialFindRepetitionsBaseDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
The base delay in [s] for find repetitions. Successive finds have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ClientService.	
Template Description	
The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay.	
M2 Parameter	
`InitialSdDelayConfig.initialRepetitionsBaseDelay`	
Mapping Rule	**Mapping Type**
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00047]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerInitialFindRepetitionsMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configuration for the maximum number of find repetitions. This parameter is mandatory for ClientService.	
Template Description	

▽

△

Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).	
M2 Parameter	
`InitialSdDelayConfig.initialRepetitionsMax`	
Mapping Rule	**Mapping Type**
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00046]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerRequestResponseMaxDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum allowable response delay to entries received by multicast in seconds. This parameter is mandatory for Consumed EventGroups.	
Template Description	
Maximum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
`RequestResponseDelay.maxValue`	
Mapping Rule	**Mapping Type**
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00036]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerRequestResponseMinDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Minimum allowable response delay to the find message in seconds. This parameter is mandatory for ConsumedEventGroups.	
Template Description	
Minimum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
`RequestResponseDelay.minValue`	
Mapping Rule	**Mapping Type**
Take information from ConsumedServiceInstance.sdClientTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00064]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerTTL	ECUC-INTEGER-PARAM-DEF

▽

△

BSW Description	
Time to live for find and subscribe messages. Note! The TTL value for find messages shall be ignored by the server service and the configuration is only kept for backward compatibility	
Template Description	
SomeipSdClientServiceInstanceConfig.serviceFindTimeToLive: This attribute represents the ability to define the time in seconds the service find is valid. Note! The TTL value for FindService entries is not used and shall be ignored by the server service. This configuration is only kept for backward compatibility. Default value if not specified shall be 0xFFFFFF. **SomeipSdClientEventGroupTimingConfig.timeToLive:** Defines the time in seconds the subscription of this event is expected by the client. this value is sent from the client to the server in the SD-subscribeEvent message.	
M2 Parameter	
SomeipSdClientServiceInstanceConfig.serviceFindTimeToLive, SomeipSdClientEventGroupTimingConfig.timeToLive	
Mapping Rule	**Mapping Type**
For SdClientTimer that is referenced from SdConsumedEventGroup by SdConsumedEventGroup TimerRef: take information from SomeipSdClientEventGroupTimingConfig.timeToLive. For SdClientTimer that is referenced from SdClientService by SdClinetServiceTimerRef: TTL for find messages is not used by the server service and the parameter is only available for backward compatibility. The value from the System Template can be ignored. Default value if not specified shall be 0xFFFFFF.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00075]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdSubscribeEventgroupRetryDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time in seconds when a subscription to an event group shall be retriggered, if no SubscribeEventGroupAck or Subscribe EventGroupNack was received.	
Template Description	
This attribute defines the interval in seconds to re-trigger a subscription to a Eventgroup, if a retry to subscribe to a Eventgroup is configured (subscribeEventgroupRetryMax > 0).	
M2 Parameter	
SomeipSdClientEventGroupTimingConfig.subscribeEventgroupRetryDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00133]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdSubscribeEventgroupRetryMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum count of retry a subscription, if a subscription to an event group is not acknowledged by SubscribeEventGroupAck or SubscribeEventGroupNack. 0x0=no retry, 0xFF=retry forever (as long as the event group is requested)	

▽

△

Template Description	
This attribute define the maximum counts of retries to subscribe to an Eventgroup. If the value is set to 0 no retry shall be done. If the value is set to 255 the retry shall be done as along as the Eventgroup is requested and no SubscribeEventGroup Ack was received.	
M2 Parameter	
SomeipSdClientEventGroupTimingConfig.subscribeEventgroupRetryMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00132]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdInstanceHostname	ECUC-STRING-PARAM-DEF
BSW Description	
Configuration parameter to specify the Hostname.	
Template Description	
Defines the fully qualified domain name (FQDN) e.g. some.example.host.	
M2 Parameter	
NetworkEndpoint.fullyQualifiedDomainName	
Mapping Rule	**Mapping Type**
Shall be derived from the NetworkEndpoint.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00012]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdInstanceMulticastRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the received PDU.	
Template Description	
M2 Parameter	
PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurpose Pdu that represents the SdPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00081]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**

▽

△

SdInstanceTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the transmitted PDU.	
Template Description	
M2 Parameter	
PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurpose Pdu that represents the SdPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00030]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdInstanceUnicastRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the received PDU.	
Template Description	
M2 Parameter	
PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurpose Pdu that represents the SdPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00027]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdServerService	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies all parameters used by Server services.	
Template Description	
Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter	
ProvidedServiceInstance	
Mapping Rule	**Mapping Type**
Create container for each existing ProvidedServiceInstance that is available in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00004]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdEventHandler	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container Element for representing an EventGroup as part of the Service Instance.	
Template Description	
M2 Parameter	
ProvidedServiceInstance.eventHandler	
Mapping Rule	**Mapping Type**
Create container for every existing EventHandler that is aggregated by the ProvidedService Instance	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00055]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	**BSW Type**
SdEventHandlerEventGroupId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The EventGroup Id of this EventGroup as a unique identifier of the EventGroup in this service. This identifier is used for Event Group entries as well. Please note, that the Eventgroup ID 0x0000 is reserved.	
Template Description	
Unique Identifier that identifies the EventGroup in SOME/IP. This Identifier is sent as Eventgroup ID in SOME/IP Service Discovery messages.	
M2 Parameter	
EventHandler.eventGroupIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00061]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	**BSW Type**
SdEventHandlerMulticast	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The subcontainer including the Routing Group for Activation of Events sent over Multicast. The activation ref is also being used for identification of the related Socket Connection in order to find the Multicast Address used in the Multicast Option referenced by the Subscribe EventGroup Ack entry.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter	
EventHandler.pduActivationRoutingGroup	
Mapping Rule	**Mapping Type**



△

This container shall be created if the EventHandler aggregates a PduActivationRoutingGroup with eventGroupControlType = activationMulticast or activationUnicastAndMulticast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00094]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerMulticast
BSW Parameter	**BSW Type**
SdEventActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Create this reference in the SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerUdp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicastAndMulticast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast or activationUnicastAndMulticast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00096]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerMulticast
BSW Parameter	**BSW Type**
SdMulticastEventSoConRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the SoAdSocketConnection representing the Eventhandler Multicast data path (UDP).	
Template Description	
Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the mulicastThreshold is exceeded. This address is transmitted in the SD-SubscribeEventGroupAck Message to client (answer to SD-SubscribeEventGroup).	
M2 Parameter	
EventHandler.eventMulticastAddress	
Mapping Rule	**Mapping Type**
Create this reference if the EventHandler in the System Description contains an eventMulticast Address reference to an ApplicationEndpoint that points to a NetworkEndpoint that in turn defines an IP MulticastAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00118]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler

BSW Parameter	BSW Type
SdEventHandlerMulticastThreshold	ECUC-INTEGER-PARAM-DEF

BSW Description

Specifies the number of subscribed clients with different endpoint information (see SWS_SD_00754) that triggers the Server to change the transmission of events via the Eventhandler Multicast connection.

If configured to 0 only Consumed Evengroup unicast connections and Consumed Eventgroup multicast connections will be used.

If configured to 1 the first client and all further subscribed clients will be served via the Eventhandler Multicast connection as configured in SdMulticastEventSoConRef.

If configured to n up to n-1 clients with different endpoint information will be served via Consumed Evengroup unicast connections and Consumed Eventgroup multicast connections. As soon as the number of subscribed clients with different endpoint information reaches n, then all subscribed clients are served via the Eventhandler Multicast connection as configured in SdMulticastEventSoConRef.

This does not influence the handling of initial events.

Template Description

Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast.

If configured to 0 only unicast will be used. If configured to 1 the first client will be already served by multicast. If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast.

This does not influence the handling of initial events, which are served using unicast only.

M2 Parameter

`EventHandler.multicastThreshold`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00097]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler

BSW Parameter	BSW Type
SdEventHandlerTcp	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over TCP.

The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to find the related client by iterating the SdEventHandlerTcp elements (remote address statically configured or automatically set by opening TCP connection before subscription).

Template Description

The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.

M2 Parameter

`EventHandler.pduActivationRoutingGroup`

Mapping Rule	Mapping Type
This container shall be created if the EventHandler aggregates a PduActiationRoutingGroup with eventGroupControlType = activationUnicast or triggerUnicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPduIdentifierTcp reference to a collection of SoCon IPduIdentifiers.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00093]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp
BSW Parameter	**BSW Type**
SdEventActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Create this reference in the SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerUdp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationUnicastAndMulticast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast or activationUnicastAndMulticast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00096]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp
BSW Parameter	**BSW Type**
SdEventTriggeringRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Create this reference if eventGroupControlType in the PduActivationRoutingGroup is set to trigger Unicast or activationAndTriggerUnicast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00095]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	**BSW Type**
SdEventHandlerTimerRef	ECUC-REFERENCE-DEF



△

BSW Description	
The reference of the SdServerTimer container for this EventGroup.	
Template Description	
Server Timing configuration settings that are EventGroup specific.	
M2 Parameter	
EventHandler.sdServerEgTimingConfig	
Mapping Rule	**Mapping Type**
The reference to the SdServerTimer shall be created pointing to the SdServerTimer container which was created based on the EventHandler.sdServerEgTimingConfig. If multiple Provided ServiceInstances reference the same SomeipSdServerServiceInstanceConfig, then one SdServer Timer shall be derived for those ProvidedServiceInstances which are referenced by the same ConsumedProvidedServiceInstanceGroup and for those which have ProvidedServiceInstance.auto Available set to true. For all other ProvidedServiceInstances an own SdServerTimer shall be derived and referenced by the corresponding SdServerService.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00113]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	**BSW Type**
SdEventHandlerUdp	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over UDP. The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to set the remote address (either unicast address or multicast address) of the client or find the related client by iterating the SdEventHandlerUdp elements (remote address statically configured or automatically set by method call before subscription).	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter	
EventHandler.pduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
This container shall be created if the EventHandler aggregates a PduActiationRoutingGroup with eventGroupControlType = activationUnicast or activationUnicastAndMulticast or triggerUnicast or activationAndTriggerUnicast and this PduActiationRoutingGroup points with the iPduIdentifierUdp reference to a collection of SoConIPduIdentifiers.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00092]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp
BSW Parameter	**BSW Type**
SdEventActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe). This is usually equal to the SdEventActivationRef referenced by SdEventHandlerUdp	
Template Description	

▽

△

This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Create this reference in the SdEventHandlerTcp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerUdp container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationUnicast or activationUnicastAndMulticast or activationAndTriggerUnicast. Create this reference in the SdEventHandlerMulticast container if the eventGroupControlType in the PduActivationRoutingGroup is set to activationMulticast or activationUnicastAndMulticast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00096]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp
BSW Parameter	**BSW Type**
SdEventTriggeringRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.	
Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Create this reference if eventGroupControlType in the PduActivationRoutingGroup is set to triggerUnicast or activationAndTriggerUnicast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00095]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdProvidedMethods	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container element for representing the needed elements of the data path for the methods provided by the service.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	
AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	**Mapping Type**

▽

△

If the ProvidedServiceInstance contains a methodActivationRoutingGroup then this container shall be created	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00087]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdProvidedMethods
BSW Parameter	**BSW Type**
SdServerServiceActivationRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a SoAdRoutingGroup to activated and deactivate the data path for methods of the service.	
Template Description	
The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods).	
M2 Parameter	
AbstractServiceInstance.methodActivationRoutingGroup	
Mapping Rule	**Mapping Type**
If the ProvidedServiceInstance contains a methodActivationRoutingGroup then this reference shall be created	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00090]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerCapabilityRecord	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" -- password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")	
Template Description	
A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter	
AbstractServiceInstance.capabilityRecord	
Mapping Rule	**Mapping Type**
1:1 mapping to ProvidedServiceInstance.capabilityRecord.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00032]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord
BSW Parameter	**BSW Type**
SdServerCapabilityRecordKey	ECUC-STRING-PARAM-DEF

▽

△

BSW Description	
Defines a CapabilityRecord key.	
Template Description	
Defines a key.	
M2 Parameter	
TagWithOptionalValue.key	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00033]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord
BSW Parameter	**BSW Type**
SdServerCapabilityRecordValue	ECUC-STRING-PARAM-DEF
BSW Description	
Defines the corresponding CapabilityRecord value.	
Template Description	
Defines the corresponding value.	
M2 Parameter	
TagWithOptionalValue.value	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00034]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceAllowedConsumers	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines a list of consumers that are allowed to access this SdServerService.	
Template Description	
NetworkEndpoints on which the ConsumedServiceInstances that are communicating with this ProvidedServiceInstance are allowed to be located so that the ACL check in the ServiceDiscovery is successful and the connection is allowed to be established.	
M2 Parameter	
ProvidedServiceInstance.allowedServiceConsumer	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00155]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerServiceAllowedConsumers
BSW Parameter	**BSW Type**
SdIpAddress	ECUC-STRING-PARAM-DEF
BSW Description	
This parameter defines the IP Address of the remote communication partner.	
Template Description	
M2 Parameter	
ApplicationEndpoint.networkEndpoint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00148]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerServiceAllowedConsumers
BSW Parameter	**BSW Type**
SdIpAddressType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
This parameter defines the IP version that is used for communication with the remote communication partner.	
Template Description	
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).	
M2 Parameter	
NetworkEndpoint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_Sd_00149]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceAutoAvailable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If existing and set to true, this Service will be set to "Available" on start.	
Template Description	
Defines that this ProvidedServiceInstance shall be offered by the service discovery at ECU start.	
M2 Parameter	
ProvidedServiceInstance.autoAvailable	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00138]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Id to identify the service. This is unique for the service interface.	
Template Description	
This attribute represents the ability to describe the SOME/IP service ID that is offered.	
M2 Parameter	
ProvidedServiceInstance.serviceIdentifier	
Mapping Rule	**Mapping Type**
Shall be derived from ProvidedServiceInstance.serviceIdentifier	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00009]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configuration parameter to specify Instance Id of the Service implemented by the Server Service.	
Template Description	
Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.	
M2 Parameter	
ProvidedServiceInstance.instanceIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00011]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceLoadBalancingPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.	
Template Description	
Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.	
M2 Parameter	
ProvidedServiceInstance.loadBalancingPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00129]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceLoadBalancingWeight	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.	
Template Description	
Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.	
M2 Parameter	
ProvidedServiceInstance.loadBalancingWeight	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00130]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceMajorVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Major version number of the Service as used in SD Entries.	
Template Description	
Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.	
M2 Parameter	
AbstractServiceInstance.majorVersion	
Mapping Rule	**Mapping Type**
Shall be derived from ProvidedServiceInstance.majorVersion	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00068]

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceMinorVersion	ECUC-INTEGER-PARAM-DEF
BSW Description	
Minor version number of the Service as used e.g. in Offer Service entries.	
Template Description	
Minor Version of the Service that is provided by this ProvidedServiceInstance.	
M2 Parameter	
ProvidedServiceInstance.minorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00069]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter	**BSW Type**	
SdServerServiceTimerRef	ECUC-REFERENCE-DEF	
BSW Description		
The reference of the SdServerTimer container for this service.		
Template Description		
Server specific configuration settings relevant for the SOME/IP service discovery.		
M2 Parameter		
`ProvidedServiceInstance.sdServerTimerConfig`		
Mapping Rule		**Mapping Type**
The reference to the SdServerTimer shall be created pointing to the SdServerTimer container which was created based on the ProvidedServiceInstance.sdServerTimerConfig.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SD_00086]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter	**BSW Type**	
SdServerServiceUdpRef	ECUC-REFERENCE-DEF	
BSW Description		
Reference to SoAdSocketConnectionGroup used for methods. This is used to access the local IP address and port for building the endpoint option for offers of this service.		
Template Description		
The local address over which the CSI is consumed (udp, tcp or both).		
M2 Parameter		
`ConsumedServiceInstance.localUnicastAddress`		
Mapping Rule		**Mapping Type**
This reference shall be derived from the ProvidedServiceInstance.localUnicastAddress reference to an ApplicationEndpoint that defines a UDP Port. If the ProvidedServiceInstance does not reference a UDP Port with the localUnicastAddress reference the SdServerServiceTcpRef shall be skipped.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SD_00089]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter	**BSW Type**	
SdServiceGroupRef	ECUC-REFERENCE-DEF	
BSW Description		
Reference to the SdServiceGroupS this SdServerService belongs to.		
Template Description		
This reference assigns a set of ConsumedServiceInstances to the ConsumedProvidedServiceInstanceGroup.		
M2 Parameter		
`ConsumedProvidedServiceInstanceGroup.providedServiceInstance`		
Mapping Rule		**Mapping Type**

▽

△

Find ConsumedProvidedServiceInstanceGroup that points to this ProvidedServiceInstance and create the reference.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00136]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdServerTimer	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies all timers used by the Service Discovery module for Server Services.	
Template Description	
Server specific configuration settings relevant for the SOME/IP service discovery.	
M2 Parameter	
ProvidedServiceInstance.sdServerTimerConfig	
Mapping Rule	**Mapping Type**
The Timing parameters can be derived from the SomeipSdServerServiceInstanceConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00035]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerInitialOfferDelayMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
Max value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
Template Description	
Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).	
M2 Parameter	
InitialSdDelayConfig.initialDelayMaxValue	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00039]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerInitialOfferDelayMin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Min value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
Template Description	

▽

△

Min Value in seconds to delay randomly the first offer or the transmission of a find message (if aggregated by SdClientConfig).	
M2 Parameter	
InitialSdDelayConfig.initialDelayMinValue	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00038]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerInitialOfferRepetitionBaseDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
The base delay in [s] for offer repetitions. Successive offers have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ServerService.	
Template Description	
The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig). Successive find messages have an exponential back off delay.	
M2 Parameter	
InitialSdDelayConfig.initialRepetitionsBaseDelay	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00041]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerInitialOfferRepetitionsMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Configure the maximum amount of offer repetition. This parameter is mandatory for ServerService.	
Template Description	
Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).	
M2 Parameter	
InitialSdDelayConfig.initialRepetitionsMax	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.sdServerTimerConfig	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00040]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**

▽

△

SdServerTimerOfferCyclicDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Interval between cyclic offers in the main phase. This parameter is mandatory for ServerService.	
Template Description	
Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds) and greater then 0.	
M2 Parameter	
SomeipSdServerServiceInstanceConfig.offerCyclicDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00076]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerRequestResponseMaxDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum allowable response delay to entries received by multicast in seconds.	
Template Description	
Maximum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
RequestResponseDelay.maxValue	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.requestResponseDelay	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00114]

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerRequestResponseMinDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Minimum allowable response delay to entries received by multicast in seconds.	
Template Description	
Minimum allowable response delay to entries received by multicast in seconds.	
M2 Parameter	
RequestResponseDelay.minValue	
Mapping Rule	**Mapping Type**
Take information from ProvidedServiceInstance.requestResponseDelay	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SD_00115]

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerTimer	
BSW Parameter		**BSW Type**
SdServerTimerTTL		ECUC-INTEGER-PARAM-DEF
BSW Description		
Time to live for offer service.		
Template Description		
Defines the time in seconds the service offer is valid.		
M2 Parameter		
SomeipSdServerServiceInstanceConfig.serviceOfferTimeToLive		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SD_00037]

BSW Module	BSW Context	
Sd	Sd/SdConfig	
BSW Parameter		**BSW Type**
SdServiceGroup		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Contains the configuration parameters of the AUTOSAR SD module's SdServiceGroupS.		
Template Description		
The AUTOSAR ServiceDiscovery is able to start and to stop ClientServices and ServerServices,respectively, at runtime. A SdServiceGroup contains several ClientServices and ServerServices, respectively.		
M2 Parameter		
ConsumedProvidedServiceInstanceGroup		
Mapping Rule		**Mapping Type**
Create container for each CoreCommunication::ConsumedProvidedServiceInstanceGroup that is contained in the ECU Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SD_00134]

C.52 SecOC

BSW Module	BSW Context	
SecOC	SecOC	
BSW Parameter		**BSW Type**
SecOCRxPduProcessing		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Contains the parameters to configure the RxPdus to be verified by the SecOC module.		
Template Description		

▽

△

If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator). If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu.	
M2 Parameter	
SecuredIPdu	
Mapping Rule	**Mapping Type**
This container shall be created for every SecuredIPdu that is received by the regarded Ecu. The information whether the SecuredIPdu is transmitted or received by the Ecu shall be derived from PduTriggering.iPduPort reference. If an IPduPort of the Ecu with the communicationDirection = out is referenced then the SecuredIPdu is transmitted. If an IPduPort of the Ecu with the communicationDirection = in is referenced then the SecuredIPdu is received.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00011]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthDataFreshnessLen	ECUC-INTEGER-PARAM-DEF
BSW Description	
The length of the external authentic PDU data in bits (uint16).	
Template Description	
This attribute defines the length in bits of the authentic PDU data that is passed to the SWC that verifies and generates the Freshness.	
M2 Parameter	
SecureCommunicationProps.authDataFreshnessLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00082]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthDataFreshnessStartPosition	ECUC-INTEGER-PARAM-DEF
BSW Description	
This value determines the start position in bits (uint16) of the Authentic PDU that shall be passed on to the Freshness SWC. The bit counting is done according to TPS_SYST_01068 and the bit ordering is done according to TPS_SYST_01069.	
Template Description	
This value determines the start position in bits of the Authentic PDU that shall be passed on to the SWC that verifies and generates the Freshness. The bit counting is done according to TPS_SYST_01068.	
M2 Parameter	
SecureCommunicationProps.authDataFreshnessStartPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00081]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthInfoTruncLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the length in bits of the authentication code to be included in the payload of the Secured I-PDU.	
Template Description	
This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.	
M2 Parameter	
SecureCommunicationAuthenticationProps.authInfoTxLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00095]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthenticationBuildAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the number of authentication build attempts.	
Template Description	
This attribute specifies the number of authentication build attempts.	
M2 Parameter	
SecureCommunicationProps.authenticationBuildAttempts	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00079]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthenticationVerifyAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the number of authentication verify attempts that are to be carried out when the verification of the authentication information failed for a given Secured I-PDU. If zero is set, then only one authentication verification attempt is done.	
Template Description	

▽

△

This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given SecuredIPdu. If zero is set than only one authentication attempt is done.	
M2 Parameter	
SecureCommunicationProps.authenticationRetries	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00080]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCDataId	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines a unique numerical identifier for the Secured I-PDU.	
Template Description	
This attribute defines a numerical identifier for the Secured I-PDU.	
M2 Parameter	
SecureCommunicationProps.dataId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00014]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCDynamicRuntimeLengthHandling	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines whether the length information for handling this received Pdu is taken from the configuration or from the actually provided length information during runtime. true: SecuredIPdu length information is taken from the actually provided length information during runtime. false: SecuredIPdu length information is taken from parameter PduLength of the Pdu.	
Template Description	
Defines whether the length information for handling this SecuredIPdu with SecuredIPdu.useSecuredPduHeader=noHeader is taken from the configuration or from the actually provided length information during runtime. true: SecuredIPdu length information is taken from the actually provided length information during runtime. false: SecuredIPdu length information is taken from the configuration.	
M2 Parameter	
SecuredIPdu.dynamicRuntimeLengthHandling	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00118]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter		**BSW Type**
SecOCFreshnessValueId		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.		
Template Description		
This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.		
M2 Parameter		
SecureCommunicationProps.freshnessValueId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SecOC_00021]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter		**BSW Type**
SecOCFreshnessValueLength		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.		
Template Description		
This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.		
M2 Parameter		
SecureCommunicationFreshnessProps.freshnessValueLength		
Mapping Rule		**Mapping Type**
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SecOC_00015]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter		**BSW Type**
SecOCFreshnessValueTruncLength		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured I-PDU.		
Template Description		

▽

△

This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.	
M2 Parameter	
SecureCommunicationFreshnessProps.freshnessValueTxLength	
Mapping Rule	**Mapping Type**
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueTxLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00094]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCRxAuthServiceConfigRef	ECUC-REFERENCE-DEF
BSW Description	
This reference is used to define which crypto service function is called for authentication. If PDUs with a dynamic length are used (e.g. CanTP or Dynamic Length PDUs) a MAC algorithm has to be chosen, that is not vulnerable to length extension attack (e.g. CMAC/HMAC). If SecOCUseAuthenticatedEncryption is set to TRUE, an AEAD algorithm has to be chosen.	
Template Description	
This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced SecuredIPdu. Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role iPdu.	
M2 Parameter	
PduTriggering.secOcCryptoServiceMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00048]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxAuthenticPduLayer
BSW Parameter	**BSW Type**
SecOCRxAuthenticLayerPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.	
M2 Parameter	
SecuredIPdu.payload	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the verified SecuredIPdu PduTriggering	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00045]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing	
BSW Parameter		**BSW Type**
SecOCRxPduSecuredArea		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator verification algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator verification algorithm.		
Template Description		
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.		
M2 Parameter		
SecureCommunicationProps.securedAreaOffset		
Mapping Rule		**Mapping Type**
Create container if the securedAreaOffset and securedAreaLength is defined for the SecuredIPdu in the System Description.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SecOC_00089]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxPduSecuredArea	
BSW Parameter		**BSW Type**
SecOCSecuredRxPduLength		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the length (in bytes) of the area within the Pdu which is secured		
Template Description		
This attribute defines the length in bytes of the area within the payload Pdu which will be secured.		
M2 Parameter		
SecureCommunicationProps.securedAreaLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SecOC_00091]

BSW Module	BSW Context	
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxPduSecuredArea	
BSW Parameter		**BSW Type**
SecOCSecuredRxPduOffset		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the start position (offset in bytes) of the area within the Pdu which is secured		
Template Description		
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.		
M2 Parameter		
SecureCommunicationProps.securedAreaOffset		
Mapping Rule		**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00090]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer
BSW Parameter	**BSW Type**
SecOCRxSecuredPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the Pdu that is received by the SecOC module from the PduR. For this Pdu the Mac verification is provided.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCRxSecuredPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = false or not defined	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00069]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu
BSW Parameter	**BSW Type**
SecOCAuthPduHeaderLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	
This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but no Header, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SecuredIPdu.useSecuredPduHeader	
Mapping Rule	**Mapping Type**
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecured IPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "secured PduHeader32Bit" the value shall be 4.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00093]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu
BSW Parameter	**BSW Type**
SecOCRxSecuredLayerPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the received SecuredIPdu PduTriggering	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00042]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu
BSW Parameter	**BSW Type**
SecOCSecuredRxPduVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.	
Template Description	
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.	
M2 Parameter	
IPduPort.rxSecurityVerification	
Mapping Rule	**Mapping Type**
SecOCSecuredRxPduVerification is true if rxSecurityVerification is not defined, otherwise SecOCSecuredRxPduVerification = rxSecurityVerification.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00092]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPdu
BSW Parameter	**BSW Type**
SecOcVerifyErrorDemEventRef	ECUC-REFERENCE-DEF
BSW Description	
This parameter reference to the DemEvent needed to be reported in case of verification success or fail. If the parameter is not defined SecOc shall not notify the DEM module.	
Template Description	
This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced SecuredIPdu. Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role iPdu.	
M2 Parameter	

▽

△

PduTriggering.secOcCryptoServiceMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00120]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer
BSW Parameter	**BSW Type**
SecOCRxSecuredPduCollection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies two Pdus that are received by the SecOC module from the PduR and a message linking between them. SecOCRxAuthenticPdu contains the original Authentic I-PDU, i.e. the secured data, and the SecOCRxCryptographicPdu contains the Authenticator, i.e. the actual Authentication Information.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCRxSecuredPduCollection if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00067]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection
BSW Parameter	**BSW Type**
SecOCRxAuthenticPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the PDU (that is received by the SecOC module from the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCRxAuthenticPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00061]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/ SecOCRxAuthenticPdu
BSW Parameter	**BSW Type**
SecOCAuthPduHeaderLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	
This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but no Header, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SecuredIPdu.useSecuredPduHeader	
Mapping Rule	**Mapping Type**
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecured IPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "secured PduHeader32Bit" the value shall be 4.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_ 00093]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/ SecOCRxAuthenticPdu
BSW Parameter	**BSW Type**
SecOCRxAuthenticPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the received Authentic SecuredIPdu PduTriggering referenced by SecuredIPdu.payload	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_ 00063]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection
BSW Parameter	**BSW Type**
SecOCRxCryptographicPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the Cryptographic Pdu that is received by the SecOC module from the PduR.	
Template Description	

▽

△

If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCRxCryptographicPdu if the SecuredIPdu is received by the regarded ECU and the attribute useAsCryptographicIPdu = true and SecuredIPdu.payload refers to a PduTriggering which the Ecu Instance receives as well.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00064]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCRxCryptographicPdu
BSW Parameter	**BSW Type**
SecOCRxCryptographicPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.	
M2 Parameter	
SecuredIPdu.payload	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the received Cryptographic SecuredIPdu Pdu Triggering	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00066]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection
BSW Parameter	**BSW Type**
SecOCSecuredRxPduVerification	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This parameter defines whether the signature authentication or MAC verification shall be performed on this Secured I-PDU. If set to false, the SecOC module extracts the Authentic I-PDU from the Secured I-PDU without verification.	
Template Description	
This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.	
M2 Parameter	
IPduPort.rxSecurityVerification	
Mapping Rule	**Mapping Type**
SecOCSecuredRxPduVerification is true if rxSecurityVerification is not defined, otherwise Sec OCSecuredRxPduVerification = rxSecurityVerification.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00092]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCUseMessageLink

BSW Parameter	BSW Type
SecOCMessageLinkLen	ECUC-INTEGER-PARAM-DEF

BSW Description
Length of the Message Linker inside the Authentic I-PDU in bits.

Template Description
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the Authentic IPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.

M2 Parameter
SecureCommunicationProps.messageLinkLength

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00060]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing/SecOCRxSecuredPduLayer/SecOCRxSecuredPduCollection/SecOCUseMessageLink

BSW Parameter	BSW Type
SecOCMessageLinkPos	ECUC-INTEGER-PARAM-DEF

BSW Description
The position of the Message Linker inside the Authentic I-PDU in bits. The bit counting is done according to 01068 and the bit ordering is done according to TPS_SYST_01069.

Template Description
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the Authentic IPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.

M2 Parameter
SecureCommunicationProps.messageLinkPosition

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00059]

BSW Module	BSW Context
SecOC	SecOC/SecOCRxPduProcessing

BSW Parameter	BSW Type
SecOCUseAuthDataFreshness	ECUC-BOOLEAN-PARAM-DEF

BSW Description

▽

△

A Boolean value that indicates if a part of the Authentic I-PDU is used for freshness generation and verification.	
Template Description	
This attribute describes whether a part of AuthenticPdu contained in a SecuredIPdu shall be passed on to the SWC that verifies and generates the Freshness. The part of the Authentic-PDU is defined by the authDataFreshnessStartPosition and authDataFreshnessLength.	
M2 Parameter	
IPduPort.useAuthDataFreshness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00083]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCRxPduProcessing
BSW Parameter	**BSW Type**
SecOCUseAuthenticatedEncryption	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If set to TRUE, Authenticated Encryption with Associated Data shall be applied.	
Template Description	
A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If it set to TRUE, Authenticated Encryption with Associated Data (AEAD) shall be applied.	
M2 Parameter	
SecureCommunicationProps.useAuthenticatedEncryption	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00125]

BSW Module	**BSW Context**
SecOC	SecOC
BSW Parameter	**BSW Type**
SecOCTxPduProcessing	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Contains the parameters to configure the TxPdus to be secured by the SecOC module.	
Template Description	
If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator). If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu.	
M2 Parameter	
SecuredIPdu	
Mapping Rule	**Mapping Type**

▽

△

This container shall be created for every SecuredIPdu that is transmitted by the regarded Ecu. The information whether the SecuredIPdu is transmitted or received by the Ecu shall be derived from PduTriggering.iPduPort reference. If an IPduPort of the Ecu with the communicationDirection = out is referenced then the SecuredIPdu is transmitted. If an IPduPort of the Ecu with the communicationDirection = in is referenced then the SecuredIPdu is received.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00012]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthInfoTruncLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the length in bits of the authentication code to be included in the payload of the Secured I-PDU.	
Template Description	
This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.	
M2 Parameter	
SecureCommunicationAuthenticationProps.authInfoTxLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00095]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCAuthenticationBuildAttempts	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter specifies the number of authentication build attempts.	
Template Description	
This attribute specifies the number of authentication build attempts.	
M2 Parameter	
SecureCommunicationProps.authenticationBuildAttempts	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00079]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCDataId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

This parameter defines a unique numerical identifier for the Secured I-PDU.	
Template Description	
This attribute defines a numerical identifier for the Secured I-PDU.	
M2 Parameter	
SecureCommunicationProps.dataId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00014]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCFreshnessValueId	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
Template Description	
This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
M2 Parameter	
SecureCommunicationProps.freshnessValueId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00021]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCFreshnessValueLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
Template Description	
This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
M2 Parameter	
SecureCommunicationFreshnessProps.freshnessValueLength	
Mapping Rule	**Mapping Type**
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00015]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing

BSW Parameter	BSW Type
SecOCFreshnessValueTruncLength	ECUC-INTEGER-PARAM-DEF

BSW Description

This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured I-PDU.

Template Description

This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.

M2 Parameter

`SecureCommunicationFreshnessProps.freshnessValueTxLength`

Mapping Rule	Mapping Type
1:1 mapping if value is set by SecureCommunicationFreshnessProps.freshnessValueTxLength. If the attribute is not set to a value the parameter value of 0 shall be assumed which means that no Freshness Value is included in the Secured I-PDU.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00094]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing

BSW Parameter	BSW Type
SecOCTxAuthServiceConfigRef	ECUC-REFERENCE-DEF

BSW Description

This reference is used to define which crypto service function is called for authentication. If PDUs with a dynamic length are used (e.g. CanTP or Dynamic Length PDUs) a MAC algorithm has to be chosen, that is not vulnerable to length extension attack (e.g. CMAC/HMAC). If SecOCUseAuthenticatedEncryption is set to TRUE, an AEAD algorithm has to be chosen.

Template Description

This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced SecuredIPdu.

Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role iPdu.

M2 Parameter

`PduTriggering.secOcCryptoServiceMapping`

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00013]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxAuthenticPduLayer

BSW Parameter	BSW Type
SecOCTxAuthenticLayerPduRef	ECUC-REFERENCE-DEF

BSW Description

Reference to the global Pdu.

Template Description

▽

△

Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.	
M2 Parameter	
SecuredIPdu.payload	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the to be secured SecuredIPdu PduTriggering	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00025]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing
BSW Parameter	**BSW Type**
SecOCTxPduSecuredArea	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies an area in the Authentic I-Pdu that will be the input to the Authenticator generation algorithm. If this container does not exist in the configuration the complete Authentic I-Pdu will be the input to the Authenticator generation algorithm.	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	
SecureCommunicationProps.securedAreaOffset	
Mapping Rule	**Mapping Type**
Create container if the securedAreaOffset and securedAreaLength is defined for the SecuredIPdu in the System Description.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00086]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxPduSecuredArea
BSW Parameter	**BSW Type**
SecOCSecuredTxPduLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the length (in bytes) of the area within the Pdu which shall be secured	
Template Description	
This attribute defines the length in bytes of the area within the payload Pdu which will be secured.	
M2 Parameter	
SecureCommunicationProps.securedAreaLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00088]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxPduSecuredArea
BSW Parameter	**BSW Type**
SecOCSecuredTxPduOffset	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the start position (offset in bytes) of the area within the Pdu which shall be secured	
Template Description	
This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.	
M2 Parameter	
SecureCommunicationProps.securedAreaOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00087]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer
BSW Parameter	**BSW Type**
SecOCTxSecuredPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies one Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. This Pdu contains the cryptographic information.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCTxSecuredPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = false or not defined	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00070]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu
BSW Parameter	**BSW Type**
SecOCAuthPduHeaderLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.	
Template Description	

▽

△

This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but no Header, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.	
M2 Parameter	
SecuredIPdu.useSecuredPduHeader	
Mapping Rule	**Mapping Type**
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecured IPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "secured PduHeader32Bit" the value shall be 4.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_ 00093]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPdu
BSW Parameter	**BSW Type**
SecOCTxSecuredLayerPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the sent SecuredIPdu PduTriggering	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_ 00027]

BSW Module	**BSW Context**
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer
BSW Parameter	**BSW Type**
SecOCTxSecuredPduCollection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated. Two separate Pdus are transmitted to the PduR: Authentic I-PDU and Cryptographic I-PDU.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCTxSecuredPduCollection if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_ 00071]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection

BSW Parameter	BSW Type
SecOCTxAuthenticPdu	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

This container specifies the PDU (that is transmitted by the SecOC module to the PduR) which contains the Secured I-PDU Header and the Authentic I-PDU.

Template Description

If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data.

If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.

M2 Parameter

SecuredIPdu.useAsCryptographicIPdu

Mapping Rule	Mapping Type
Create the SecOCTxAuthenticPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00072]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

BSW Parameter	BSW Type
SecOCAuthPduHeaderLength	ECUC-INTEGER-PARAM-DEF

BSW Description

This parameter indicates the length (in bytes) of the Secured I-PDU Header in the Secured I-PDU. The length of zero means there's no header in the PDU.

Template Description

This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but no Header, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

M2 Parameter

SecuredIPdu.useSecuredPduHeader

Mapping Rule	Mapping Type
The SecOCAuthPduHeaderLength value shall be derived from useSecuredPduHeader in the following way: If useSecuredIPduHeader is set to "noHeader" the value shall be 0. If useSecured IPduHeader is set to "securedPduHeader08Bit" the value shall be 1. If useSecuredIPduHeader is set to "securedPduHeader16Bit" the value shall be 2. If useSecuredIPduHeader is set to "secured PduHeader32Bit" the value shall be 4.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00093]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxAuthenticPdu

BSW Parameter	BSW Type
SecOCTxAuthenticPduRef	ECUC-REFERENCE-DEF

△

BSW Description	
Reference to the global Pdu.	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the sent Authentic SecuredIPdu PduTriggering referenced by SecuredIPdu.payload	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00056]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection
BSW Parameter	**BSW Type**
SecOCTxCryptographicPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the Cryptographic Pdu that is transmitted by the SecOC module to the PduR after the Mac was generated.	
Template Description	
If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.	
M2 Parameter	
SecuredIPdu.useAsCryptographicIPdu	
Mapping Rule	**Mapping Type**
Create the SecOCTxCryptographicPdu if the SecuredIPdu is sent by the regarded ECU and the attribute useAsCryptographicIPdu = true and SecuredIPdu.payload refers to a PduTriggering which the Ecu Instance sends as well.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00073]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCTxCryptographicPdu
BSW Parameter	**BSW Type**
SecOCTxCryptographicPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global Pdu.	
Template Description	
Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.	
M2 Parameter	
SecuredIPdu.payload	
Mapping Rule	**Mapping Type**
Reference to the EcuC Pdu which represents the sent Cryptographic SecuredIPdu PduTriggering	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_SecOC_00058]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCUseMessageLink

BSW Parameter	BSW Type
SecOCMessageLinkLen	ECUC-INTEGER-PARAM-DEF

BSW Description
Length of the Message Linker inside the Authentic I-PDU in bits.

Template Description
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the Authentic IPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.

M2 Parameter
SecureCommunicationProps.messageLinkLength

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00060]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing/SecOCTxSecuredPduLayer/SecOCTxSecuredPduCollection/SecOCUseMessageLink

BSW Parameter	BSW Type
SecOCMessageLinkPos	ECUC-INTEGER-PARAM-DEF

BSW Description
The position of the Message Linker inside the Authentic I-PDU in bits. The bit counting is done according to 01068 and the bit ordering is done according to TPS_SYST_01069.

Template Description
SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the Authentic IPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.

M2 Parameter
SecureCommunicationProps.messageLinkPosition

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00059]

BSW Module	BSW Context
SecOC	SecOC/SecOCTxPduProcessing

BSW Parameter	BSW Type
SecOCUseAuthenticatedEncryption	ECUC-BOOLEAN-PARAM-DEF

▽

△

BSW Description	
A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If it set to TRUE, Authenticated Encryption with Associated Data shall be applied	
Template Description	
A Boolean value that indicates if the Authentic I-PDU shall be handled as Confidential I-PDU. If it set to TRUE, Authenticated Encryption with Associated Data (AEAD) shall be applied.	
M2 Parameter	
SecureCommunicationProps.useAuthenticatedEncryption	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SecOC_00124]

C.53 SoAd

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig	
BSW Parameter		**BSW Type**
SoAdPduRoute		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Describes the path of a PDU from an upper layer of the SoAd to the socket in the TCP/IP stack for transmission. This PDU can consume meta data items of type SOCKET_CONNECTION_ID_16.		
Template Description		
Reference to a Pdu that is transmitted over a socket connection.		
M2 Parameter		
SoConIPduIdentifier.pduTriggering		
Mapping Rule		**Mapping Type**
The SoAdPduRoute container shall be created for every PduTriggering that is transmitted by the regarded ECU. The information whether the Pdu is received or transmitted over a Socket Connection shall be derived from the PduTriggering element. The PduTriggering element contains references to IPduPorts of an EcuInstance. The IPduPort element contains a communication Direction.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SoAd_00007]

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter		**BSW Type**
SoAdPduRouteDest		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Specifies the PDU route destination.		
Template Description		

▽

△

ApplicationEndpoint: An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication. **StaticSocketConnection.iPduIdentifier:** Assignment of IPduIdentifiers that are transmitted over the static SocketConnection.	
M2 Parameter	
ApplicationEndpoint, StaticSocketConnection.iPduIdentifier	
Mapping Rule	**Mapping Type**
SOME/IP communication: If the regarded EcuInstance that transmits the PduTriggering is referenced by an ApplicationEndpoint that in turn is referenced by a ProvidedServiceInstance that has a relationship to the PduTriggering via a PduActivationRoutingGroup and this Application Endpoint contains a maxNumberOfConnections setting then this container shall be created as many times as the maxNumberOfConnections value indicates. StaticSocketConnection: If the PduTriggering that is transmitted by the Ecu has a relationship to a StaticSocketConnection via a SoConIPduIdentifier then this container shall be created once for this PduTriggering.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00119]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest
BSW Parameter	**BSW Type**
SoAdTxPduHeaderId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID to be sent on the TCP/IP connection if the PDU header option is enabled.	
Template Description	
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.	
M2 Parameter	
SoConIPduIdentifier.headerId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00120]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest
BSW Parameter	**BSW Type**
SoAdTxRoutingGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the routing group.	
Template Description	
Group of Pdus that can be activated or deactivated for transmission over a socket connection.	
M2 Parameter	
PduActivationRoutingGroup	
Mapping Rule	**Mapping Type**

▽

△

If the PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPdu Identifier that in turn has a relationship to a PduActivationRoutingGroup then this reference shall point to the SoAdRoutingGroup that is derived from the PduActivationRoutingGroup.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00123]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest
BSW Parameter	**BSW Type**
SoAdTxSocketConnOrSocketConnBundleRef	ECUC-CHOICE-REFERENCE-DEF
BSW Description	
Choice Reference to a SocketConnection or to a SocketConnectionGroup on which the PDU is to be sent on. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API.	
Template Description	
StaticSocketConnection: Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address. **ProvidedServiceInstance.localUnicastAddress:** The local address over which the PSI is provided (udp, tcp or both). **ConsumedServiceInstance.localUnicastAddress:** The local address over which the CSI is consumed (udp, tcp or both).	
M2 Parameter	
StaticSocketConnection, ProvidedServiceInstance.localUnicastAddress, ConsumedServiceInstance.localUnicastAddress	
Mapping Rule	**Mapping Type**
SOME/IP Communication: The PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPduIdentifier that in turn has a relationship to a PduActivationRouting Group. The PduActivationRoutingGroup is aggregated either by a ProvidedServiceInstance, a ConsumedServiceInstance, or an EventHandler. In the first two cases, the local address is determined by ProvidedServiceInstance.localUnicastAddress and ConsumedServiceInstance.local UnicastAddress. For EventHandlers the local address is determined from the local Address of the ProvidedServiceInstance which contains the EventHandler. StaticSocketConnection: The PduTriggering for which the SoAdPduRoute is created is referenced by a SoConIPduIdentifier that in turn has a relationship to a StaticSocketConnection. The static So AdSocketConnection is aggregated by a SocketAddress for which the SoAdSocketConnection Group was created. If the SocketConnectionGroup contains two or more SocketConnections and the PDU shall be sent on all of them or the PDU shall be received from any of them, set the reference to the SoAdSocketConnectionGroup container. If the PDU shall be sent or received via exactly one SocketConnection of the SocketConnectionGroup, reference the corresponding SoAd SocketConnection container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00034]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest
BSW Parameter	**BSW Type**
SoAdTxUdpTriggerMode	ECUC-ENUMERATION-PARAM-DEF
BSW Description	

▽

△

Specifies whether a PDU triggers the transmission of the nPduUdpTxBuffer. If this parameter is set to TRIGGER_NEVER, So Ad shall use an nPduUdpTxBuffer for the related socket connection. nPduUdpTxBuffer can only be used for upper layers with IF API, i.e. this parameter shall only be set to TRIGGER_NEVER if all upper layers belonging to the related socket connection have SoAdTxUpperLayerType set to "IF". This parameter is only relevant for UDP connections.	
Template Description	
Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.	
M2 Parameter	
SoConIPduIdentifier.pduCollectionTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00136]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest
BSW Parameter	**BSW Type**
SoAdTxUdpTriggerTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the timeout in [s] the nPduUdpTxBuffer shall be transmitted at the latest after this PDU is put into the buffer. This optional parameter is only relevant if SoAdTxUdpTriggerMode is TRIGGER_NEVER.	
Template Description	
Defines the timeout in seconds the PDU collection shall be transmitted at the latest after this PDU has been put into the buffer.	
M2 Parameter	
SoConIPduIdentifier.pduCollectionPduTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00150]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	**BSW Type**
SoAdTxPduCollectionSemantics	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies if this PDU shall be collected using a queued or last-is-best semantics. This parameter is only relevant if the PDU collection feature is enabled. Shall only be set to SOAD_COLLECT_LAST_IS_BEST if the related upper layer is configured with SoAdIfTriggerTransmit set to TRUE.	
Template Description	
Specifies if the referenced PduTriggering shall be collected using a queued (i.e. all PDU instances) or last-is-best (i.e. only the last PDU instance) semantics. If this attribute is not present the behavior of "queued" is assumed.	
M2 Parameter	
SoConIPduIdentifier.pduCollectionSemantics	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00160]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdTxPduCollectionSemantics
BSW Parameter	**BSW Type**
SOAD_COLLECT_LAST_IS_BEST	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The PDU data will be fetched via <Up>_[SoAd][If]TriggerTransmit just before the transmission executes.	
Template Description	
Only the latest PDU instances are transmitted.	
M2 Parameter	
PduCollectionSemanticsEnum.lastIsBest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdTxPduCollectionSemantics
BSW Parameter	**BSW Type**
SOAD_COLLECT_QUEUED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
The PDU data will instantly be stored in the context of the SoAd_IfTransmit API.	
Template Description	
All instances of PDUs are transmitted.	
M2 Parameter	
PduCollectionSemanticsEnum.queued	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	**BSW Type**
SoAdTxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global PDU structure	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SoConIPduIdentifier.pduTriggering	
Mapping Rule	**Mapping Type**
This SoAdTxPduRef reference shall be derived from the PduTriggering that is referenced by the SoConIPduIdentifier.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00030]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdPduRoute
BSW Parameter	**BSW Type**
SoAdTxUpperLayerType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies the upper layer interface type (must be "IF" in case of multiple PduRoutes).	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
The SoAdTxUpperLayerType parameter can be derived from the actual type of the PDU: DcmIPdu -> "Tp" UUDT DcmIPdu: "If" (according to [SWS_Dcm_01101]) ISignalIPdu -> "If" if the Pdu Triggering refering to the ISignalIPdu is NOT referenced by a TpConnection, "Tp" if the Pdu Triggering refering to the ISignalIPdu is referenced by a TpConnection. NmPdu -> "If" General PurposePdu with category SD -> "If" GeneralPurposePdu with category DoIP -> "If" for UDP, "Tp" for TCP GeneralPurposePdu with category GLOBAL_TIME -> "If" GeneralPurposeIPdu with category = XCP -> "If" UserDefinedIPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP UserDefinedPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP MultiplexedIPdu: "If" ContainerIPdu: "If" SecuredIPdu: "If" (see limitation in AUTOSAR_SWS_SecureOnboard Communication)	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00118]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig
BSW Parameter	**BSW Type**
SoAdRoutingGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Each container describes a specific routing group which can be enabled or disabled. A routing group consists of PDUs. Routing of PDUs can either be forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a SoAdPduRoute or the other way around specified by a SoAdSocketRoute.	
Template Description	
Group of Pdus that can be activated or deactivated for transmission over a socket connection.	
M2 Parameter	
PduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
The SoAdRoutingGroup container shall be created for every PduActivationRoutingGroup element that is available in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00109]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup
BSW Parameter	**BSW Type**
SoAdRoutingGroupTxTriggerable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if the If-TxPDUs related to the PduRouteDest containers referenced by this routing group can be triggered via So Ad_IfRoutingGroupTransmit (TRUE) or not (FALSE).	

△

Template Description	
This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.	
M2 Parameter	
PduActivationRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
The SoAdRoutingGroupTxTriggerable parameter shall be derived from the eventGroupControlType enumeration. If • PduActivationRoutingGroup is aggregated by an EventHandler in the role pduActivationRouting Group and • eventGroupControlType equals triggerUnicast or activationAndTriggerUnicast then SoAdRoutingGroupTxTriggerable shall be set to true. In all other cases SoAdRoutingGroupTx Triggerable shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00146]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig
BSW Parameter	**BSW Type**
SoAdSocketConnectionGroup	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies the configuration of a socket connection group, i.e. specifies the socket connections belonging to the group and the parameters which are common for all socket connections of the group. A socket connection specifies how data can be received and transmited via a TCP or UDP socket.	
Template Description	
ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both). **ConsumedServiceInstance.localUnicastAddress:** The local address over which the CSI is consumed (udp, tcp or both). **StaticSocketConnection:** Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.	
M2 Parameter	
ProvidedServiceInstance.localUnicastAddress, ConsumedServiceInstance.localUnicastAddress, StaticSocketConnection	
Mapping Rule	**Mapping Type**

▽

△

Unicast SocketConnectionGroup rules: - SOME/IP Communication without configured remote UnicastAddress: Collect all ProvidedServiceInstances and ConsumedServiceInstances that are pointing to the same localUnicastAddress of the regarded Ecu and create a SoAdConnection Group for the referenced SocketAddress. Each relevant SocketAddress aggregates the referenced ApplicationEndpoint and points with the connector reference to the Ecu for which the Ecu Configuration is created. - SOME/IP Communication with configured remoteUnicastAddress: Collect all ProvidedService Instances and ConsumedServiceInstances that are pointing to the same localUnicastAddress and to the same remoteUnicastAddress and create a SoAdConnectionGroup for the referenced Socket Address. Each relevant SocketAddress aggregates the referenced ApplicationEndpoint that is referenced by the localUnicastAddress and points with the connector reference to the Ecu for which the Ecu Configuration is created. - StaticSocketConnection: Create a SocketConnectionGroup for each StaticSocketConnection that is aggregated by an ApplicationEndpoint of the regarded Ecu or points with the remoteUnicast Address to an ApplicationEndpoint of the regarded Ecu. Multicast SocketConnectionGroup rules: - Server side: If a ProvidedServiceInstance points with the localUnicastAddress to an ApplicationEndpoint that in turn is aggregated by a SocketAddress that refers the regarded EcuInstance and this ProvidedServiceInstance contains an EventHandler that contains an eventMulticastAddress reference then a multicast SocketConnectionGroup shall be created. The local Address of the SocketConnectionGroup shall be the address that is derived from the ProvidedServiceInstance.localUnicast reference. The remote Address of the Socket ConnectionGroup shall be the multicastAddress that is derived from EventHandler.eventMulticast Address. - Client side: If a ConsumedServiceInstance points with the localUnicastAddress to an Application Endpoint that in turn is aggregated by a SocketAddress that refers the regarded EcuInstance and this ConsumedServiceInstance contains a ConsumedEventGroup that contains an eventMulticast Ref then a multicast SocketConnectionGroup shall be created. The local Address of the Socket ConnectionGroup shall be derived from the ApplicationEndpoint that is referenced by the event MulticastAddress reference. The remote Address shall be derived from the ApplicationEndpoint that is referenced by ProvidedServiceInstance.localUnicastAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00130]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdPduHeaderEnable	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables the transmission of the PDU header (ID, length) on this socket connection. TRUE: add SoAd PDU header before PDU data FALSE: No SoAd PDU header is used	
Template Description	
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.	
M2 Parameter	
`SoConIPduIdentifier.headerId`	
Mapping Rule	**Mapping Type**
If a SoConIPduIdentifier element has a headerId set to a value and this SoConIPduIdentifier is referenced by a PduActivationGroup that is aggregated by a relevant ProvidedServiceInstance/ ConsumedServiceInstance/EventHandler/ConsumedEventGroup then this parameter shall be set to true. If a StaticSocketConnection refers to a SoConIPduIdentifier element that has a headerId set to a value then this parameter shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00131]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup

BSW Parameter	BSW Type
SoAdSocketConnection	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Specifies the socket connection (Id and remote address information). Note: Parameters which are common to all socket connections of a socket connection group are specified directly at the group.

Template Description

ApplicationEndpoint.maxNumberOfConnections:
This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.

StaticSocketConnection:
Definition of static SocketConnection between the Socket that is defined by the aggregating SocketAddress and the remote Address.

ProvidedServiceInstance.remoteMulticastSubscriptionAddress:
This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.

ProvidedServiceInstance.remoteUnicastAddress:
This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.

EventHandler.eventMulticastAddress:
Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the mulicastThreshold is exceeded.

This address is transmitted in the SD-SubscribeEventGroupAck Message to client (answer to SD-SubscribeEventGroup).

M2 Parameter

`ApplicationEndpoint.maxNumberOfConnections`, `StaticSocketConnection`, `ProvidedServiceInstance.remoteMulticastSubscriptionAddress`, `ProvidedServiceInstance.remoteUnicastAddress`, `EventHandler.eventMulticastAddress`

Mapping Rule	Mapping Type
The SoAdSocketConnectionGroup is created for an ApplicationEndpoint aggregated by a Socket Address that contains a connector reference to the regarded Ecu. This SoAdSocketConnection Group defines the context for the creation of the following SoAdSocketConnections. Unicast SoAdSocketConnections based on maxNumberOfConnections: The relevant Application Endpoint is referenced by the localUnicastAddress reference from ProvidedServiceInstances/ ConsumedServiceInstances or by the eventMulticastSubscriptionAddress from ConsumedService Instance. The relevant ApplicationEndpoint has the attribute maxNumberOfConnections that defines how many unicast SocketConnections inside the SoAdSocketConnectionGroup need to be created. If this attribute is not set then only a single unicast SoAdSocketConnection shall be created unless the ProvidedServiceInstance/ConsumedServiceInstance refer to statically known remote Addresses via remoteUnicastAddress and/or remoteMulticastSubscriptionAddress. The remotePort of each created SoAdSocketConnection shall be set to 0 and the IP Address to ANY. Unicast and multicast SoAdSocketConnections based on remoteUnicastAddress and/or remote MulticastSubscriptionAddress: The relevant ApplicationEndpoint is referenced by the localUnicast Address reference from ProvidedServiceInstances/ConsumedServiceInstances. For each ProvidedServiceInstance/ConsumedServiceInstance that has a remoteUnicastAddress and/or remoteMulticastSubscriptionAddress defined one single SoAdSocketConnection shall be defined. The remotePort of each created SoAdSocketConnection shall be derived from the Application Endpoint/NetworkEndpoint combination that is referenced by the remoteUnicastAddress / remote MulticastSubscriptionAddress. Multicast SoAdSocketConnection: For each unique (wrt. the multicast address and port values) ApplicationEndpoint that points to a multicast NetworkEndpoint that is referenced by an Event Handler.eventMulticastAddress that in turn is contained by a ProvidedServiceInstances which references the ApplicationEndpoint associated with the SoAdSocketConnectionGroup create one multicast SoAdSocketConnection. StaticSocketConnection: If the SocketConnectionBundle was created for a SocketAddress containing one or more StaticSocketConnections, one SoAdSocketConnection container shall be created for each of the contained StaticSocketConnections.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00009]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection

BSW Parameter	BSW Type
SoAdSocketRemoteAddress	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Subcontainer of SoAdSocketConnection to specify the remote address (IP address and port) for a socket connection. If SoAd SocketRemoteAddress is not specified the remote address has to be set by the upper layer via SoAd_SetRemoteAddr().

Template Description

ConsumedServiceInstance.remoteUnicastAddress:
This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery.

ProvidedServiceInstance.remoteUnicastAddress:
This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.

StaticSocketConnection.remoteAddress:
RemoteAddress of the static SocketConnection.

ProvidedServiceInstance.remoteMulticastSubscriptionAddress:
This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.

M2 Parameter

`ConsumedServiceInstance.remoteUnicastAddress`, `ProvidedServiceInstance.remoteUnicastAddress`, `StaticSocketConnection.remoteAddress`, `ProvidedServiceInstance.remoteMulticastSubscriptionAddress`

Mapping Rule	Mapping Type
SOME/IP Communication without configured remoteUnicastAddress and remoteMulticast SubscriptionAddress: If neither remoteUnicastAddress nor remoteMulticastSubscriptionAddress is used it means that the remoteUnicastAddress is derived via Service Discovery at runtime. In this case the remote Port shall be set to 0 and the IP Address to ANY. SOME/IP Communication with configured remoteUnicastAddress and/or remoteMulticast SubscriptionAddress: If the remoteUnicastAddress and/or remoteMulticastSubscriptionAddress is used the remoteAddressPort and the remote IP Address shall be derived from the referenced ApplicationEndpoint. StaticSocketConnection: In case of the StaticSocketConnection the remoteAddressPort and IP Address shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn is referenced by the StaticSocketConnection.remoteAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00113]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemote Address

BSW Parameter	BSW Type
SoAdSocketRemoteIpAddress	ECUC-STRING-PARAM-DEF

BSW Description

IP address of remote node. The configured address must be of the same TcpIpDomainType (i.e. IPv4 or IPv6) as the TcpIp LocalAddr referred by SoAdSocketLocalAddressRef . To accept any remote IP address, set SoAdSocketRemoteIpAddress to "ANY". See message acceptance policy for more details.

Template Description

▽

△

ProvidedServiceInstance.remoteUnicastAddress: This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **ConsumedServiceInstance.remoteUnicastAddress:** This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. **StaticSocketConnection.remoteAddress:** RemoteAddress of the static SocketConnection. **ProvidedServiceInstance.remoteMulticastSubscriptionAddress:** This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.
M2 Parameter
`ProvidedServiceInstance.remoteUnicastAddress`, `ConsumedServiceInstance.remoteUnicastAddress`, `StaticSocketConnection.remoteAddress`, `ProvidedServiceInstance.remoteMulticastSubscriptionAddress`

Mapping Rule	**Mapping Type**
SOME/IP Communication without configured remoteUnicastAddress and remoteMulticast SubscriptionAddress: If neither remoteUnicastAddress nor remoteMulticastSubscriptionAddress is used it means that the remoteAddress is derived via ServiceDiscovery at runtime. In this case this attribute shall be set to ANY. SOME/IP Communication with configured remoteUnicastAddress and/or remoteMulticast SubscriptionAddress: If the remoteUnicastAddress is used the remoteAddress IP Address shall be derived from the referenced ApplicationEndpoint that in turn refers a NetworkEndpoint that contains the searched IP Address. StaticSocketConnection: In case of the StaticSocketConnection the remoteAddress shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn is referenced by the StaticSocketConnection.remoteAddress. The ApplicationEndpoint refers to a NetworkEndpoint that contains the searched IP Address.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00019]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemote Address

BSW Parameter	**BSW Type**
SoAdSocketRemotePort	ECUC-INTEGER-PARAM-DEF
BSW Description	
Remote UDP or TCP port used for this connection. To accept any remote port, set SoAdSocketRemotePort to 0. See message acceptance policy for more details.	
Template Description	
ProvidedServiceInstance.remoteUnicastAddress: This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **ConsumedServiceInstance.remoteUnicastAddress:** This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. **StaticSocketConnection.remoteAddress:** RemoteAddress of the static SocketConnection. **ProvidedServiceInstance.remoteMulticastSubscriptionAddress:** This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime.	
M2 Parameter	

▽

△

ProvidedServiceInstance.remoteUnicastAddress, ConsumedServiceInstance.remoteUnicastAddress, StaticSocketConnection.remoteAddress, ProvidedServiceInstance.remoteMulticastSubscriptionAddress	
Mapping Rule	**Mapping Type**
SOME/IP Communication without configured remoteUnicastAddress and remoteMulticastSubscriptionAddress: If neither remoteUnicastAddress nor remoteMulticastSubscriptionAddress is used it means that the remotePort is derived via ServiceDiscovery at runtime.In this case this attribute shall be set to 0. SOME/IP Communication with configured remoteUnicastAddress and/or remoteMulticastSubscriptionAddress: If the remoteUnicastAddress and/or remoteMulticastSubscriptionAddress is used the remoteAddressPort shall be derived from the referenced ApplicationEndpoint. StaticSocketConnection: In case of the StaticSocketConnection the remoteAddressPort shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn is referenced by the StaticSocketConnection.remoteAddress.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00020]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketDifferentiatedServicesField	ECUC-INTEGER-PARAM-DEF
BSW Description	
The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.	
Template Description	
The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.	
M2 Parameter	
SocketAddress.differentiatedServiceField	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00158]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketFlowLabel	ECUC-INTEGER-PARAM-DEF
BSW Description	
The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.	
Template Description	
The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.	
M2 Parameter	
SocketAddress.flowLabel	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00157]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketFramePriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the priority of the Ethernet frame. If IEEE 802.1Q VLAN Tags are used, the specified priority will be used in the VLAN Tag PCP filed. If this optional parameter is not available the default priority specified in the TcpIp module is used.	
Template Description	
VlanMembership.defaultPriority: Standard output-priority outgoing Frames will be tagged with. Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed. In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified. **ConsumedEventGroup.priority:** Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed. **NetworkEndpoint:** The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address). **ApplicationEndpoint.priority:** Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed. **ProvidedServiceInstance.priority:** Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.	
M2 Parameter	
VlanMembership.defaultPriority, ConsumedEventGroup.priority, NetworkEndpoint, ApplicationEndpoint.priority, ProvidedServiceInstance.priority	
Mapping Rule	**Mapping Type**
It shall be considered that the priority in the SoAd is defined only once per SocketConnection Group. The SocketConnections in the system description shall be created adequate. The priority in the system description can be defined at the ConsumedEventGroup, Provided ServiceInstance, at the ApplicationEndpoint and at the NetworkEndpoint. A default priority can be set at CouplingPort.VlanMembership. The priority defined on the ProvidedServiceInstance or ConsumedEventGroup overrides the priority defined on the ApplicationEndpoint. The priority on the ApplicationEndpoint overrides the priority on the NetworkEndpoint. The priority on the Network Endpoint overrides the default priority.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00138]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketLocalAddressRef	ECUC-REFERENCE-DEF
BSW Description	
Local IP address and interface used for this connection.	
Template Description	

▽

△

ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both). **ConsumedServiceInstance.remoteUnicastAddress:** This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. **SocketAddress.staticSocketConnection:** Definition of a static SocketConnection.
M2 Parameter
ProvidedServiceInstance.localUnicastAddress, ConsumedServiceInstance.remoteUnicastAddress, SocketAddress.staticSocketConnection

Mapping Rule	Mapping Type
The SoAdSocketLocalAddressRef shall be derived from the attributes IPv4Configuration.ipv4 Address or IPv6Configuration.ipv6Address from the NetworkEndpoint that is referenced by the ApplicationEndpoint that is aggregated by the SocketAddress referenced from the ProvidedService Instance, ConsumedServiceInstance or that aggregates the StaticSocketConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00017]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketLocalPort	ECUC-INTEGER-PARAM-DEF
BSW Description	
Local UDP or TCP port used for this connection. If this parameter set to 0 SoAd requests TcpIp to select an ephemeral port.	
Template Description	
ProvidedServiceInstance.localUnicastAddress: The local address over which the PSI is provided (udp, tcp or both). **ConsumedServiceInstance.localUnicastAddress:** The local address over which the CSI is consumed (udp, tcp or both). **SocketAddress.staticSocketConnection:** Definition of a static SocketConnection.	
M2 Parameter	
ProvidedServiceInstance.localUnicastAddress, ConsumedServiceInstance.localUnicastAddress, SocketAddress.staticSocketConnection	

Mapping Rule	Mapping Type
The SoAdSocketLocalPort shall be derived from the value of the attribute TpPort.portNumber in the ApplicationEndpoint referenced from ProvidedServiceInstance/ConsumedServiceInstance in the role localUnicastAddress. In case of a StaticSocketConnection the local port shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress that in turn aggregates the StaticSocketConnection.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00018]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol
BSW Parameter	**BSW Type**
SoAdSocketTcp	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies that TCP is used as transport protocol for the socket connection group and parameters only related to TCP socket connections.	

▽

△

Template Description	
An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.	
M2 Parameter	
ApplicationEndpoint	
Mapping Rule	**Mapping Type**
Server Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ProvidedServiceInstance in the role localUnicastAddress. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol. Client Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ConsumedServiceInstance in the role localUnicastAddress. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00141]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpAutoConnectTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the time in seconds how long 'TCP connect attempts are repeated to reach SOAD_SOCON_ONLINE. This parameter is restricted to socket connection groups which are initiating a TCP connection and are under control of SoAd.	
Template Description	
Specifies the time in seconds how long TCP connect attempts are repeated to reach SOAD_SOCON_ONLINE. This attribute is restricted to socket connection groups which are initiating a TCP connection and are under control of SoAd.	
M2 Parameter	
StaticSocketConnection.tcpConnectTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00174]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpInitiate	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies the initiator for this TCP connection. It will not be defined for UDP sockets. TRUE: This TCP connection is initiated by this module. FALSE: This TCP connection is to be initiated in the listen mode.	
Template Description	
Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.	
M2 Parameter	
StaticSocketConnection.tcpRole	
Mapping Rule	**Mapping Type**

▽

△

SOME/IP Communication: - Server Ecu: SoAdSocketTcpInitiate can be set to false here since Servers do not initiate Tcp connections by themselves. - Client Ecu: SoAdSocketTcpInitiate can be set to true if TcpTp is configured and the Ecu is in the client role. StaticSocketConnection: If StaticSocketConnection.tcpRole = connect then this attribute shall be set to TRUE. If StaticSocketConnection.tcpRole = listen then this attribute shall be set to TRUE.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00022]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpKeepAlive	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies to use the keep-alive mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will use the keep-alive mechanism. FALSE: This TCP connection will not use the keep-alive mechanism. Note: This parameter must not be set to TRUE if TcpIpTcpKeepAliveEnabled is set to FALSE.	
Template Description	
Indicates if Keep-Alive messages are sent.	
M2 Parameter	
TcpTp.keepAlives	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00148]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpKeepAliveInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the interval in seconds between subsequent keepalive probes.	
Template Description	
Specifies the interval in seconds between subsequent keepalive probes.	
M2 Parameter	
TcpTp.keepAliveInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00152]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpKeepAliveProbesMax	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Maximum number of times that TCP retransmits an individual data segment before aborting the connection.	
Template Description	
Maximum number of times that TCP retransmits an individual data segment before aborting the connection.	
M2 Parameter	
TcpTp.keepAliveProbesMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00151]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpKeepAliveTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the time in seconds between the last data packet sent and the first keepalive probe.	
Template Description	
Specifies the time in seconds between the last data packet sent and the first keepalive probe.	
M2 Parameter	
TcpTp.keepAliveTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00153]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpNoDelay	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies not to use the congestion control mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will NOT use congestion control. FALSE: This TCP connection will use congestion control. If the optional parameter is not enabled, the default behavior configured for TcpIp via the parameter TcpIpTcpNagleEnabled is applied. Note: This parameter must not be set to FALSE if TcpIpTcpNagleEnabled is set to FALSE.	
Template Description	
Indicates if Nagle's Algorithm is used.	
M2 Parameter	
TcpTp.naglesAlgorithm	
Mapping Rule	**Mapping Type**
If TcpTp.naglesAlgorithm in the System Extract is set to true then SoAdSocketTcpNoDelay shall be set to false. If TcpTp.naglesAlgorithm in is set to false then SoAdSocketTcpNoDelay shall be set to true. If TcpTp.naglesAlgorithm in the System Extract is not defined then SoAdSocketTcpNoDelay shall not be set.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00023]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpRetransmissionTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in [s] before an unacknowledged TCP segment is sent again. This parameter overrules TcpIpTcpRetransmission Timeout for this SoAdSocketConnectionGroup. If this parameter is enabled together with a maximum like TcpIpTcpMax RetransmissionTimeout or SoAdSocketTcpMaxRetransmissionTimeout then SoAdSocketTcpRetransmissionTimeout is considered as initial value for first retransmission before the next valid acknowledgment arrives.	
Template Description	
Defines the timeout in seconds before an unacknowledged TCP segment is sent again. If the tcpRetransmissionTimeout is not defined or set to "INF", no TCP segments shall be re-transmitted.	
M2 Parameter	
TcpTp.tcpRetransmissionTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00171]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpTlsConnectionRef	ECUC-REFERENCE-DEF
BSW Description	
If set the TCP socket is assigned to a TLS connection. The SoAd need to call Tcpip_ChangeParameter with the reference to the TLS connection as the parameter.	
Template Description	
This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS).	
M2 Parameter	
TlsCryptoServiceMapping	
Mapping Rule	**Mapping Type**
Create this reference if there is an associated TcpIpTlsConnection derived out of the TlsCypto ServiceMapping.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00163]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol
BSW Parameter	**BSW Type**
SoAdSocketUdp	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies that UDP is used as transport protocol for the socket connection group and parameters only related to UDP socket connections.	
Template Description	
An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.	

▽

△

M2 Parameter	
ApplicationEndpoint	
Mapping Rule	**Mapping Type**
Server Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ProvidedServiceInstance in the role localUnicastAddress. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol. Client Ecu: SoAdSocketTcp shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced by a ConsumedServiceInstance in the role localUnicastAddress. SoAdSocketTcp shall be created if TcpTp is used as Transport Protocol.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00140]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp
BSW Parameter	**BSW Type**
SoAdSocketUdpChecksumEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if UDP checksum calculation shall be enabled (TRUE) or skipped (FALSE) on the related socket. FALSE implies that the upper layer of the socket connection is either capable to handle malformed messages or applies a checksum mechanism itself.	
Template Description	
Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksumDisabled) on the related socket connection.	
M2 Parameter	
SocketAddress.udpChecksumHandling	
Mapping Rule	**Mapping Type**
If udpChecksumHandling.udpChecksumEnabled is set the value shall be TRUE; if udpChecksum Handling.udpChecksumDisabled is set the value shall be FALSE;	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00159]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp
BSW Parameter	**BSW Type**
SoAdSocketUdpTriggerTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the timeout in [s] a nPduUdpTxBuffer is waiting for a PDU with TriggerMode = TRIGGER_ALWAYS, i.e. when the timeout expires the nPduUdpTxBuffer is transmitted. Timer is reset after each UDP transmission. This optional parameter is only relevant if a nPduUdpTxBuffer is used.	
Template Description	
Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.	
M2 Parameter	
SocketAddress.pduCollectionTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00133]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp
BSW Parameter	**BSW Type**
SoAdSocketnPduUdpTxBufferMin	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the amount of data in bytes (PDU data provided by the upper layer and PDU Header if used) the SoAd shall be able to buffer for data transmission via this socket connection in case the UDP message shall be buffered for transmission of multiple PDUs per UDP. Note: in case of a UDP socket and an upper layer with TP API is configured, the required buffer size can be determined automatically. This optional parameter is only relevant if a nPduUdpTxBuffer is used.	
Template Description	
Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.	
M2 Parameter	
SocketAddress.pduCollectionMaxBufferSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00135]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketTimeToLive	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines a value set in the header of an Internet Protocol (IP) packet that tells network devices the maximum number of router hops the packet can make before it is discarded. The TTL value is a counter that is decremented by 1 every time the packet passes through a router.	
Template Description	
This attribute defines a value set in the header of an Internet Protocol (IP) packet that tells network devices the maximum number of router hops the packet can make before it is discarded. The TTL value is a counter that is decremented by 1 every time the packet passes through a router.	
M2 Parameter	
SocketAddress.ttl	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00178]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketTpRxBufferMin	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Specifies the amount of data in bytes (PDU data for the upper layer and PDU Header if used) the SoAd shall at least be able to buffer for data reception via each socket connection of the socket connection group and using an upper layer with TP. Note: in case of a TCP socket where PduHeaderMode is used and an upper layer with IF-API, the required buffer size can be determined automatically.	
Template Description	
Minimum size of the TCP receive window in bytes.	
M2 Parameter	
TcpTp.receiveWindowMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00134]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig
BSW Parameter	**BSW Type**
SoAdSocketRoute	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the path of a PDU from a socket in the TCP/IP stack to an upper layer of the SoAd after reception in the TCP/IP Stack.	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SoConIPduIdentifier.pduTriggering	
Mapping Rule	**Mapping Type**
The SoAdSocketRoute container shall be created for every PduTriggering that is received by the regarded ECU. The information whether the Pdu is received or transmitted over a Socket Connection shall be derived from the PduTriggering element. The PduTriggering element contains references to IPduPorts of an EcuInstance. The IPduPort element contains a communication Direction. If the PduTriggering is referenced by a SoConIPduIdentifier that in turn is referenced by a PduActivationRoutingGroup with eventGroupControlType set to activationUnicastAndMulticast, then two SoAdSocketRoute instances have to be created, one for unicast and another for multicast.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00008]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketRoute
BSW Parameter	**BSW Type**
SoAdRxPduHeaderId	ECUC-INTEGER-PARAM-DEF
BSW Description	
ID contained in the packet received on the TCP/IP connection if the PDU header option is enabled.	
Template Description	
If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.	
M2 Parameter	
SoConIPduIdentifier.headerId	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00036]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketRoute
BSW Parameter	**BSW Type**
SoAdRxSocketConnOrSocketConnBundleRef	ECUC-CHOICE-REFERENCE-DEF

BSW Description

Choice Reference to a SocketConnection or to a SocketConnectionGroup on which the PDU was received. The reference to a SocketConnectionGroup shall only be used for upper layers with IF API.

Template Description

SocketAddress.staticSocketConnection:
Definition of a static SocketConnection.

ConsumedServiceInstance.localUnicastAddress:
The local address over which the CSI is consumed (udp, tcp or both).

ProvidedServiceInstance.localUnicastAddress:
The local address over which the PSI is provided (udp, tcp or both).

ConsumedEventGroup.eventMulticastAddress:
This reference defines the multicast address or a multicast address resource where the events of the event group are received.

If the multicast address is determined via configuration and not at runtime via service discovery this reference points to the multicast address over which the events will be received.

If the multicast address is determined at runtime via service discovery this reference shall be used to define the necessary local multicast address resources, i.e. RAM space in the TcpIp module in which the multicast address is stored at runtime. Please note that in this case the referenced address may be defined as ANY UDP port and ANY IP address since the multicast address will be received at runtime. If several multicast addresses are considered to be used the ConsumedEvent Group shall point to different ApplicationEndpoint objects to reserve the necessary resources in the configuration.

M2 Parameter

`SocketAddress.staticSocketConnection`, `ConsumedServiceInstance.localUnicastAddress`, `ProvidedServiceInstance.localUnicastAddress`, `ConsumedEventGroup.eventMulticastAddress`

Mapping Rule	**Mapping Type**
SOME/IP Communication: The PduTriggering for which the SoAdSocketRoute is created is referenced by a SoConIPduIdentifier that in turn has a relationship to a PduActivationRouting Group. The PduActivationRoutingGroup is aggregated either by a ProvidedServiceInstance, a ConsumedServiceInstance, or a ConsumedEventGroup. In the first two cases, the local address is determined by ProvidedServiceInstance.localUnicastAddress and ConsumedServiceInstance.local UnicastAddress. For PduActivationRoutingGroups of ConsumedEventGroups that have the event GroupControlType set to "activationUnicast", "triggerUnicast" or "activationAndTriggerUnicast" the local address is determined by ConsumedServiceInstance.localUnicastAddress of the Consumed ServiceInstance containing the ConsumedEventGroup. In this case the SoAdTxSocketConnOr SocketConnBundleRef shall refer to the SocketConnectionGroup container that was created for the determined local address. For PduActivationRoutingGroups of ConsumedEventGroups that have the eventGroupControlType set to "activationMulticast", the local address is determined by ConsumedEventGroup.eventMulticastAddress. For PduActivationRoutingGroups of Consumed EventGroups that have the eventGroupControlType set to "activationUnicastAndMulticast", two So AdRxSocketConnOrSocketConnBundleRef instances shall be created (see also SoAdSocket Route): One SoAdRxSocketConnOrSocketConnBundleRef shall refer to the SocketConnection Group container that was created for the ConsumedServiceInstance.localUnicastAddress. The second SoAdTxSocketConnOrSocketConnBundleRef shall refer to the SocketConnectionGroup container that was created for the ConsumedEventGroup.eventMulticastAddress. The reference parameter shall refer to the SocketConnectionGroup container that was created for the determined local address.	full

▽

▽

△

△ StaticSocketConnection: The PduTriggering for which the SoAdSocketRoute is created is referenced by a SoConIPduIdentifier that in turn has a relationship to a StaticSocketConnection. The StaticSocketConnection is aggregated by a SocketAddress for which the SocketConnection Group was created. If the SocketConnectionGroup contains two or more SocketConnections and the PDU shall be sent on all of them or the PDU shall be received from any of them, set the reference to the SoAdSocketConnectionGroup container. If the PDU shall be sent or received via exactly one SocketConnection of the SocketConnectionGroup, reference the corresponding SoAd SocketConnection container.	
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00035]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketRoute
BSW Parameter	**BSW Type**
SoAdSocketRouteDest	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Describes the upper layer destination PDU for a message received on a TcpIp socket. This PDU can produce meta data items of type SOCKET_CONNECTION_ID_16. Multiple socket route destinations in the SoAdSocketRoute can only be used for upper layers of interface type (IF) and only for SoAdSocketRoute referring a SocketConnectionGroup. In this case SoAdRoutingGroups shall be used to map each SoAd SocketRouteDest uniquely to different socket connections of the SocketConnectionGroup.	
Template Description	
Identification of Pdu content on a socket connection. This Identifier is required in case that multiple Pdus are transmitted over the same socket connection.	
M2 Parameter	
SoConIPduIdentifier	
Mapping Rule	**Mapping Type**
The SoAdSocketRouteDest container shall always be created for a SoAdSocketRoute.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00114]

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest
BSW Parameter	**BSW Type**
SoAdRxPduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the global PDU structure	
Template Description	
Reference to a Pdu that is transmitted over a socket connection.	
M2 Parameter	
SoConIPduIdentifier.pduTriggering	
Mapping Rule	**Mapping Type**
The SoAdRxPduRef reference shall be derived from the PduTriggering that is referenced by the So ConIpduIdentifier.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00038]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest
BSW Parameter	**BSW Type**
SoAdRxRoutingGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to the routing group. Mandatory if the parent SoAdSocketRoute contains more than one SoAdSocketRouteDest.	
Template Description	
Group of Pdus that can be activated or deactivated for transmission over a socket connection.	
M2 Parameter	
PduActivationRoutingGroup	
Mapping Rule	**Mapping Type**
The SoAdRxRoutingGroupRef reference to SoAdRoutingGroups shall be derived from the Pdu ActivationRoutingGroup that refers the SoConIPduIdentifer.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00117]

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest
BSW Parameter	**BSW Type**
SoAdRxUpperLayerType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies the upper layer interface type (must be "IF" in case of multiple RxPdus).	
Template Description	
M2 Parameter	
PduTriggering.iPdu	
Mapping Rule	**Mapping Type**
The SoAdRxUpperLayerType parameter can be derived from the actual type of the PDU: DcmIPdu -> "Tp" UUDT DcmIPdu: "If" (according to [SWS_Dcm_01101]) ISignalIPdu -> "If" if ComIPdu Type=NORMAL, "Tp" if ComIPduType=TP NmPdu -> "If" GeneralPurposePdu with category SD -> "If" GeneralPurposePdu with category DoIP -> "If" for UDP, "Tp" for TCP GeneralPurposePdu with category GLOBAL_TIME -> "If" GeneralPurposeIPdu with category = XCP -> "If" UserDefined IPdu: "If" if CddPduRApiType=IF, "Tp" if CddPduRApiType=TP UserDefinedPdu: "If" if CddPdu RApiType=IF, "Tp" if CddPduRApiType=TP MultiplexedIPdu: "If" ContainerIPdu: "If" SecuredIPdu: "If" (see limitation in AUTOSAR_SWS_SecureOnboardCommunication)	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SoAd_00115]

C.54 SomeIpTp

BSW Module	BSW Context
SomeIpTp	SomeIpTp
BSW Parameter	**BSW Type**
SomeIpTpRxChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This container contains the configuration parameters of the SomeIpTp reception channel.	
Template Description	
A connection identifies the sender and the receiver of this particular communication. The SOME/IP TP module routes a Pdu through this connection.	
M2 Parameter	
SomeipTpConnection	
Mapping Rule	**Mapping Type**
Create an instance for each SomeipTpConnection where the referenced PduTriggering is received by the EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_00026]

BSW Module	**BSW Context**
SomeIpTp	SomeIpTp/SomeIpTpRxChannel
BSW Parameter	**BSW Type**
SomeIpTpRxNPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the configuration parameters of the NPdu that is received from a lower layer	
Template Description	
Reference to the segmented IPdu.	
M2 Parameter	
SomeipTpConnection.transportPdu	
Mapping Rule	**Mapping Type**
Create container if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is received by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_00011]

BSW Module	**BSW Context**
SomeIpTp	SomeIpTp/SomeIpTpRxChannel
BSW Parameter	**BSW Type**
SomeIpTpRxNSdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the upper layer Sdus assembled in the context of the SomeIpTpRxChannel. This PDU can forward meta data items of type SOCKET_CONNECTION_ID_16.	
Template Description	
SomeipTpConnection.tpSdu: Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tpConcurrentProcessing Sdu references shall be used. **SomeipTpConnection.tpConcurrentProcessingSdu:** Reference to a set of IPdus that are used to implement concurrent processing of several upper layer IPdus related to one transportPdu by the SOME/IP TP segmenter.	
M2 Parameter	
SomeipTpConnection.tpSdu, SomeipTpConnection.tpConcurrentProcessingSdu	
Mapping Rule	**Mapping Type**

▽

△

Create this reference for every SomeipTpConnection.tpSdu and SomeipTpConnection.tp ConcurrentProcessingSdu reference where the referenced PduTriggering is received by the Ecu Instance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_ 00008]

BSW Module	**BSW Context**
SomeIpTp	SomeIpTp/SomeIpTpRxChannel/SomeIpTpRxNSdu

BSW Parameter	**BSW Type**
SomeIpTpRxSduRef	ECUC-REFERENCE-DEF

BSW Description
Reference to a Pdu in the COM-Stack that represents the assembled RxPdu which is passed via the PduR to the upper layer.
Template Description
Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tpConcurrentProcessing Sdu references shall be used.
M2 Parameter
SomeipTpConnection.tpSdu

Mapping Rule	**Mapping Type**
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role tpSdu that is received by the EcuInstance that is contained in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_ 00010]

BSW Module	**BSW Context**
SomeIpTp	SomeIpTp/SomeIpTpRxChannel

BSW Parameter	**BSW Type**
SomeIpTpRxTimeoutTime	ECUC-FLOAT-PARAM-DEF

BSW Description
Timer to monitor the successful reception (see FO_PRS_SOMEIP_00378). It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received. The value shall be calculated as follows: (SomeIpTpRxTimeoutTime = SomeIpTpNPduSeparationTime + budget), where the time budget compensates intermediary hops and jitters within the ECU implementation.
Template Description
Timer to monitor the successful reception. It is started when the first NPdu is received, restarted after reception of intermediate NPdus, and is stopped when the last NPdu has been received.
M2 Parameter
SomeipTpChannel.rxTimeoutTime

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_ 00023]

BSW Module	BSW Context	
SomeIpTp	SomeIpTp	
BSW Parameter		**BSW Type**
SomeIpTpTxChannel		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration parameters of the SomeIpTp transmission channel.		
Template Description		
A connection identifies the sender and the receiver of this particular communication. The SOME/IP TP module routes a Pdu through this connection.		
M2 Parameter		
SomeipTpConnection		
Mapping Rule		**Mapping Type**
Create an instance for each SomeipTpConnection where the referenced PduTriggering is transmitted by the EcuInstance.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SomeIpTp_00025]

BSW Module	BSW Context	
SomeIpTp	SomeIpTp/SomeIpTpTxChannel	
BSW Parameter		**BSW Type**
SomeIpTpNPduSeparationTime		ECUC-FLOAT-PARAM-DEF
BSW Description		
Sets the duration of the minimum time in seconds the SomeIpTp module shall wait between the transmissions of N-PDUs.		
Template Description		
Sets the duration of the minimum time in seconds the SOME/IP TP module shall wait between the transmissions of NPdus.		
M2 Parameter		
SomeipTpChannel.separationTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SomeIpTp_00006]

BSW Module	BSW Context	
SomeIpTp	SomeIpTp/SomeIpTpTxChannel	
BSW Parameter		**BSW Type**
SomeIpTpTxBurstSize		ECUC-INTEGER-PARAM-DEF
BSW Description		
Specifies the number of segments SomeIpTp shall transmit without applying the SomeIpTpNPduSeparationTime.		
Template Description		
Specifies the number of segments that shall be transmitted in a burst ignoring separationTime. SeparationTime will then only be applied between bursts. If not configured, SeparationTime will be applied between all frames.		
M2 Parameter		
SomeipTpChannel.burstSize		
Mapping Rule		**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_00024]

BSW Module	**BSW Context**	
SomeIpTp	SomeIpTp/SomeIpTpTxChannel	
BSW Parameter		**BSW Type**
SomeIpTpTxNPdu		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container contains the configuration parameters of the segmented Tx NPdus that are transmitted to a lower layer.		
Template Description		
Reference to the segmented IPdu.		
M2 Parameter		
SomeipTpConnection.transportPdu		
Mapping Rule		**Mapping Type**
Create container if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SomeIpTp_00016]

BSW Module	**BSW Context**	
SomeIpTp	SomeIpTp/SomeIpTpTxChannel/SomeIpTpTxNPdu	
BSW Parameter		**BSW Type**
SomeIpTpTxNPduRef		ECUC-REFERENCE-DEF
BSW Description		
Reference to a global Pdu that is used to harmonize HandleIDs in the COM-Stack.		
Template Description		
Reference to the segmented IPdu.		
M2 Parameter		
SomeipTpConnection.transportPdu		
Mapping Rule		**Mapping Type**
Create reference if a SomeipTpConnection exists that points to a PduTriggering in the role transportPdu that is transmitted by the EcuInstance that is contained in the Ecu Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SomeIpTp_00018]

BSW Module	**BSW Context**	
SomeIpTp	SomeIpTp/SomeIpTpTxChannel	
BSW Parameter		**BSW Type**
SomeIpTpTxNSdu		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		

▽

△

This container defines the upper layer Sdus to be transmitted in the context of the SomeIpTpTxChannel. This PDU can forward meta data items of type SOCKET_CONNECTION_ID_16.	
Template Description	
SomeipTpConnection.tpSdu: Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tpConcurrentProcessing Sdu references shall be used. **SomeipTpConnection.tpConcurrentProcessingSdu:** Reference to a set of IPdus that are used to implement concurrent processing of several upper layer IPdus related to one transportPdu by the SOME/IP TP segmenter.	
M2 Parameter	
SomeipTpConnection.tpSdu, SomeipTpConnection.tpConcurrentProcessingSdu	
Mapping Rule	**Mapping Type**
Create this container for every SomeipTpConnection.tpSdu and SomeipTpConnection.tp ConcurrentProcessingSdu reference where the referenced PduTriggering is transmitted by the Ecu Instance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_ 00009]

BSW Module	**BSW Context**
SomeIpTp	SomeIpTp/SomeIpTpTxChannel/SomeIpTpTxNSdu
BSW Parameter	**BSW Type**
SomeIpTpTxNSduRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a global Pdu in the COM-Stack that represents the original TxSdu which is segmented and passed via the Pdu R to the lower layer.	
Template Description	
SomeipTpConnection.tpSdu: Reference to an IPdu that is segmented by the SOME/IP Transport Protocol. This reference defines the non concurrent processing of that IPdu by the SOME/IP TP segmenter. If concurrent processing is required, then tpConcurrentProcessing Sdu references shall be used. **SomeipTpConnection.tpConcurrentProcessingSdu:** Reference to a set of IPdus that are used to implement concurrent processing of several upper layer IPdus related to one transportPdu by the SOME/IP TP segmenter.	
M2 Parameter	
SomeipTpConnection.tpSdu, SomeipTpConnection.tpConcurrentProcessingSdu	
Mapping Rule	**Mapping Type**
Create this reference for every SomeipTpConnection.tpSdu and SomeipTpConnection.tp ConcurrentProcessingSdu reference where the referenced PduTriggering is transmitted by the Ecu Instance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SomeIpTp_ 00015]

C.55 StbM

BSW Module	BSW Context
StbM	StbM/StbMFreshnessValueInformation
BSW Parameter	**BSW Type**
StbMFreshnessValue	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Container with the Freshness Value configurations	
Template Description	
Reference to a SecureComProps definition to be used for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvSecureComProps	
Mapping Rule	**Mapping Type**
For each referenced SecOcSecureComProps in the role GlobalTimeDomain.icvSecureComProps create one instance of this container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00082]

BSW Module	BSW Context
StbM	StbM/StbMFreshnessValueInformation/StbMFreshnessValue
BSW Parameter	**BSW Type**
StbMFreshnessValueId	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.	
Template Description	
This attribute defines the Id of the Freshness Value for the Integrity Check Value (ICV) calculation and verification.	
M2 Parameter	
GlobalTimeDomain.icvFreshnessValueId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00083]

BSW Module	BSW Context
StbM	StbM/StbMFreshnessValueInformation/StbMFreshnessValue
BSW Parameter	**BSW Type**
StbMFreshnessValueLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.	
Template Description	
This attribute defines the complete length in bits of the Freshness Value.	
M2 Parameter	
SecOcSecureComProps.freshnessValueLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_StbM_00084]

BSW Module	BSW Context	
StbM	StbM/StbMFreshnessValueInformation/StbMFreshnessValue	
BSW Parameter		**BSW Type**
StbMFreshnessValueTruncLength		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the length in bits of the Freshness Value to be included in the payload of the Secured Time Synchronization Messages. This length is specific to the least significant bits of the complete Freshness Counter. If the parameter is 0 no Freshness Value is included in the Secured Time Synchronization Messages.		
Template Description		
This attribute defines the length in bits of the Freshness Value to be included in the payload of the secured message.		
M2 Parameter		
SecOcSecureComProps.freshnessValueTxLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_StbM_00085]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		**BSW Type**
StbMClearTimeleapCount		ECUC-INTEGER-PARAM-DEF
BSW Description		
This attribute describes the required number of updates to the Time Base where the time difference to the previous value has to remain below StbMTimeLeapPastThreshold/StbMTimeLeapFutureThreshold until the TIMELEAP_PAST/TIMELEAP_FUTURE bit within timeBaseStatus of the Time Base is cleared.		
Template Description		
Defines the required number of updates to the Time Base where the time difference to the previous received value has to remain within the bounds of timeLeapFutureThreshold and timeLeapPastThreshold until that Time Base is considered healed.		
M2 Parameter		
GlobalTimeSlave.timeLeapHealingCounter		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_StbM_00037]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		**BSW Type**
StbMIsSystemWideGlobalTimeMaster		ECUC-BOOLEAN-PARAM-DEF
BSW Description		

▽

△

This parameter shall be set to true for a Global Time Master that acts as a system-wide source of time information with respect to Global Time. It is possible that several Global Time Masters exist that have set this parameter set to true because the Global Time Masters exist once per Global Time Domain and one ECU may own several Global Time Domains on different buses it is connected to.	
Template Description	
If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.	
M2 Parameter	
GlobalTimeMaster.isSystemWideGlobalTimeMaster	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00036]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase
BSW Parameter	**BSW Type**
StbMMaxProgressionMismatchThreshold	ECUC-FLOAT-PARAM-DEF
BSW Description	
This represents the maximum allowed difference between local time and fallback time of the time base [unit: seconds].	
Template Description	
This attribute defines the maximum allowed difference between local time and fallback time of the time base in seconds.	
M2 Parameter	
GlobalTimeDomain.maxProgressionMismatchThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_StbM_00088]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase
BSW Parameter	**BSW Type**
StbMSyncLossTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain. Unit: seconds	
Template Description	
This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain.	
M2 Parameter	
GlobalTimeDomain.syncLossTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00028]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase
BSW Parameter	**BSW Type**
StbMSynchronizedTimeBaseType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Definition of the type of a Time Base.	
Template Description	
This represents the ability to define a global time domain.	
M2 Parameter	
GlobalTimeDomain	
Mapping Rule	**Mapping Type**
Partially derived from GlobalTimeDomain.category: • category = SYNCHRONIZED -> TBTYPE_SYNCHRONIZED TBTYPE_PURELOCAL is not derived out of the system description.	partial
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_StbM_00100]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMAllowMasterRateCorrection	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
This attribute describes whether the rate correction value of a Time Base can be set by StbM_SetRateCorrection() or by a clone operation: • false: the rate correction value can not be set • true: the rate correction value can be set	
Template Description	
Defines whether the rate correction value of a Time Base can be set by means of the method setRateCorrection(). false: rate correction cannot be set by method setRateCorrection(). true: rate correction can be set by method setRateCorrection().	
M2 Parameter	
GlobalTimeCorrectionProps.allowProviderRateCorrection	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00043]

BSW Module	BSW Context
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMMasterRateDeviationMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
This attribute describes the maximum allowed absolute value of the rate deviation value to be set by StbM_SetRate Correction() or by a clone operation [unit: ppm].	
Template Description	
This attribute describes the maximum allowed absolute value of the rate deviation value [unit: ppm].	
M2 Parameter	

▽

△

GlobalTimeCorrectionProps.providerRateDeviationMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00044]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMOffsetCorrectionAdaptionInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation [unit: seconds].	
Template Description	
Defines the interval during which the adaptive rate correction cancels out the rate- and time deviation.	
M2 Parameter	
GlobalTimeCorrectionProps.offsetCorrectionAdaptionInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00057]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMOffsetCorrectionJumpThreshold	ECUC-FLOAT-PARAM-DEF
BSW Description	
Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump [unit: seconds].	
Template Description	
Threshold for the correction method. Deviations below this value will be corrected by a linear reduction over a defined timespan. Values equal- and greater than this value will be corrected by immediately setting the correct time- and rate in form of a jump.	
M2 Parameter	
GlobalTimeCorrectionProps.offsetCorrectionJumpThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00056]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMRateCorrectionMeasurementDuration	ECUC-FLOAT-PARAM-DEF
BSW Description	

▽

△

Definition of the time span [s] which is used to calculate the rate deviation.	
Template Description	
Definition of the time span which is used to calculate the rate deviation.	
M2 Parameter	
GlobalTimeCorrectionProps.rateCorrectionMeasurementDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00054]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMRateCorrectionThreshold	ECUC-INTEGER-PARAM-DEF
BSW Description	
Threshold for rate correction calculation to determine whether the absolute value of the calculated rate deviation exceeds the accepted range. A value of 0 deactivates the check.	
Template Description	
This attribute defines the threshold for rate correction calculation to determine whether the absolute value of the calculated rate deviation exceeds the accepted range. The value is given in ppm. A value of 0 deactivates the check.	
M2 Parameter	
GlobalTimeCorrectionProps.rateCorrectionThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00099]

BSW Module	**BSW Context**
StbM	StbM/StbMSynchronizedTimeBase/StbMTimeCorrection
BSW Parameter	**BSW Type**
StbMRateCorrectionsPerMeasurementDuration	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of simultaneous rate measurements to determine the current rate deviation.	
Template Description	
Defines the number of simultaneous rate measurements to determine the current rate deviation.	
M2 Parameter	
GlobalTimeCorrectionProps.rateCorrectionsPerMeasurementDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_StbM_00055]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		**BSW Type**
StbMTimeLeapFutureThreshold		ECUC-FLOAT-PARAM-DEF
BSW Description		
This represents the maximum allowed positive difference between a newly received Global Time Base value and the current Local Time Base value [unit: seconds].		
Template Description		
Defines the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value.		
M2 Parameter		
GlobalTimeSlave.timeLeapFutureThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_StbM_00041]

BSW Module	BSW Context	
StbM	StbM/StbMSynchronizedTimeBase	
BSW Parameter		**BSW Type**
StbMTimeLeapPastThreshold		ECUC-FLOAT-PARAM-DEF
BSW Description		
This represents the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value [unit: seconds].		
Template Description		
Defines the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value.		
M2 Parameter		
GlobalTimeSlave.timeLeapPastThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_StbM_00042]

C.56 SwCluC

BSW Module	BSW Context	
SwCluC	SwCluC/SwCluCDefinitionSet/SwCluCDefinition	
BSW Parameter		**BSW Type**
SwCluCMachineId		ECUC-INTEGER-PARAM-DEF
BSW Description		
Unique number of the (virtual or physical) machine to which the Software Cluster belongs.		
Template Description		
Unique number of the (virtual or physical) machine to which the Software Cluster is mapped.		

▽

△

M2 Parameter	
CpSoftwareClusterToEcuInstanceMapping.machineId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_SwCluC_00008]

BSW Module	BSW Context	
SwCluC	SwCluC/SwCluCDefinitionSet/SwCluCDefinition	
BSW Parameter		**BSW Type**
SwCluCSoftwareClusterType		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
The type of the Software Cluster		
Template Description		
This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings.		
M2 Parameter		
CpSoftwareCluster		
Mapping Rule		**Mapping Type**
If CpSoftwareCluster.category == HOST_SOFTWARE_CLUSTER the SwCluCSoftwareCluster Type shall be set to HOST_SW_CLUSTER If CpSoftwareCluster.category == APPLICATION_SOFTWARE_CLUSTER the SwCluCSoftware ClusterType shall be set to APPLICATION_SW_CLUSTER		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_SwCluC_00009]

C.57 TcpIp

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl	
BSW Parameter		**BSW Type**
TcpIpIpFramePrioDefault		ECUC-INTEGER-PARAM-DEF
BSW Description		
Specifies the default value for the priority for all outgoing frames. Note: the value can be changed for each socket individually via TcpIp_ChangeParameter() service. If this optional parameter is not available, 0 is used as default priority.		
Template Description		
Standard output-priority outgoing Frames will be tagged with. Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed. In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.		
M2 Parameter		

▽

△

VlanMembership.defaultPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00081]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig
BSW Parameter	**BSW Type**
TcpIpDhcpAddressAssignment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Defines a Ethernet Switch port based IP address assignment.	
Template Description	
Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddressAssignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.	
M2 Parameter	
VlanMembership.dhcpAddressAssignment	
Mapping Rule	**Mapping Type**
The existence of a dhcpAddressAssignment leads to one container.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00191]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig/TcpIpDhcpAddressAssignment
BSW Parameter	**BSW Type**
TcpIpDhcpAddressLowerBound	ECUC-STRING-PARAM-DEF
BSW Description	
The lower bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned.	
Template Description	
Ipv4DhcpServerConfiguration.addressRangeLowerBound: Lower range of IP addresses to be issued to DHCP clients. IPv4 Address. Notation: 255.255.255.255. **Ipv6DhcpServerConfiguration.addressRangeLowerBound:** Lower range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.	
M2 Parameter	
Ipv4DhcpServerConfiguration.addressRangeLowerBound, Ipv6DhcpServerConfiguration.addressRangeLowerBound	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00193]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig/TcpIpDhcpAddressAssignment
BSW Parameter	**BSW Type**

▽

△

TcpIpDhcpAddressUpperBound	ECUC-STRING-PARAM-DEF
BSW Description	
The upper bound IP address which shall be assigned. If lower bound and upper bound are identical exactly this IP address shall be assigned.	
Template Description	
Ipv4DhcpServerConfiguration.addressRangeUpperBound: Upper range of IP addresses to be issued to DHCP clients. Pv4 Address. Notation: 255.255.255.255. **Ipv6DhcpServerConfiguration.addressRangeUpperBound:** Upper range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.	
M2 Parameter	
Ipv4DhcpServerConfiguration.addressRangeUpperBound, Ipv6DhcpServerConfiguration.addressRangeUpperBound	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00194]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig/TcpIpDhcpAddressAssignment
BSW Parameter	**BSW Type**
TcpIpDhcpSwitchPortRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to Ethernet Switch port. Optional in case the Dhcp server is operating without an Ethernet switch.	
Template Description	
A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.	
M2 Parameter	
CouplingPort	
Mapping Rule	**Mapping Type**
The Switch CouplingPort the VlanMembership.dhcpAddressAssignment belongs to.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00192]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig
BSW Parameter	**BSW Type**
TcpIpDhcpDefaultRouter	ECUC-STRING-PARAM-DEF
BSW Description	
IP address of default router (gateway).	
Template Description	
Ipv4DhcpServerConfiguration.defaultGateway: IP address of the default gateway. Notation 255.255.255.255 **Ipv6DhcpServerConfiguration.defaultGateway:** IP address of the default gateway. Notation 255.255.255.255	
M2 Parameter	

▽

△

Ipv4DhcpServerConfiguration.defaultGateway, Ipv6DhcpServerConfiguration.defaultGateway	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00190]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig
BSW Parameter	**BSW Type**
TcpIpDhcpEthIfSwitchRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to EthIfSwitch representation. Optional in case the Dhcp server is operating without an Ethernet switch.	
Template Description	
Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddressAssignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.	
M2 Parameter	
VlanMembership.dhcpAddressAssignment	
Mapping Rule	**Mapping Type**
The Switch CouplingElement the VlanMembership.dhcpAddressAssignment belongs to.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00188]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpDhcpServerConfig
BSW Parameter	**BSW Type**
TcpIpDhcpNetmask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.	
Template Description	
Ipv4DhcpServerConfiguration.networkMask: Default network mask to be used by DHCP clients. Notation 255.255.255.255 **Ipv6DhcpServerConfiguration.networkMask:** Default network mask to be used by DHCP clients. Notation 255.255.255.255	
M2 Parameter	
Ipv4DhcpServerConfiguration.networkMask, Ipv6DhcpServerConfiguration.networkMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00189]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**

▽

△

TcpIpArpDefensiveProcessing	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the ARP shall only process ARP replies which are received in reaction to a previously transmitted ARP request as well as skipping updates to the ARP table based on received Gratuitous ARP packets. If disabled all ARP packets shall be processed as specified in IETF RFC 826.	
Template Description	
This attribute enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.	
M2 Parameter	
Ipv4ArpProps.tcpIpArpPacketQueueEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00326]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**
TcpIpArpNumGratuitousARPonStartup	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.	
Template Description	
This attribute specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.	
M2 Parameter	
Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00054]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**
TcpIpArpPacketQueueEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.	
Template Description	
This attribute enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.	
M2 Parameter	
Ipv4ArpProps.tcpIpArpPacketQueueEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00170]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**
TcpIpArpRequestTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the TcpIp shall skip the transmission of any further ARP requests to the same destination within a duration of TcpIpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1) The value for this parameter shall be an integral multiple of TcpIpMainFunctionPeriod or 0. If this parameter set to 0 this features is disabled and no delay between ARP requests is enforced.	
Template Description	
This attribute specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the TcpIp shall skip the transmission of any further ARP requests to the same destination within a duration of tcpIpArpRequest Timeout seconds. (IETF RFC 1122, section 2.3.2.1).	
M2 Parameter	
`Ipv4ArpProps.tcpIpArpRequestTimeout`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00218]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**
TcpIpArpTableEntryTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in seconds after which an unused ARP entry is removed.	
Template Description	
This attribute specifies the timeout in seconds after which an unused ARP entry is removed.	
M2 Parameter	
`Ipv4ArpProps.tcpIpArpTableEntryTimeout`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00053]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpArpConfig
BSW Parameter	**BSW Type**
TcpIpArpTableSizeMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of entries in the ARP table.	
Template Description	
This attribute specifies the size of neighbor cache or ARP table in units of entries.	
M2 Parameter	
`EthernetCommunicationConnector.neighborCacheSize`	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00052]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpAutoIpConfig
BSW Parameter	**BSW Type**
TcpIpAutoIpInitTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
The time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.	
Template Description	
This attribute specifies the time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.	
M2 Parameter	
Ipv4AutoIpProps.tcpIpAutoIpInitTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00074]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIcmpConfig
BSW Parameter	**BSW Type**
TcpIpIcmpEchoReplyEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.	
Template Description	
This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.	
M2 Parameter	
TcpIpIcmpv4Props.tcpIpIcmpV4EchoReplyEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00213]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIcmpConfig
BSW Parameter	**BSW Type**
TcpIpIcmpTtl	ECUC-INTEGER-PARAM-DEF
BSW Description	
Default Time-to-live value of outgoing ICMP packets.	
Template Description	

▽

△

This attribute is only relevant in case that ICMP (Internet Control Message Protocol) is used. It specifies the default Time-to-live value of outgoing ICMP packets.	
M2 Parameter	
TcpIpIcmpv4Props.tcpIpIcmpV4Ttl	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00055]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpFragmentationRxEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).	
Template Description	
Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).	
M2 Parameter	
Ipv4FragmentationProps.tcpIpIpFragmentationRxEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00077]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpNumFragments	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the maximum number of IP fragments per datagram. Note: this parameter is only relevant if TcpIpIpFragmentation RxEnabled is TRUE.	
Template Description	
Specifies the maximum number of IP fragments per datagram.	
M2 Parameter	
Ipv4FragmentationProps.tcpIpIpNumFragments	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00078]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
BSW Parameter	**BSW Type**

▽

△

TcpIpIpNumReassDgrams	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel. Note: this parameter is only relevant if TcpIpIpFragmentationRxEnabled is TRUE.	
Template Description	
Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel.	
M2 Parameter	
Ipv4FragmentationProps.tcpIpIpNumReassDgrams	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00080]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV4Config/TcpIpIpFragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpReassTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the timeout in [s] after which an incomplete datagram gets discarded. Note: this parameter is only relevant if TcpIp IpFragmentationRxEnabled is TRUE.	
Template Description	
Specifies the timeout in [s] after which an incomplete datagram gets discarded.	
M2 Parameter	
Ipv4FragmentationProps.tcpIpIpReassTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00079]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6CnfDelayMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum delay (s) before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
Template Description	
Maximum delay in seconds before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6CnfDelayMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00116]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6CnfDelayMin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Minimum delay (s) before the first Confirm message will be sent.	
Template Description	
Minimum delay in seconds before the first Confirm message will be sent.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6CnfDelayMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00117]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6InfDelayMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum delay (s) before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
Template Description	
Maximum delay in seconds before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6InfDelayMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00118]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6InfDelayMin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Minimum delay (s) before the first Information Request message will be sent.	
Template Description	
Minimum delay (s) before the first Information Request message will be sent.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6InfDelayMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00119]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6SolDelayMax	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum delay (s) before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
Template Description	
Maximum delay in seconds before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6SolDelayMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00120]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpDhcpV6Config
BSW Parameter	**BSW Type**
TcpIpDhcpV6SolDelayMin	ECUC-FLOAT-PARAM-DEF
BSW Description	
Minimum delay (s) before the first Solicit message will be sent.	
Template Description	
Minimum delay (s) before the first Solicit message will be sent.	
M2 Parameter	
Dhcpv6Props.tcpIpDhcpV6SolDelayMin	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00121]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIcmpV6Config
BSW Parameter	**BSW Type**
TcpIpIcmpV6EchoReplyAvoidFragmentation	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled, the stack will respond only to incoming ICMPv6 Echo Requests (Pings) that fit the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation. Only relevant if TcpIpIcmpV6EchoReplyEnabled is enabled.	
Template Description	
This attribute defines whether the echo reply is only transmitted in case that the incoming ICMPv6 Echo Request (Pings) fits the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation.	
M2 Parameter	
TcpIpIcmpv6Props.tcpIpIcmpV6EchoReplyAvoidFragmentation	
Mapping Rule	**Mapping Type**
1:1 mapping	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00212]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIcmpV6Config
BSW Parameter	**BSW Type**
TcpIpIcmpV6EchoReplyEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled, the stack will respond to incoming ICMPv6 Echo Requests (Pings).	
Template Description	
This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.	
M2 Parameter	
TcpIpIcmpv6Props.tcpIpIcmpV6EchoReplyEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00149]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIcmpV6Config
BSW Parameter	**BSW Type**
TcpIpIcmpV6HopLimit	ECUC-INTEGER-PARAM-DEF
BSW Description	
Default Hop-Limit value of outgoing ICMPv6 packets.	
Template Description	
Default Hop-Limit value of outgoing ICMPv6 packets.	
M2 Parameter	
TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00152]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIcmpV6Config
BSW Parameter	**BSW Type**
TcpIpIcmpV6MsgDestinationUnreachableEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Dis/Enables transmission of Destination Unreachable Messages	
Template Description	
This attribute Enables/Disables the transmission of Destination Unreachable Messages.	
M2 Parameter	
TcpIpIcmpv6Props.tcpIpIcmpV6MsgDestinationUnreachableEnabled	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00153]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIcmpV6Config
BSW Parameter	**BSW Type**
TcpIpIcmpV6MsgParameterProblemEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet. [RFC8200 4. IPv6 Extension Headers]	
Template Description	
If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet.	
M2 Parameter	
TcpIpIcmpv6Props.tcpIpIcmpV6MsgParameterProblemEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00151]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config
BSW Parameter	**BSW Type**
TcpIpIpV6ConfigExtHeaderFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the white list for the filtering of IPv6 extension headers, i.e. frames containing IPv6 extension headers not listed here shall be silently dropped.	
Template Description	
Permitted list for the filtering of IPv6 extension headers.	
M2 Parameter	
IPv6ExtHeaderFilterList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00198]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6ConfigExtHeaderFilter
BSW Parameter	**BSW Type**
TcpIpIpV6ConfigExtHeaderFilterEntry	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

IPv6 Extension Header type allowed by this filter.	
Template Description	
IPv6 Extension Header type allowed by this filter.	
M2 Parameter	
IPv6ExtHeaderFilterList.allowedIPv6ExtHeader	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00199]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6ReassemblyBufferCount	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded. A value of 0 disables fragment reassembly. [RFC8200 5. Packet Size Issues] "In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets)."	
Template Description	
Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded. A value of 0 disables fragment reassembly.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00157]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6ReassemblyBufferSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
[RFC8200 5. Packet Size Issues] "A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. A node is permitted to accept fragmented packets that reassemble to more than 1500 octets."	
Template Description	
Size of each fragment tx buffer in bytes.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00158]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6ReassemblySegmentCount	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed. To deal with fragments received out of order this value should be configured bigger than 1.	
Template Description	
Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed. To deal with fragments received out of order this value should be configured bigger than 1.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpReassemblySegmentCount	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00160]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6ReassemblyTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
[RFC8200 4.5 Fragment Header] Default: 60 seconds	
Template Description	
Specifies the timeout in seconds after which an incomplete datagram gets discarded.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpReassemblyTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00159]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6TxFragmentBufferCount	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented. A value of 0 disables tx fragmentation. If the upper layer transmits packets that do not fit into the link or path MTU, the IpV6 will split-up the packet into fragments. see "Enable Fragment Reassembly"	
Template Description	
These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented. A value of 0 disables tx fragmentation.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00161]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpIpV6FragmentationConfig
BSW Parameter	**BSW Type**
TcpIpIpV6TxFragmentBufferSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Size of each fragment tx buffer in bytes	
Template Description	
Size of each fragment tx buffer in bytes.	
M2 Parameter	
Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00162]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpDefaultReachableTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables]. "The time a neighbor is considered reachable after receiving a reachability confirmation." If "TcpIpNdpDynamicReachableTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements. Default: REACHABLE_TIME = 30 seconds	
Template Description	
Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDefaultReachableTime	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00130]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpDefaultRetransTimer	ECUC-FLOAT-PARAM-DEF
BSW Description	
Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables]. "The time between retransmissions of Neighbor Solicitation messages to a neighbor when resolving the address or when probing the reachability of a neighbor." If "TcpIpNdpDynamicRetransTimeEnabled" is checked, this value may be reconfigured based on received Router Advertisements. Default: RETRANS_TIMER = 1 second	
Template Description	
Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].	
M2 Parameter	
`Ipv6NdpProps.tcpIpNdpDefaultRetransTimer`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00165]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpDefensiveProcessing	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861. [RFC4861 7.2.5. Receipt of Neighbor Advertisements]	
Template Description	
If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861.	
M2 Parameter	
`Ipv6NdpProps.tcpIpNdpDefensiveProcessing`	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00201]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpDelayFirstProbeTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Delay before sending the first NUD probe in (s). [RFC4861 7.3.3. Node Behavior] Default: DELAY_FIRST_PROBE_TIME = 5 seconds	
Template Description	
Delay before sending the first NUD probe in (s).	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00133]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpMaxNeighborCacheSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of entries in the neighbor cache. [RFC4861 5.1. Conceptual Data Structures]	
Template Description	
This attribute specifies the size of neighbor cache or ARP table in units of entries.	
M2 Parameter	
EthernetCommunicationConnector.neighborCacheSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00129]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpMaxRandomFactor	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum random factor used for randomization [RFC4861 10. Protocol Constants] Default: 15 (MAX_RANDOM_FACTOR = 1.5)	
Template Description	
Maximum random factor used for randomization	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpMaxRandomFactor	



△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00135]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpMinRandomFactor	ECUC-INTEGER-PARAM-DEF
BSW Description	
Minimum random factor used for randomization [RFC4861 10. Protocol Constants] Default: 5 (MIN_RANDOM_FACTOR = 0.5)	
Template Description	
Minimum random factor used for randomization	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpMinRandomFactor	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00134]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpNeighborUnreachabilityDetectionEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.	
Template Description	
Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpNeighborUnreachabilityDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00136]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpNumMulticastSolicitations	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Maximum number of multicast solicitations that will be sent when performing address resolution. [RFC4861 7.2.2. Sending Neighbor Solicitations] Default: MAX_MULTICAST_SOLICIT = 3	
Template Description	
Maximum number of multicast solicitations that will be sent when performing address resolution.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpNumMulticastSolicitations	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00132]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpNumUnicastSolicitations	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection. [RFC4861 7.3.3. Node Behavior] Default: MAX_UNICAST_SOLICIT = 3	
Template Description	
Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00131]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpPacketQueueEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.	
Template Description	
Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpPacketQueueEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00171]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpArNudConfig
BSW Parameter	**BSW Type**
TcpIpNdpRandomReachableTimeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time [RFC4861 6.3.2. Host Variables / ReachableTime]	
Template Description	
If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpRandomReachableTimeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00137]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpDefaultRouterListSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of default router entries. [RFC4861 5.1. Conceptual Data Structures]	
Template Description	
Maximum number of default router entries.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDefaultRouterListSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00139]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpDestinationCacheSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of entries in the destination cache. [RFC4861 5.1. Conceptual Data Structures]	
Template Description	
Maximum number of entries in the destination cache.	
M2 Parameter	



△

Ipv6NdpProps.tcpIpNdpDestinationCacheSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00138]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpDynamicHopLimitEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the default hop limit may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements]	
Template Description	
If enabled the default hop limit may be reconfigured based on received Router Advertisements.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDynamicHopLimitEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00147]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpDynamicMtuEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Allow dynamic reconfiguration of link MTU via Router Advertisements. [RFC4861 4.6.4. MTU]	
Template Description	
Allow dynamic reconfiguration of link MTU via Router Advertisements.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDynamicMtuEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00148]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**

▽

△

TcpIpNdpDynamicReachableTimeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements] Default: Enabled	
Template Description	
If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDynamicReachableTimeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00146]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpDynamicRetransTimeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements. [RFC4861 6.3.4. Processing Received Router Advertisements] Default: Enabled	
Template Description	
If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpDynamicRetransTimeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00145]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpMaxRtrSolicitationDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum delay before the first Router Solicitation will be sent after interface initialization in (s). [RFC4861 6.3.7. Sending Router Solicitations] Default: MAX_RTR_SOLICITATION_DELAY = 1 second	
Template Description	
Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpMaxRtrSolicitationDelay	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00143]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpMaxRtrSolicitations	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received. 0 = No Router Solicitations will be sent. This has no impact on handling Router Advertisements. [RFC4861 6.3.7. Sending Router Solicitations] Default: MAX_RTR_SOLICITATIONS = 3 transmissions	
Template Description	
Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00142]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpPrefixList	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies a list of prefixes to be treated as “on-link” according to IETF RFC 4861 Section 5.1.	
Template Description	
Internet Protocol version 6 (IPv6) configuration.	
M2 Parameter	
Ipv6Configuration	
Mapping Rule	**Mapping Type**
A distinct list of all prefixes used within the same local network shall be retrieved for the respective ECU configuration. This can be achieved by following all socket connections of this ECU, identify the communication partners and their NetworkEndPoint elements via the ApplicationEndpoint references, retrieve the prefixes of NetworkEndPoint/Ipv6Configuration/ipAddressPrefixLength and NetworkEndPoint/Ipv6 Configuration/ipv6Address and create a distinct list of them.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00205]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig/TcpIpNdpPrefixList
BSW Parameter	**BSW Type**
TcpIpNdpPrefixListEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Single entry in the prefix list.	
Template Description	
Internet Protocol version 6 (IPv6) configuration.	
M2 Parameter	
Ipv6Configuration	
Mapping Rule	**Mapping Type**
see upstream mapping in TcpIpNdpPrefixList	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00206]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig/TcpIpNdpPrefixList/TcpIpNdpPrefixListEntry
BSW Parameter	**BSW Type**
TcpIpNdpPrefixListEntryPrefixAddress	ECUC-STRING-PARAM-DEF
BSW Description	
The prefix of an IP address. This prefix can be used for on-link determination.	
Template Description	
Internet Protocol version 6 (IPv6) configuration.	
M2 Parameter	
Ipv6Configuration	
Mapping Rule	**Mapping Type**
see upstream mapping in TcpIpNdpPrefixList	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00208]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig/TcpIpNdpPrefixList/TcpIpNdpPrefixListEntry
BSW Parameter	**BSW Type**
TcpIpNdpPrefixListEntryPrefixLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
The number of leading bits in the Prefix that are valid.	
Template Description	
Internet Protocol version 6 (IPv6) configuration.	
M2 Parameter	
Ipv6Configuration	
Mapping Rule	**Mapping Type**
see upstream mapping in TcpIpNdpPrefixList	full



△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00207]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpPrefixListSize	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of entries in the on-link prefix list. [RFC4861 5.1. Conceptual Data Structures]	
Template Description	
Maximum number of entries in the on-link prefix list.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpPrefixListSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00140]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpRndRtrSolicitationDelayEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds. [RFC4861 6.3.7. Sending Router Solicitations] Default: Enabled	
Template Description	
If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpRndRtrSolicitationDelayEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00141]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpPrefixRouterDiscoveryConfig
BSW Parameter	**BSW Type**
TcpIpNdpRtrSolicitationInterval	ECUC-FLOAT-PARAM-DEF

▽

△

BSW Description	
Interval between consecutive Router Solicitations in (s). [RFC4861 6.3.7. Sending Router Solicitations] Default: RTR_SOLICITATION_INTERVAL = 4 seconds	
Template Description	
Interval between consecutive Router Solicitations in (s).	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00144]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig
BSW Parameter	**BSW Type**
TcpIpNdpSlaacDadNumberOfTransmissions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state. [RFC4861 5.1. Node Configuration Variables] Default: DupAddrDetectTransmits = 1 Setting this value to 0 turns off DAD.	
Template Description	
Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpSlaacDadNumberOfTransmissions	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00128]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig
BSW Parameter	**BSW Type**
TcpIpNdpSlaacDadRetransmissionDelay	ECUC-FLOAT-PARAM-DEF
BSW Description	
Sets the maximum value for the address configuration delay (s). According to [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] this value should be the same as MAX_RTR_SOLICITATION_DELAY. Default: MAX_RTR_SOLICITATION_DELAY = 1 second	
Template Description	
Sets the maximum value for the address configuration delay (s).	
M2 Parameter	

▽

△

Ipv6NdpProps.tcpIpNdpSlaacDadRetransmissionDelay	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00127]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig
BSW Parameter	**BSW Type**
TcpIpNdpSlaacDelayEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY]. "This serves to alleviate congestion when many nodes start up on the link at the same time, such as after a power failure, and may help to avoid race conditions when more than one node is trying to solicit for the same address at the same time." "The delay will avoid similar congestion when multiple nodes are going to configure addresses by receiving the same single multicast router advertisement." [RFC4861 5.4.2. Sending Neighbor Solicitation Messages] Default: True	
Template Description	
If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpSlaacDelayEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00125]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig/TcpIpIpV6Config/TcpIpNdpConfig/TcpIpNdpSlaacConfig
BSW Parameter	**BSW Type**
TcpIpNdpSlaacOptimisticDadEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.	
Template Description	
Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.	
M2 Parameter	
Ipv6NdpProps.tcpIpNdpSlaacOptimisticDadEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00126]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet
BSW Parameter	**BSW Type**
TcpIpSpdEntry	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Entry of the Security Policy Database (SPD).	
Template Description	
This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.	
M2 Parameter	
IPSecRule	
Mapping Rule	**Mapping Type**
Each SpdEntry shall be derived from the IPSecRules that are defined on a NetworkEndpoint that defines the local IP Address range.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00293]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpIpSecHeaderType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Header type specifying the IPsec security mechanism.	
Template Description	
Header type specifying the IPsec security mechanism.	
M2 Parameter	
IPSecRule.headerType	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00297]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecHeaderType
BSW Parameter	**BSW Type**
TCPIP_IPSEC_HDR_AH	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
AH (Authentication Header)	
Template Description	
Authentication Header (AH)	
M2 Parameter	
IPsecHeaderTypeEnum.ah	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecHeaderType	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_HDR_ESP		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
ESP (Encapsulating Security Payload)		
Template Description		
Encapsulating Security Payloads (ESP)		
M2 Parameter		
IPsecHeaderTypeEnum.esp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecHeaderType	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_HDR_NONE		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
None		
Template Description		
No header		
M2 Parameter		
IPsecHeaderTypeEnum.none		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry	
BSW Parameter		**BSW Type**
TcpIpIpSecPolicy		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Policy for usage of IPsec.		
Template Description		
An IPsec policy defines the rules that determine which type of IP traffic needs to be secured using IPsec and how that traffic is secured.		
M2 Parameter		
IPSecRule.policy		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00295]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecPolicy	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_POLICY_BYPASS		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Bypass (no IPSec processing)		
Template Description		
Signifying that no IPsec processing should be done at all.		
M2 Parameter		
IPsecPolicyEnum.passthrough		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecPolicy	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_POLICY_DISCARD		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Discard (drop IP datagram)		
Template Description		
Signifying that packets should be discarded		
M2 Parameter		
IPsecPolicyEnum.drop		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecPolicy	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_POLICY_OPTIONAL		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Optional		
Template Description		
Signifying that packets should be discarded and a diagnostic ICMP returned.		
M2 Parameter		
IPsecPolicyEnum.reject		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpIpSecPolicy	
BSW Parameter		**BSW Type**
TCPIP_IPSEC_POLICY_PROTECT		ECUC-ENUMERATION-LITERAL-DEF
BSW Description		
Protect (enable IPSec processing)		
Template Description		
Signifying that packets should be protected.		
M2 Parameter		
IPsecPolicyEnum.ipsec		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry	
BSW Parameter		**BSW Type**
TcpIpIpSecPriority		ECUC-INTEGER-PARAM-DEF
BSW Description		
Priority of the SPD entry. The processing of entries is based on priority, starting with the highest priority "0". The first matching SPD entry defines the policy.		
Template Description		
This attribute defines the priority of the IPSecRule (SPD entry). The processing of entries is based on priority, starting with the highest priority "0".		
M2 Parameter		
IPSecRule.priority		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00296]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry	
BSW Parameter		**BSW Type**
TcpIpLocalIpAddrEnd		ECUC-STRING-PARAM-DEF
BSW Description		
End value of the remote IP address range.		
Template Description		
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).		
M2 Parameter		
NetworkEndpoint		
Mapping Rule		**Mapping Type**

▽

△

The local IP address range shall be derived from the NetworkEndpoint that aggregates the IPSec Config and the collection of IPSecRules from which the Spd entries are derived. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.network Mask.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00301]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpLocalIpAddrStart	ECUC-STRING-PARAM-DEF
BSW Description	
Start value of the local IP address range.	
Template Description	
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).	
M2 Parameter	
NetworkEndpoint	
Mapping Rule	**Mapping Type**
The local IP address range shall be derived from the NetworkEndpoint that aggregates the IPSec Config and the collection of IPSecRules from which the Spd entries are derived. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.network Mask.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00300]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpLocalPortRangeEnd	ECUC-INTEGER-PARAM-DEF
BSW Description	
End value of the local port range.	
Template Description	
This attribute restricts the traffic monitoring and defines an end value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
IPSecRule.localPortRangeEnd	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00299]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**

▽

△

TcpIpLocalPortRangeStart	ECUC-INTEGER-PARAM-DEF
BSW Description	
Start value of the local port range.	
Template Description	
This attribute restricts the traffic monitoring and defines a start value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
IPSecRule.localPortRangeStart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00298]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpProtocol	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Relevant IP protocol. Note: As specified in IETF Rfc 4301 section 6, ICMP error messages will always be BYPASSed. The policy for TCPIP_PROTOCOL_ICMP only applies to ICMP non-error messages. (Echo reply/response).	
Template Description	
This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.	
M2 Parameter	
IPSecRule.ipProtocol	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00306]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpProtocol
BSW Parameter	**BSW Type**
TCPIP_PROTOCOL_ANY	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Any	
Template Description	
ANY protocol	
M2 Parameter	
IPsecIpProtocolEnum.any	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpProtocol	
BSW Parameter	**BSW Type**	
TCPIP_PROTOCOL_ICMP	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
ICMP		
Template Description		
Internet Control Message Protocol (ICMP)		
M2 Parameter		
IPsecIpProtocolEnum.icmp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpProtocol	
BSW Parameter	**BSW Type**	
TCPIP_PROTOCOL_TCP	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
TCP		
Template Description		
TCP Protocol		
M2 Parameter		
IPsecIpProtocolEnum.tcp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry/TcpIpProtocol	
BSW Parameter	**BSW Type**	
TCPIP_PROTOCOL_UDP	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
UDP		
Template Description		
UDP Protocol		
M2 Parameter		
IPsecIpProtocolEnum.udp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpRemIpAddrEnd	ECUC-STRING-PARAM-DEF
BSW Description	
End value of the remote IP address range.	
Template Description	
Definition of the remote NetworkEndpoint. With this reference the connection between the local NetworkEndpoint and the remote NetworkEndpoint is described on which the traffic is monitored.	
M2 Parameter	
IPSecRule.remoteIpAddress	
Mapping Rule	**Mapping Type**
The remote IP address range shall be derived from the NetworkEndpoints that are referenced by IPSecRule.remoteIpAddress. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00303]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpRemIpAddrStart	ECUC-STRING-PARAM-DEF
BSW Description	
Start value of the remote IP address range.	
Template Description	
Definition of the remote NetworkEndpoint. With this reference the connection between the local NetworkEndpoint and the remote NetworkEndpoint is described on which the traffic is monitored.	
M2 Parameter	
IPSecRule.remoteIpAddress	
Mapping Rule	**Mapping Type**
The remote IP address range shall be derived from the NetworkEndpoints that are referenced by IPSecRule.remoteIpAddress. Please note that the networkEndpointAddress is allowed to define a subnet with the Ipv4Configuration.networkMask.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00302]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpRemPortRangeEnd	ECUC-INTEGER-PARAM-DEF
BSW Description	
End value of the remote port range.	
Template Description	

▽

△

This attribute restricts the traffic monitoring and defines an end value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
IPSecRule.remotePortRangeEnd	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00305]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpIpSecConfigSet/TcpIpSpdEntry
BSW Parameter	**BSW Type**
TcpIpRemPortRangeStart	ECUC-INTEGER-PARAM-DEF
BSW Description	
Start value of the remote port range.	
Template Description	
This attribute restricts the traffic monitoring and defines a start value for the remote port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.	
M2 Parameter	
IPSecRule.remotePortRangeStart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00304]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig
BSW Parameter	**BSW Type**
TcpIpLocalAddr	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies the local IP (Internet Protocol) addresses used for IP communication.	
Template Description	
The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).	
M2 Parameter	
NetworkEndpoint	
Mapping Rule	**Mapping Type**
Create container for each NetworkEndpoint element that is defined in the ECU Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00020]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr
BSW Parameter	**BSW Type**
TcpIpAddrAssignment	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container is a subcontainer of TcpIpLocalAddr and specifies the assignment policy for the IP address.	
Template Description	
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.	
M2 Parameter	
NetworkEndpointAddress	
Mapping Rule	**Mapping Type**
Create container for each NetworkEndpointAddress element that is defined inside of the enclosing NetworkEndpoint.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00033]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment
BSW Parameter	**BSW Type**
TcpIpAssignmentLifetime	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Defines the lifetime of a dynamically fetched IP address. If TcpIpAssignmentMethod = TCPIP_STATIC then TcpIpAssignmentLifetime shall be omitted.	
Template Description	
Ipv4Configuration.ipAddressKeepBehavior: Defines the lifetime of a dynamically fetched IP address. **Ipv6Configuration.ipAddressKeepBehavior:** Defines the lifetime of a dynamically fetched IP address.	
M2 Parameter	
Ipv4Configuration.ipAddressKeepBehavior, Ipv6Configuration.ipAddressKeepBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00186]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentLifetime
BSW Parameter	**BSW Type**
TCPIP_FORGET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
After a dynamic IP address has been assigned just use it for this link-up time.	
Template Description	
After a dynamic IP address has been assigned just use it for this session.	
M2 Parameter	
IpAddressKeepEnum.forget	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentLifetime
BSW Parameter	**BSW Type**
TCPIP_STORE	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
After a dynamic IP address has been assigned store the address persistently.	
Template Description	
After a dynamic IP address has been assigned store the address persistently.	
M2 Parameter	
IpAddressKeepEnum.storePersistently	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment
BSW Parameter	**BSW Type**
TcpIpAssignmentMethod	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Method of address assignment	
Template Description	
Ipv4Configuration.ipv4AddressSource: Defines how the node obtains its IP address. **Ipv6Configuration.ipv6AddressSource:** Defines how the node obtains its IP address.	
M2 Parameter	
Ipv4Configuration.ipv4AddressSource, Ipv6Configuration.ipv6AddressSource	
Mapping Rule	**Mapping Type**
Derive parameter from the AddressSource attributes.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00035]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentMethod
BSW Parameter	**BSW Type**
TCPIP_DHCP	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Dynamic Assigned IP Address using DHCP	

▽

△

Template Description
Ipv4AddressSourceEnum.dhcpv4: DHCP is a service for the automatic IP configuration of a client. **Ipv6AddressSourceEnum.dhcpv6:** DHCP is a service for the automatic IP configuration of a client.
M2 Parameter
Ipv4AddressSourceEnum.dhcpv4, Ipv6AddressSourceEnum.dhcpv6

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentMethod

BSW Parameter	BSW Type
TCPIP_IPV6_ROUTER	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Dynamic Configured IPv6 Address by Router Advertisement	
Template Description	
IPv6 Stateless Autoconfiguration.	
M2 Parameter	
Ipv6AddressSourceEnum.routerAdvertisement	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentMethod

BSW Parameter	BSW Type
TCPIP_LINKLOCAL	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Linklocal IPv4/IPv6 Address Assignment	
Template Description	
Ipv4AddressSourceEnum.autoIp: AutoIP is used to dynamically assign IP addresses at device startup. **Ipv6AddressSourceEnum.linkLocal:** LinkLocal is intended only for communications within the segment of a local network (a link) or a point-to-point connection that a host is connected to.	
M2 Parameter	
Ipv4AddressSourceEnum.autoIp, Ipv6AddressSourceEnum.linkLocal	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentMethod
BSW Parameter	**BSW Type**
TCPIP_LINKLOCAL_DOIP	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Linklocal IPv4/IPv6 Address Assignment using DoIP Parameters	
Template Description	
Ipv4AddressSourceEnum.autoIp_doip: Linklocal IPv4 Address Assignment using DoIP Parameters **Ipv6AddressSourceEnum.linkLocal_doip:** Linklocal IPv6 Address Assignment using DoIP Parameters	
M2 Parameter	
Ipv4AddressSourceEnum.autoIp_doip, Ipv6AddressSourceEnum.linkLocal_doip	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment/TcpIpAssignmentMethod
BSW Parameter	**BSW Type**
TCPIP_STATIC	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Static Assigned IP Address	
Template Description	
Ipv4AddressSourceEnum.fixed: The IP Address shall be declared manually. **Ipv6AddressSourceEnum.fixed:** The IP Address shall be declared manually.	
M2 Parameter	
Ipv4AddressSourceEnum.fixed, Ipv6AddressSourceEnum.fixed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment
BSW Parameter	**BSW Type**
TcpIpAssignmentPriority	ECUC-INTEGER-PARAM-DEF
BSW Description	
Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.	
Template Description	

▽

△

Ipv4Configuration.assignmentPriority: Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority. **Ipv6Configuration.assignmentPriority:** Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
M2 Parameter
Ipv4Configuration.assignmentPriority, Ipv6Configuration.assignmentPriority

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00037]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr

BSW Parameter	**BSW Type**
TcpIpAddressType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Address type.	
Template Description	
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.	
M2 Parameter	
NetworkEndpointAddress	

Mapping Rule	**Mapping Type**
shall be derived from the IP Address (see more details in upstream mapping of enum literals).	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00031]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddressType

BSW Parameter	**BSW Type**
TCPIP_ANYCAST	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
Anycast address	
Template Description	
This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).	
M2 Parameter	
Ipv6Configuration.enableAnycast	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddressType	
BSW Parameter	**BSW Type**	
TCPIP_MULTICAST	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Multicast address.		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
NetworkEndpointAddress		
Mapping Rule		**Mapping Type**
Shall be set if Multicast Address is used. IPv4: 224.0.0.0 to 239.255.255.255 IPv6: address with the prefix ff00::/8.		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddressType	
BSW Parameter	**BSW Type**	
TCPIP_UNICAST	ECUC-ENUMERATION-LITERAL-DEF	
BSW Description		
Unicast address		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
NetworkEndpointAddress		
Mapping Rule		**Mapping Type**
Shall be set if Unicast Address is used.		full
Mapping Status		**ECUC Parameter ID**
valid		

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr	
BSW Parameter	**BSW Type**	
TcpIpDomainType	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		
Address family.		
Template Description		
To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter		
NetworkEndpointAddress		
Mapping Rule		**Mapping Type**
Derive this parameter from the NetworkEndpointAddress.		full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00030]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpDomainType
BSW Parameter	**BSW Type**
TCPIP_AF_INET	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
IPv4 address	
Template Description	
Internet Protocol version 4 (IPv4) configuration.	
M2 Parameter	
`Ipv4Configuration`	
Mapping Rule	**Mapping Type**
Set literal to TCPIP_AF_INETwhen the NetworkEndpoint contains an Ipv4Configuration.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpDomainType
BSW Parameter	**BSW Type**
TCPIP_AF_INET6	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
IPv6 address	
Template Description	
Internet Protocol version 6 (IPv6) configuration.	
M2 Parameter	
`Ipv6Configuration`	
Mapping Rule	**Mapping Type**
Set literal to TCPIP_AF_INET6 when the NetworkEndpoint contains an Ipv6Configuration.	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr
BSW Parameter	**BSW Type**
TcpIpLocalAddrIPv6ExtHeaderFilterRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a set of IPv6 Extension Headers which are allowed for this local IPv6 address. Note: this parameter is only relevant if the related TcpIpDomainType is TCPIP_AF_INET6.	
Template Description	
Reference to a list of IPv6 Extension Headers allowed for this SocketConnection. If no list is referenced all IPv6 Extension Headers are allowed and processed.	
M2 Parameter	
`SocketAddress.allowedIPv6ExtHeaders`	

▽

△

Mapping Rule	Mapping Type
1:1 mapping. constraint: All related SocketConnections shall reference either no or exactly the same IPv6ExtHeaderFilterList.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00200]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig
BSW Parameter	**BSW Type**
TcpIpDefaultRouter	ECUC-STRING-PARAM-DEF
BSW Description	
IP address of default router (gateway)	
Template Description	
Ipv6Configuration.defaultRouter: IP address of the default router. **Ipv4Configuration.defaultGateway:** IP address of the default gateway.	
M2 Parameter	
Ipv6Configuration.defaultRouter, Ipv4Configuration.defaultGateway	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00040]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig
BSW Parameter	**BSW Type**
TcpIpNetmask	ECUC-INTEGER-PARAM-DEF
BSW Description	
Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.	
Template Description	
Ipv4Configuration.networkMask: Network mask. Notation 255.255.255.255 **Ipv6Configuration.ipAddressPrefixLength:** IPv6 prefix length defines the part of the IPv6 address that is the network prefix.	
M2 Parameter	
Ipv4Configuration.networkMask, Ipv6Configuration.ipAddressPrefixLength	
Mapping Rule	**Mapping Type**
- 1:1 mapping for Ipv6 - conversion to CIDR notation for Ipv4	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00039]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig
BSW Parameter	**BSW Type**
TcpIpStaticIpAddress	ECUC-STRING-PARAM-DEF
BSW Description	
Static IP Address. To specify any IP address for a certain EthIfCtrl, "ANY" has to be set as wildcard. See TcpIp_Bind() for more details.	
Template Description	
Ipv4Configuration.ipv4Address: IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used. **Ipv6Configuration.ipv6Address:** IPv6 Address. Notation: FFFF:...:FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used.	
M2 Parameter	
Ipv4Configuration.ipv4Address, Ipv6Configuration.ipv6Address	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00038]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig
BSW Parameter	**BSW Type**
TcpIpTcpConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.	
Template Description	
Content Model for TCP configuration.	
M2 Parameter	
TcpTp	
Mapping Rule	**Mapping Type**
This container shall be created if the TcpTp element is used in the ECU Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00025]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpDelayedAckTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
The maximal time an acknowledgment is delayed for transmission in seconds.For further details, see also IETF RfC 1122 section 4.2.3.2.	
Template Description	
The maximal time an acknowledgement is delayed for transmission in seconds.	
M2 Parameter	
TcpProps.tcpDelayedAckTimeout	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00318]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpConfigOptionFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the white list for the filtering of TCP options, i.e. segments containing TCP options not listed here shall be silently dropped.	
Template Description	
Permitted list for the filtering of TCP options.	
M2 Parameter	
TcpOptionFilterList	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00202]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig/TcpIpTcpConfigOptionFilter
BSW Parameter	**BSW Type**
TcpIpTcpConfigOptionFilterEntry	ECUC-INTEGER-PARAM-DEF
BSW Description	
TCP option kind allowed by this filter.	
Template Description	
TCP option kind allowed by this filter.	
M2 Parameter	
TcpOptionFilterList.allowedTcpOption	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00204]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig/TcpIpTcpConfigOptionFilter
BSW Parameter	**BSW Type**
TcpIpTcpConfigOptionFilterId	ECUC-INTEGER-PARAM-DEF
BSW Description	
Identification of the TCP option filter.	
Template Description	
TCP option kind allowed by this filter.	

▽

△

M2 Parameter	
TcpOptionFilterList.allowedTcpOption	
Mapping Rule	**Mapping Type**
The list of allowedTcpOptions in TcpOptionFilterList is ordered. The Id of an entry can be derived from the order.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00203]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpCongestionAvoidanceEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.	
Template Description	
Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.	
M2 Parameter	
TcpProps.tcpCongestionAvoidanceEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00061]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpFastRecoveryEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.	
Template Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.	
M2 Parameter	
TcpProps.tcpFastRecoveryEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00063]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpFastRetransmitEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.	

▽

△

Template Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.	
M2 Parameter	
TcpProps.tcpFastRetransmitEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00062]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpFinWait2Timeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.	
Template Description	
Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.	
M2 Parameter	
TcpProps.tcpFinWait2Timeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00066]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpKeepAliveEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6	
Template Description	
Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6.	
M2 Parameter	
TcpProps.tcpKeepAliveEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00082]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**

▽

△

TcpIpTcpKeepAliveInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the interval in [s] between subsequent keepalive probes.	
Template Description	
Specifies the interval in seconds between subsequent keepalive probes.	
M2 Parameter	
TcpProps.tcpKeepAliveInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00070]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpKeepAliveProbesMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.	
Template Description	
Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.	
M2 Parameter	
TcpProps.tcpKeepAliveProbesMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00071]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpKeepAliveTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe. Note: Setting this configuration parameter to a value smaller or equal to the value of TcpIpMainFunctionPeriod results in the transmission of keep alive probes within every MainFunction cycle.	
Template Description	
Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe.	
M2 Parameter	
TcpProps.tcpKeepAliveTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00087]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpMaxRtx	ECUC-INTEGER-PARAM-DEF
BSW Description	
Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if TcpIpTcpRetransmissionTimeout/TcpIpTcpMaxRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.	
Template Description	
Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if tcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.	
M2 Parameter	
TcpProps.tcpMaxRtx	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00069]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpMsl	ECUC-FLOAT-PARAM-DEF
BSW Description	
Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcpIpTcpMsl - to ensure that the remote node received the acknowledgment to its connection termination request.)	
Template Description	
Maximum segment lifetime in [s].	
M2 Parameter	
TcpProps.tcpMsl	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00067]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpNagleEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated via TcpIp_ ChangeParameter() API.	
Template Description	
Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated per Socket (with the attribute TcpTp.nagleAlgorithm).	
M2 Parameter	

△

TcpProps.tcpNagleEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00059]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpReceiveWindowMax	ECUC-INTEGER-PARAM-DEF
BSW Description	
Default value of maximum receive window in bytes.	
Template Description	
Default value of maximum receive window in bytes.	
M2 Parameter	
TcpProps.tcpReceiveWindowMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00073]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpRetransmissionTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled or set to INF, no TCP segments shall be retransmitted. Value can be overwritten by TcpIp_ChangeParameter() API for a particular connection. If TcpIpTcpMaxRetransmissionTimeout is enabled then TcpIpTcpRetransmissionTimeout or value overwritten by TcpIp_ ChangeParameter() API is considered as initial value for first retransmission before the next valid acknowledgment arrives.	
Template Description	
Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled, no TCP segments shall be retransmitted.	
M2 Parameter	
TcpProps.tcpRetransmissionTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00068]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig
BSW Parameter	**BSW Type**
TcpIpTcpSynMaxRtx	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcpIpTcpRetransmissionTimeout/ TcpIpTcpMaxRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcpIpTcpRetransmissionTimeout/TcpIpTcpMaxRetransmissionTimeout has been expired.
Template Description
Maximum number of times that a TCP SYN is retransmitted.
M2 Parameter
TcpProps.tcpSynMaxRtx

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00064]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig

BSW Parameter	**BSW Type**
TcpIpTcpSynReceivedTimeout	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgment after having both received and sent a connection request.	
Template Description	
Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgement after having both received and sent a connection request.	
M2 Parameter	
TcpProps.tcpSynReceivedTimeout	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00065]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTcpConfig

BSW Parameter	**BSW Type**
TcpIpTcpTtl	ECUC-INTEGER-PARAM-DEF
BSW Description	
Default Time-to-live value of outgoing TCP packets.	
Template Description	
Default Time-to-live value of outgoing TCP packets.	
M2 Parameter	
TcpProps.tcpTtl	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00072]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig
BSW Parameter	**BSW Type**
TcpIpTlsConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Specifies the configuration parameters of the TLS (Transport Layer Security) sub module.	
Template Description	
This reference identifies the applicable TlsCryptoServiceMapping that adds the ability for TLS-based encryption on the enclosing ApplicationEndpoint.	
M2 Parameter	
ApplicationEndpoint.tlsCryptoMapping	
Mapping Rule	**Mapping Type**
Create this container if at least 1 ApplicationEndpoint.tlsCryptoMapping exists in the scope of this EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00219]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites
BSW Parameter	**BSW Type**
TcpIpTlsCertificateIdentity	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container provides information about the certificates used for ciphersuites.	
Template Description	
TlsCryptoCipherSuite.remoteCertificate: This reference identifies the applicable remote certificate. **TlsCryptoCipherSuite.certificate:** This reference identifies the applicable local certificate.	
M2 Parameter	
TlsCryptoCipherSuite.remoteCertificate, TlsCryptoCipherSuite.certificate	
Mapping Rule	**Mapping Type**
Create one container for each local/remote certificate pair (or single local or remote certificate) referenced from a TlsCryptoCipherSuite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00240]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCertificateIdentity
BSW Parameter	**BSW Type**
TcpIpTlsCipherKeyMLocalCertificate	ECUC-REFERENCE-DEF
BSW Description	
Reference to a KeyM certificate used to address the local certificate.	
Template Description	
This reference identifies the applicable local certificate.	
M2 Parameter	
TlsCryptoCipherSuite.certificate	



△

Mapping Rule	Mapping Type
Reference to Representation of Local Certificate in KeyM. For servers configured if key exchange cipher suite is used, for clients only if key exchange cipher suite is used and TcpIpTlsConnection/ TcpIpTlsUseClientAuthenticationRequest is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00286]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCertificateIdentity
BSW Parameter	**BSW Type**
TcpIpTlsCipherKeyMRemoteCertificate	ECUC-REFERENCE-DEF
BSW Description	
Reference to KeyM certificate container to reference the remote certificate.	
Template Description	
This reference identifies the applicable remote certificate.	
M2 Parameter	
TlsCryptoCipherSuite.remoteCertificate	
Mapping Rule	**Mapping Type**
Reference to Representation of Remote Certificate in KeyM. For clients configured if key exchange cipher suite is used, for servers only if key exchange cipher suite is used and TcpIpTlsConnection/ TcpIpTlsUseClientAuthenticationRequest is set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00287]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCertificateIdentity
BSW Parameter	**BSW Type**
TcpIpTlsServerNameIdentification	ECUC-STRING-PARAM-DEF
BSW Description	
Defines a server identification name. If present, the name will be added as an extension with the "TLS client hello" handshake message. The TLS server will check for the name to identify the server certificate.	
Template Description	
Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.	
M2 Parameter	
CryptoServiceCertificate.serverNameIdentification	
Mapping Rule	**Mapping Type**
Retrieve the CryptoServiceCertificate representing the certificate of the server from TlsCrypto CipherSuite. Use CryptoServiceCertificate.serverNameIdentification to configure TcpIpTlsServerName Identification that hints to the server certificate which needs identification. Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00278]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites	
BSW Parameter		**BSW Type**
TcpIpTlsCiphersuiteDefinition		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This container provides the static information of a ciphersuite used by TLS.		
Template Description		
This meta-class represents a cipher suite for describing cryptographic operations in the context of establishing a connection of ApplicationEndpoints that is protected by TLS.		
M2 Parameter		
TlsCryptoCipherSuite		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00237]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsCiphersuiteId		ECUC-INTEGER-PARAM-DEF
BSW Description		
ID that represents the ciphersuite according to IETF, e.g. RFC4492, Sect. 6, RFC8446, Appendix B.4 or RFC5246, Appendix A.5.		
Template Description		
Identification of the CipherSuite according to the IANA assignments list.		
M2 Parameter		
TlsCryptoCipherSuite.cipherSuiteId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00242]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsCiphersuiteName		ECUC-STRING-PARAM-DEF
BSW Description		
Provides a verbal name for the ciphersuite. The name should be the one defined in the respective RFC, e.g. TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (TLS 1.2) or TLS_AES_128_GCM_SHA256 (TLS 1.3)		
Template Description		
Name of the CipherSuite according to the IANA assignments list.		

▽

△

M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteShortLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00244]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsCiphersuitePriority		ECUC-INTEGER-PARAM-DEF
BSW Description		
Defines the priority of the cipher. The higher the number the lower the priority.		
Template Description		
This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.		
M2 Parameter		
TlsCryptoCipherSuite.priority		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00243]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsUseAEADCipher		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Specifies if the ciphersuite supports AEAD for data en-/decryption.		
Template Description		
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.		
M2 Parameter		
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption		
Mapping Rule		**Mapping Type**
Derive from TlsCryptoCipherSuite.cipherSuiteId or TlsCryptoCipherSuite.encryption.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00247]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsUsePresharedKeys		ECUC-BOOLEAN-PARAM-DEF

▽

△

BSW Description	
Defines if this ciphersuite uses pre-shared keys. If so, additional configuration or callbacks will be used for pre-shared key negotiation.	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.pskIdentity:** Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.pskIdentity	
Mapping Rule	**Mapping Type**
Derive from TlsCryptoCipherSuite if whether PSK is supported.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00245]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsUseSecurityExtensionForceEncryptThenMac		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Defines if the security extension according to IETF RFC 7366 shall be supported. This is useful for ciphersuites using CBC mode.		
Template Description		
Defines if the security extension according to IETF RFC 7366 shall be supported. This is useful for cipher suites using CBC mode.		
M2 Parameter		
TlsCryptoCipherSuiteProps.tcpIpTlsUseSecurityExtensionForceEncryptThenMac		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00246]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteDefinition	
BSW Parameter		**BSW Type**
TcpIpTlsVersion		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
Declares the TLS version that this ciphersuite shall be used for.		
Template Description		
This attribute supports the definition of the applicable version of TLS.		
M2 Parameter		
TlsCryptoCipherSuite.version		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00248]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherAEADCipherKeyLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the key length for en- / decryption with authentication data (AEAD).	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the byte length of the keys used for message encryption if the cipher suite represented by TlsCryptoCipherSuite uses AEAD for message encryption.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00254]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmDecryptJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the data decryption operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports encryption.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00255]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmDecryptKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key associated to the CSM job that performs the data decryption operation	
Template Description	

▽

△

TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports encryption.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00256]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmEncryptJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the data encryption operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports encryption.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00251]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmEncryptKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key associated to the CSM job that performs the data encryption operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports encryption.	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_TcpIp_00252]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmMacGenerateJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the MAC generate operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports message authentication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00258]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmMacGenerateKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key associated to the CSM job that performs the MAC generate operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports message authentication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00259]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmMacVerifyJobRef	ECUC-REFERENCE-DEF

▽

△

BSW Description	
Reference to a CSM job to perform the MAC verify operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports message authentication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00260]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmMacVerifyKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key associated to the CSM job that performs the MAC verify operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports message authentication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00261]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherCsmRneJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM random number generation job used for DTLS 1.3 cipher operations. This parameter is separate from TcpIpTlsCsmRandomGenerateJobRef (ECUC_TcpIp_00221) which is scoped to the Tls branch.	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	

▽

△

TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the reference if the cipher suite represented by TlsCryptoCipherSuite supports message authentication.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00371]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherEncryptKeyLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the key length used for en- or decryption. The key length is valid for (symmetric) encryption and decryption.	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.encryption:** This reference identifies the crypto service primitive for the execution of encryption.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.encryption	
Mapping Rule	**Mapping Type**
Configure the byte length of the keys used for message encryption if the cipher suite represented by TlsCryptoCipherSuite supports encryption.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00253]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCipherMacKeyLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the length of the MAC key	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.authentication:** This reference identifies the crypto service primitive for the generation and verification of MACs.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.authentication	
Mapping Rule	**Mapping Type**
Configure the byte length of the keys used for message authentication by the cipher suite represented by TlsCryptoCipherSuite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00257]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsCiphersuiteDefinitionRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a ciphersuite definition container	
Template Description	
This meta-class represents a cipher suite for describing cryptographic operations in the context of establishing a connection of ApplicationEndpoints that is protected by TLS.	
M2 Parameter	
TlsCryptoCipherSuite	
Mapping Rule	**Mapping Type**
Reference the TcpIpTlsCiphersuiteDefinition container that has been created for the TlsCrypto CipherSuite element the TcpIpTlsCiphersuiteWorker container has been created for as well.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00250]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsCiphersuiteWorker
BSW Parameter	**BSW Type**
TcpIpTlsConnectionHandshakeRef	ECUC-REFERENCE-DEF
BSW Description	
References the container that contains the jobs and keys for handshake operation. Referencing multiple handshake containers allow to share them between workers and to choose the next unused during the handshake.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.signatureScheme:** This reference points to the properties of a TLS Signature Scheme. **TlsCryptoCipherSuite.ellipticCurve:** This references point to the properties of elliptic curves.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoCipherSuite.signatureScheme, TlsCryptoCipherSuite.ellipticCurve	
Mapping Rule	**Mapping Type**
Reference all TcpIpTlsHandshake containers created for the TlsCryptoCiphersuite this container has been created for as well.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00249]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites
BSW Parameter	**BSW Type**
TcpIpTlsHandshake	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

This container provides information that is needed to process a handshake. It contains the appropriate references to jobs and keys of the CSM to perform the key exchange cryptographic for the ciphersuite and involved certificates. Entries may reuse job references across supported groups, but key binding and runtime context shall remain handshake-specific to support parallel handshakes.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.ellipticCurve:** This references point to the properties of elliptic curves. **TlsCryptoCipherSuite.signatureScheme:** This reference points to the properties of a TLS Signature Scheme.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoCipherSuite.ellipticCurve, TlsCryptoCipherSuite.signatureScheme	
Mapping Rule	**Mapping Type**
Create one container for each TlsCryptoCipherSuite that is aggregated by any TlsCryptoService Mapping of the configured EcuInstance if the TlsCryptoCipherSuite does not use key exchange. Create one container for each supported curve/signature scheme combination supported by any TlsCryptoCipherSuite that is aggregated by any TlsCryptoServiceMapping of the configured Ecu Instance if the TlsCryptoCipherSuite is using key exchange.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00239]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsHandshake
BSW Parameter	**BSW Type**
TcpIpTlsCsmHashVerifyJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the hash operation for the whole handshake.	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.keyExchange:** This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange	
Mapping Rule	**Mapping Type**
Reference into Csm: Job for calculating Hash of Handshake Payload, can be derived from cipher suite id: PRF P_SHA256 per default, unless the cipher suite explicitly specifies a different PRF.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00265]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsHandshake
BSW Parameter	**BSW Type**
TcpIpTlsCsmMasterSecretKeyRef	ECUC-REFERENCE-DEF
BSW Description	
This is the reference to the master key that is calculated during the session.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm: Master secret key derived from the shared (PreMaster) Secret.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00266]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsHandshake
BSW Parameter	**BSW Type**
TcpIpTlsCsmPrfMacJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the PRF hash operation	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm: Job for calculating the MasterSecret from the PreMaster Secret.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00262]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsHandshake

BSW Parameter	BSW Type
TcpIpTlsCsmPrfMacKeyRef	ECUC-REFERENCE-DEF

BSW Description

Reference to a CSM key associated to the CSM job that performs the PRF hash operation

Template Description

TlsCryptoCipherSuite.keyExchange:
This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase.

TlsCryptoServiceMapping.keyExchange:
This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase.

TlsCryptoCipherSuite.cipherSuiteId:
Identification of the CipherSuite according to the IANA assignments list.

M2 Parameter

`TlsCryptoCipherSuite.keyExchange`, `TlsCryptoServiceMapping.keyExchange`, `TlsCryptoCipherSuite.cipherSuiteId`

Mapping Rule	Mapping Type
Reference into Csm: Key that is used for deriving the Master secret from the PreMaster Secret.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00263]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites

BSW Parameter	BSW Type
TcpIpTlsKeyExchangeDefinition	ECUC-PARAM-CONF-CONTAINER-DEF

BSW Description

Defines a key exchange group and its priority for TLS and DTLS connections.

Template Description

TlsCryptoCipherSuite.keyExchange:
This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase.

TlsCryptoCipherSuite.ellipticCurve:
This references point to the properties of elliptic curves.

TlsCryptoCipherSuite.signatureScheme:
This reference points to the properties of a TLS Signature Scheme.

M2 Parameter

`TlsCryptoCipherSuite.keyExchange`, `TlsCryptoCipherSuite.ellipticCurve`, `TlsCryptoCipherSuite.signatureScheme`

Mapping Rule	Mapping Type
Create one container for each TlsCryptoCipherSuite that is aggregated by any TlsCryptoService Mapping of the configured EcuInstance if the TlsCryptoCipherSuite does not use key exchange. Create one container for each supported curve/signature scheme combination supported by any TlsCryptoCipherSuite that is aggregated by any TlsCryptoServiceMapping of the configured Ecu Instance if the TlsCryptoCipherSuite is using key exchange.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00348]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeCalcPubValJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the DH Key Exchange algorithm operation	
Template Description	
TlsCryptoCipherSuite.cipherSuiteId: Identification of the CipherSuite according to the IANA assignments list. **TlsCryptoCipherSuite.keyExchange:** This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase.	
M2 Parameter	
TlsCryptoCipherSuite.cipherSuiteId, TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange	
Mapping Rule	**Mapping Type**
Reference into Csm: Job for calculating Public and private Key Pair for DH Key Exchange.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00267]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeCalcSecretJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform the Key Exchange algorithm operation	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm: Job for calculating shared secret in DH Key Exchange.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00269]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeDecryptJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform data decryption, e.g. with RSA key exchange operation.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm, only for ECDH_RSA, ECDHE_RSA cipher suite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00276]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeDecryptKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key to perform data decryption, e.g. with RSA, used for exchange operation.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm, only for ECDH_RSA, ECDHE_RSA cipher suite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00277]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeEncryptJobRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM job to perform data encryption, e.g. with RSA key exchange operation.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm, only for ECDH_RSA, ECDHE_RSA cipher suite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00274]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeEncryptKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key to perform data encryption, e.g. with RSA, used for exchange operation.	
Template Description	
TlsCryptoCipherSuite.keyExchange: This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoServiceMapping.keyExchange:** This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **TlsCryptoCipherSuite.cipherSuiteId:** Identification of the CipherSuite according to the IANA assignments list.	
M2 Parameter	
TlsCryptoCipherSuite.keyExchange, TlsCryptoServiceMapping.keyExchange, TlsCryptoCipherSuite.cipherSuiteId	
Mapping Rule	**Mapping Type**
Reference into Csm, only for ECDH_RSA, ECDHE_RSA cipher suite.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00275]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker

BSW Parameter	BSW Type
TcpIpTlsCsmKeyExchangeKeyRef	ECUC-REFERENCE-DEF

BSW Description

Reference to a CSM key used for Diffie Hellman (DH) key exchange operation.

Template Description

TlsCryptoCipherSuite.keyExchange:
This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase.

TlsCryptoServiceMapping.keyExchange:
This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase.

TlsCryptoCipherSuite.cipherSuiteId:
Identification of the CipherSuite according to the IANA assignments list.

M2 Parameter

`TlsCryptoCipherSuite.keyExchange`, `TlsCryptoServiceMapping.keyExchange`, `TlsCryptoCipherSuite.cipherSuiteId`

Mapping Rule	Mapping Type
Reference into Csm: Key containing public/private Keys and Secret of DH Key Exchange.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00268]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker

BSW Parameter	BSW Type
TcpIpTlsCsmKeyExchangeSignatureGenerateJobRef	ECUC-REFERENCE-DEF

BSW Description

Reference to a CSM job to perform signature generation for DH operation

Template Description

TlsCryptoCipherSuite.signatureScheme:
This reference points to the properties of a TLS Signature Scheme.

TlsCryptoCipherSuite.keyExchangeAuthentication:
This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm.

TlsCryptoServiceMapping.useClientAuthenticationRequest:
Defines if client authentication shall be applied for this TLS connection.

M2 Parameter

`TlsCryptoCipherSuite.signatureScheme`, `TlsCryptoCipherSuite.keyExchangeAuthentication`, `TlsCryptoServiceMapping.useClientAuthenticationRequest`

Mapping Rule	Mapping Type
Configure this reference if the local node has to authenticate itself towards the remote node, taking into account the role of the local node (client or server) TlsCryptoServiceMapping.category and Tls CryptoServiceMapping.useClientAuthenticationRequest.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00270]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker

BSW Parameter	BSW Type
TcpIpTlsCsmKeyExchangeSignatureGenerateKeyRef	ECUC-REFERENCE-DEF

BSW Description

Reference to a CSM key to perform signature generation for DH operation

Template Description

TlsCryptoCipherSuite.signatureScheme:
This reference points to the properties of a TLS Signature Scheme.

TlsCryptoCipherSuite.keyExchangeAuthentication:
This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm.

TlsCryptoServiceMapping.useClientAuthenticationRequest:
Defines if client authentication shall be applied for this TLS connection.

M2 Parameter

`TlsCryptoCipherSuite.signatureScheme`, `TlsCryptoCipherSuite.keyExchangeAuthentication`, `TlsCryptoServiceMapping.useClientAuthenticationRequest`

Mapping Rule	Mapping Type
Configure this reference if the local node has to authenticate itself towards the remote node, taking into account the role of the local node (client or server) TlsCryptoServiceMapping.category and TlsCryptoServiceMapping.useClientAuthenticationRequest.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00271]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker

BSW Parameter	BSW Type
TcpIpTlsCsmKeyExchangeSignatureVerifyJobRef	ECUC-REFERENCE-DEF

BSW Description

Reference to a CSM job to perform signature verification for DH operation

Template Description

TlsCryptoCipherSuite.signatureScheme:
This reference points to the properties of a TLS Signature Scheme.

TlsCryptoCipherSuite.keyExchangeAuthentication:
This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm.

TlsCryptoServiceMapping.useClientAuthenticationRequest:
Defines if client authentication shall be applied for this TLS connection.

M2 Parameter

`TlsCryptoCipherSuite.signatureScheme`, `TlsCryptoCipherSuite.keyExchangeAuthentication`, `TlsCryptoServiceMapping.useClientAuthenticationRequest`

Mapping Rule	Mapping Type
Configure this reference if the local node has to verify the authenticity of the remote node, taking into account the role of the local node (client or server) TlsCryptoServiceMapping.category and TlsCryptoServiceMapping.useClientAuthenticationRequest.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00272]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsKeyExchangeWorker
BSW Parameter	**BSW Type**
TcpIpTlsCsmKeyExchangeSignatureVerifyKeyRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a CSM key to perform signature verification for DH operation	
Template Description	
TlsCryptoCipherSuite.signatureScheme: This reference points to the properties of a TLS Signature Scheme. **TlsCryptoCipherSuite.keyExchangeAuthentication:** This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm. **TlsCryptoServiceMapping.useClientAuthenticationRequest:** Defines if client authentication shall be applied for this TLS connection.	
M2 Parameter	
TlsCryptoCipherSuite.signatureScheme, TlsCryptoCipherSuite.keyExchangeAuthentication, TlsCryptoServiceMapping.useClientAuthenticationRequest	
Mapping Rule	**Mapping Type**
Configure this reference if the local node has to verify the authenticity of the remote node, taking into account the role of the local node (client or server) TlsCryptoServiceMapping.category and Tls CryptoServiceMapping.useClientAuthenticationRequest.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00273]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites
BSW Parameter	**BSW Type**
TcpIpTlsPskIdentity	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container provides information about static definition of pre-shared keys. It is used during the handshake to negotiate pre-shared keys between a client and a server. Note: The callbacks for pre-shared keys are an alternative to the static definition. The callbacks allow to define the associated keys at runtime if pre-shared keys are used but no static definition is available. The container definition is used for static configuration.	
Template Description	
Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.	
M2 Parameter	
TlsCryptoCipherSuite.pskIdentity	
Mapping Rule	**Mapping Type**
Create one container for each TlsPskIdentity element aggregated by any TlsCryptoCipherSuite that in turn is aggregated by any TlsCryptoServiceMapping of the configured EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00241]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsPskIdentity
BSW Parameter	**BSW Type**
TcpIpTlsPresharedKeyCsmKeyRef	ECUC-REFERENCE-DEF



△

BSW Description	
Reference to a CSM key associated to the CSM job that performs the PRF hash operation	
Template Description	
This element is used to describe the pre-shared key shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.	
M2 Parameter	
TlsPskIdentity	
Mapping Rule	**Mapping Type**
Reference into the Csm: Pre-Shared symmetric Key.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00280]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsPskIdentity
BSW Parameter	**BSW Type**
TcpIpTlsPresharedKeyIdentity	ECUC-STRING-PARAM-DEF
BSW Description	
This item provides the key identification. The TLS client selects the pre-shared key based on the identification hint provided by the server and returns the key identification name back to the server.	
Template Description	
This attribute provides the key identification.	
M2 Parameter	
TlsPskIdentity.pskIdentity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00284]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig/TcpIpTlsCiphersuites/TcpIpTlsPskIdentity
BSW Parameter	**BSW Type**
TcpIpTlsPresharedKeyIdentityHint	ECUC-STRING-PARAM-DEF
BSW Description	
Provides the identity hint for a pre-shared key. This information is transmitted by the TLS Server to provide its identification to the TLS client. The TLS client uses the same information to select the pre-shared key.	
Template Description	
This attribute provides the identity hint for a pre-shared key.	
M2 Parameter	
TlsPskIdentity.pskIdentityHint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00279]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpTlsConfig	
BSW Parameter		**BSW Type**
TcpIpTlsConnectionGroup		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
This optional container is used to collect all TlsConnections that belong to a TlsConnectionGroup. The intention of a TLS connection group is to share resources among TLS connections collected in a group, because only one connection of a group can be used at a time.		
Template Description		
Defines a collection of TlsCryptoServiceMappings which will not be active at the same time during runtime.		
M2 Parameter		
TlsConnectionGroup		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00224]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpIpUdpConfig		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module		
Template Description		
Content Model for UDP configuration.		
M2 Parameter		
UdpTp		
Mapping Rule		**Mapping Type**
This container shall be created if the UdpTp element is used in the ECU Extract.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00026]

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpUdpConfig	
BSW Parameter		**BSW Type**
TcpIpUdpTtl		ECUC-INTEGER-PARAM-DEF
BSW Description		
Default Time-to-live value of outgoing UDP packets.		
Template Description		
Default Time-to-live value of outgoing UDP packets.		
M2 Parameter		
UdpProps.udpTtl		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00075]

BSW Module	BSW Context
TcpIp	TcpIp
BSW Parameter	**BSW Type**
TcpIpTlsConnection	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container defines the properties of a TLS/DTLS protocol	
Template Description	
This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS).	
M2 Parameter	
TlsCryptoServiceMapping	
Mapping Rule	**Mapping Type**
Create one container for each TlsCryptoServiceMapping of the configured EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00223]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpTlsConnection
BSW Parameter	**BSW Type**
TcpIpTlsCertificateIdentityRef	ECUC-REFERENCE-DEF
BSW Description	
References the container that contains the certificate and identity information.	
Template Description	
TlsCryptoCipherSuite.certificate: This reference identifies the applicable local certificate. **TlsCryptoServiceMapping.tlsCipherSuite:** This aggregation represents the collection of supported cipher suites.	
M2 Parameter	
TlsCryptoCipherSuite.certificate, TlsCryptoServiceMapping.tlsCipherSuite	
Mapping Rule	**Mapping Type**
References to all TcpIpTlsCertificateIdentity containers that were created for the TlsCryptoService Mapping the TcpIpTlsConnection container was created for.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00235]

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpTlsConnection
BSW Parameter	**BSW Type**
TcpIpTlsConnectionGroupRef	ECUC-REFERENCE-DEF
BSW Description	
Assigns the TLS connection to a connection group.	
Template Description	
Reference target is member of a TlsConnectionGroup.	
M2 Parameter	
TlsConnectionGroup.tlsConnection	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00233]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpTlsConnection
BSW Parameter	**BSW Type**
TcpIpTlsConnectionPskIdentityRef	ECUC-REFERENCE-DEF
BSW Description	
References the container that contains information about pre-shared keys.	
Template Description	
TlsCryptoCipherSuite.pskIdentity: Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key. **TlsCryptoServiceMapping.tlsCipherSuite:** This aggregation represents the collection of supported cipher suites.	
M2 Parameter	
TlsCryptoCipherSuite.pskIdentity, TlsCryptoServiceMapping.tlsCipherSuite	
Mapping Rule	**Mapping Type**
References to all TcpIpTlsPskIdentity containers that were created for the TlsCryptoService Mapping the TcpIpTlsConnection container was created for.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00236]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpTlsConnection
BSW Parameter	**BSW Type**
TcpIpTlsConnectionType	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
Specifies if the TLS or DTLS secured connection is used as a server or a client.	
Template Description	
The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
TlsCryptoServiceMapping.category	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00226]

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpTlsConnection
BSW Parameter	**BSW Type**
TcpIpTlsIpAddressAssignment	ECUC-REFERENCE-DEF
BSW Description	

▽

△

Contains additional information about the endpoint IP address information. If this reference is present, the IP address of the connecting socket shall also be checked if a TLS connection shall be assigned automatically to a socket.	
Template Description	
M2 Parameter	
ApplicationEndpoint.networkEndpoint	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00229]

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpTlsConnection	
BSW Parameter		**BSW Type**
TcpIpTlsPortAssignment		ECUC-INTEGER-PARAM-DEF
BSW Description		
Specifies the port address that is used for TLS communication.		
Template Description		
Port Number.		
M2 Parameter		
TpPort.portNumber		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00285]

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpTlsConnection	
BSW Parameter		**BSW Type**
TcpIpTlsUseClientAuthenticationRequest		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
Defines if client authentication shall be applied for this TLS connection.		
Template Description		
Defines if client authentication shall be applied for this TLS connection.		
M2 Parameter		
TlsCryptoServiceMapping.useClientAuthenticationRequest		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00230]

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpTlsConnection/TcpIpTlsUseExtensions	
BSW Parameter		**BSW Type**

▽

△

TcpIpTlsUseExtensionRecordSizeLimit	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Defines if the security extension for record_size_limit shall be supported as defined in IETF RFC 8449, chapter 4.1.	
Template Description	
Defines if the security extension for max_fragment_length shall be supported as defined in IETF RFC 8449, chapter 4.1.	
M2 Parameter	
TlsCryptoServiceMapping.useSecurityExtensionRecordSizeLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_TcpIp_00231]

C.58 UdpNm

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmBusSynchronizationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmBusSynchronizationEnabled.	
Template Description	
Enables bus synchronization support.	
M2 Parameter	
NmEcu.nmBusSynchronizationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00006]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmChannelConfig	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container contains the channel-specific configuration parameters of the UdpNm.	
Template Description	
Udp specific NmCluster attributes	
M2 Parameter	
UdpNmCluster	

▽

△

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00017]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmAllNmMessagesKeepAwake	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Specifies if UdpNm drops irrelevant NM PDUs. false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling true: Every NM PDU triggers the standard RX indication handling	
Template Description	
Specifies if Nm drops irrelevant NM PDUs. false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake	
M2 Parameter	
UdpNmNode.allNmMessagesKeepAwake	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00089]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmDynamicPncToChannelMappingEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Channel-specific parameter to enable the dynamic PNC-to-channel-mapping feature. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
1:1 Mapping. If M2 Parameter not defined then do not create UdpNmDynamicPncToChannel MappingEnabled.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_UdpNm_00095]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmImmediateNmCycleTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Defines the immediate NM PDU cycle time in seconds which is used for UdpNmImmediateNmTransmissions NM PDU transmissions.	
Template Description	
Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmImmediateNmTransmissions is greater one.	
M2 Parameter	
UdpNmCluster.nmImmediateNmCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00079]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmImmediateNmTransmissions	ECUC-INTEGER-PARAM-DEF
BSW Description	
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immeditate NM PDUs is defined by UdpNmImmediateNmCycleTime.	
Template Description	
Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.	
M2 Parameter	
UdpNmCluster.nmImmediateNmTransmissions	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00075]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmMsgCycleOffset	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time offset in the periodic transmission node. It determines the start delay of the transmission. < UdpNmMsgCycleTime This parameter is only valid if UdpNmPassiveModeEnabled is disabled.	
Template Description	

▽

△

Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
M2 Parameter	
UdpNmNode.nmMsgCycleOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00029]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmMsgCycleTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Period of a NM-message. It determines the periodic rate and is the basis for transmit scheduling. NmTimeoutTime = n * Udp NmMsgCycleTime This parameter is only valid if UdpNmPassiveModeEnabled is disabled.	
Template Description	
Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.	
M2 Parameter	
UdpNmCluster.nmMsgCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00028]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmNodeDetectionEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the node detection support. This parameter shall be derived from NmNodeDetectionEnabled. This parameter shall only be enabled if UdpNmNodeId Enabled == true. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False).	
Template Description	
Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter	
NmCluster.nmNodeDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00090]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	**BSW Type**	
UdpNmNodeId	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Node identifier of local node.		
Template Description		
Node identifier of local NmNode. Shall be unique in the NmCluster.		
M2 Parameter		
NmNode.nmNodeId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_UdpNm_00031]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	**BSW Type**	
UdpNmNodeIdEnabled	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		
Pre-processor switch for enabling the source node identifier. This parameter shall be derived from NmNodeIdEnabled.		
Template Description		
Enables the source node identifier.		
M2 Parameter		
NmCluster.nmNodeIdEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_UdpNm_00091]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter	**BSW Type**	
UdpNmPduCbvPosition	ECUC-ENUMERATION-PARAM-DEF	
BSW Description		

▽

△

Defines the position of the control bit vector within the NM PACKET.
The value of the parameter represents the location of the control bit vector in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means the control bit vector is not part of the NM PACKET)
See also UdpNmPduNidPosition
if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition != UDPNM_PDU_OFF) then UdpNmPduCbv Position != UdpNmPduNidPosition
if (UdpNmPduCbvPosition != UDPNM_PDU_OFF && UdpNmPduNidPosition == UDPNM_PDU_OFF) then UdpNmPduCbv Position = UDPNM_PDU_BYTE0

Template Description
Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
M2 Parameter
UdpNmCluster.nmCbvPosition

Mapping Rule	**Mapping Type**
Derive byte position from nmCbvPosition attribute. If this optional attribute is missing set UDPNM_ PDU_OFF as value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_ 00026]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig

BSW Parameter	**BSW Type**
UdpNmPduNidPosition	ECUC-ENUMERATION-PARAM-DEF

BSW Description
Defines the position of the source node identifier within the NM PACKET. ImplementationType: UdpNm_PduPositionType The value of the parameter represents the location of the source node identifier in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means source node identifier is not part of the NM PACKET) See also UdpNmPduCbvPosition if (UdpNmPduNidPosition!= UDPNM_PDU_OFF && UdpNmPduCbvPosition != UDPNM_ PDU_OFF) then UdpNmPduNidPosition != UdpNmPduCbvPosition if (UdpNmPduNidPosition != UDPNM_PDU_OFF && UdpNmPduCbvPosition == UDPNM_PDU_OFF) then UdpNmPduNid Position = UDPNM_PDU_BYTE0
Template Description
Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
M2 Parameter
UdpNmCluster.nmNidPosition

Mapping Rule	**Mapping Type**
Derive byte position from nmNidPosition attribute. If this optional attribute is missing set UDPNM_ PDU_OFF as value.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_ 00025]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPnEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported	
Template Description	
Defines whether this NmCluster contributes to the partial network mechanism.	
M2 Parameter	
NmCluster.nmPncParticipation	
Mapping Rule	**Mapping Type**
If NmCluster.nmPncParticipation has the value "true" or is not defined then UdpNmPnEnabled shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00061]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPnHandleMultipleNetworkRequests	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
false: UdpNm_NetworkRequest is ignored in NO. true: UdpNm_NetworkRequest triggers a change from NO to RM.	
Template Description	
Defines if Nm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).	
M2 Parameter	
UdpNmCluster.nmPnHandleMultipleNetworkRequests	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00063]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmRemoteSleepIndTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Remote Sleep Indication. It defines the time in [s] how long it shall take to recognize that all other nodes are ready to sleep. Typically it should be equal to: n * UdpNmMsgCycleTime, where n denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the Remote Sleep Indication procedure.	
Template Description	
Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.	

▽

△

M2 Parameter	
UdpNmCluster.nmRemoteSleepIndicationTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00023]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmRepeatMessageTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.	
Template Description	
Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
M2 Parameter	
UdpNmCluster.nmRepeatMessageTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00022]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmRepeatMsgIndEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable/disable the notification that a RepeatMessageRequest bit has been received.	
Template Description	
Switch for enabling the Repeat Message Bit Indication.	
M2 Parameter	
NmCluster.nmRepeatMsgIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00092]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF

▽

△

BSW Description	
This container describes the UdpNm RX PDU's.	
Template Description	
M2 Parameter	
NmNode.rxNmPdu	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that is received on the regarded Nm cluster	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00038]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmTimeoutTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Network Timeout for NM packets. It denotes the time in [s] how long the NM shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be greater than UdpNmMsgCycleTime. Typically, it should be equal to: x * UdpNmMsgCycleTime, where n denotes the number of NM PACKET cycle times in the Ready Sleep State before transition into the Bus-Sleep Mode is initiated. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the coordination algorithm.	
Template Description	
Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.	
M2 Parameter	
UdpNmCluster.nmNetworkTimeout	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00020]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container describes the UdpNm TX PDU's.	
Template Description	
M2 Parameter	
NmNode.txNmPdu	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that is transmitted on the regarded Nmcluster	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00036]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmUserDataTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
Preprocessor switch for enabling the Tx path of Com User Data. Use case: Setting of NMUserData via SWC.	
Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00056]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmWaitBusSleepTime	ECUC-FLOAT-PARAM-DEF
BSW Description	
Timeout for bus calm down phase. It denotes the time in [s] how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be long enough to empty all Tx-buffer empty.	
Template Description	
Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.	
M2 Parameter	
UdpNmCluster.nmWaitBusSleepTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00021]

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmComControlEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
Template Description	
Enables the Communication Control support.	
M2 Parameter	
NmEcu.nmComControlEnabled	
Mapping Rule	**Mapping Type**



△

1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00013]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmComUserDataSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Enable/disable the user data support.	
Template Description	
M2 Parameter	
NmPdu.iSignalToIPduMapping	
Mapping Rule	**Mapping Type**
If an NmPdu contains user data defined via the existence of NmPdu.iSignalToIPduMapping and is consequently handled via the PduR and Com this attribute shall be set to true.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00055]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmDynamicPncToChannelMappingSupport	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Precompile time switch to enable the dynamic PNC-to-channel-mapping handling. False: Dynamic PNC-to-channel-mapping is disabled True: Dynamic PNC-to-channel-mapping is enabled	
Template Description	
Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this Communication Connector and its respective PhysicalChannel.	
M2 Parameter	
CommunicationConnector.dynamicPncToChannelMappingEnabled	
Mapping Rule	**Mapping Type**
If at least one dynamicPncToChannelMappingEnabled attribute is defined and if at least one CommunicationConnector of the EcuInstance has dynamicPncToChannelMappingEnabled set to true, then UdpNmDynamicPncToChannelMappingSupport shall be set to true. Otherwise UdpNm DynamicPncToChannelMappingSupport shall be set to false.	full
Mapping Status	**ECUC Parameter ID**
draft	[ECUC_UdpNm_00094]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**

▽

△

UdpNmImmediateRestartEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the immediate transmission of a NM PACKET upon bus-communication request in Prepare-Bus-Sleep mode. Must not be defined if UdpNmPassiveModeEnabled== true.	
Template Description	
Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
M2 Parameter	
UdpNmClusterCoupling.nmImmediateRestartEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00009]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmPassiveModeEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling support of the Passive Mode.	
Template Description	
Enables support of the Passive Mode. The passive mode is configurable per channel.	
M2 Parameter	
NmNode.nmPassiveModeEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping nmNode.nmPassiveModeEnabled shall always have the same value in all NmClusters with the same bus protocol in the scope of one EcuInstance.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00010]

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmPduRxIndicationEnabled	ECUC-BOOLEAN-PARAM-DEF
BSW Description	
Pre-processor switch for enabling the PDU Rx Indication. This parameter shall be derived from NmPduRxIndicationEnabled.	
Template Description	
Switch for enabling the PDU Rx Indication.	
M2 Parameter	
NmEcu.nmPduRxIndicationEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_UdpNm_00011]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	**BSW Type**	
UdpNmRemoteSleepIndEnabled	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		
Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only. It must not be defined if UdpNmPassiveModeEnabled==true. This parameter shall be derived from NmRemoteSleepIndEnabled.		
Template Description		
Switch for enabling remote sleep indication support.		
M2 Parameter		
NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_UdpNm_00005]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	**BSW Type**	
UdpNmStateChangeIndEnabled	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		
Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NmState ChangeIndEnabled.		
Template Description		
Enables the CAN Network Management state change notification.		
M2 Parameter		
NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_UdpNm_00012]

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter	**BSW Type**	
UdpNmUserDataEnabled	ECUC-BOOLEAN-PARAM-DEF	
BSW Description		

▽

△

Pre-processor switch for enabling user data support. This parameter shall be derived from NmUserDataEnabled.	
Template Description	
Switch for enabling user data support.	
M2 Parameter	
NmEcu.nmUserDataEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_UdpNm_00004]

C.59 VdpCmR

BSW Module	**BSW Context**
VdpCmR	VdpCmR/VdpCmRCsl
BSW Parameter	**BSW Type**
VdpCmRCslRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the general rx message.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**
1:1 mapping if generalPurposeIPdu.category == VDP	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_VdpCmR_00022]

BSW Module	**BSW Context**
VdpCmR	VdpCmR/VdpCmRCsl
BSW Parameter	**BSW Type**
VdpCmRCslSecuredTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the secured tx message.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**

▽

△

1:1 mapping if generalPurposeIPdu.category == VDP	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_VdpCmR_00028]

BSW Module	**BSW Context**
VdpCmR	VdpCmR/VdpCmRCsl
BSW Parameter	**BSW Type**
VdpCmRCslTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies the general tx message.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**
1:1 mapping if generalPurposeIPdu.category == VDP	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_VdpCmR_00025]

C.60 Xcp

BSW Module	**BSW Context**
Xcp	Xcp/XcpConfig
BSW Parameter	**BSW Type**
XcpCommunicationChannel	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container represents the configuration of the communication channel of XCP.	
Template Description	
This meta-class allows to describe the relationship between several PduTriggerings that are defined on the same Physical Channel, e.g. to create a link between Rx and Tx Pdu that are used for request/response.	
M2 Parameter	
GeneralPurposeConnection	
Mapping Rule	**Mapping Type**
For each GeneralPurposeConnection of category XcpChannel one XcpCommunicationChannel shall be created.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xcp_00183]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig
BSW Parameter	**BSW Type**
XcpPdu	ECUC-CHOICE-CONTAINER-DEF
BSW Description	
Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.	
Template Description	
This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template.	
M2 Parameter	
GeneralPurposeIPdu	
Mapping Rule	**Mapping Type**
Create this container if a GeneralPurposeIPdu with the category "Xcp" is defined in the Ecu Extract.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xcp_00100]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu
BSW Parameter	**BSW Type**
XcpRxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies received PDUs.	
Template Description	
M2 Parameter	
PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurpose Pdu that represents the XcpPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xcp_00105]

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpPdu
BSW Parameter	**BSW Type**
XcpTxPdu	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This container specifies transmission PDUs.	
Template Description	
M2 Parameter	
PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurpose Pdu that represents the XcpPdu.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xcp_00101]

C.61 Xfrm

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping/XfrmSignal/XfrmSignalChoice/XfrmISignalGroupRefChoice
BSW Parameter	**BSW Type**
XfrmISignalGroupRef	ECUC-FOREIGN-REFERENCE-DEF
BSW Description	
Reference to the ISignalGroup in the system description that transports the transformed data.	
Template Description	
SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)	
M2 Parameter	
ISignalGroup	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xfrm_00010]

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping/XfrmSignal/XfrmSignalChoice/XfrmISignalRefChoice
BSW Parameter	**BSW Type**
XfrmISignalRef	ECUC-FOREIGN-REFERENCE-DEF
BSW Description	
Reference to the ISignal in the system description that transports the transformed data.	
Template Description	
Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different Signal IPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup.	
M2 Parameter	
ISignal	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xfrm_00008]

BSW Module	BSW Context
Xfrm	Xfrm/XfrmImplementationMapping
BSW Parameter	**BSW Type**
XfrmTransformationTechnologyRef	ECUC-FOREIGN-REFERENCE-DEF

▽

△

BSW Description	
Reference to the TransformationTechnology in the DataTransformation of the system description for which the implementation (BswModuleEntry) shall be mapped.	
Template Description	
A TransformationTechnology is a transformer inside a transformer chain.	
M2 Parameter	
TransformationTechnology	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Xfrm_00003]

BSW Module	BSW Context	
Xfrm	Xfrm/XfrmImplementationMapping	
BSW Parameter		**BSW Type**
XfrmVariableDataPrototypeInstanceRef		ECUC-INSTANCE-REFERENCE-DEF
BSW Description		
Instance Reference to a VariableDataPrototype in case a dedicated transformer BswModuleEntry is required per Variable DataPrototype access.		
Template Description		
This attribute defines whether the Transformer has an internal state or not.		
M2 Parameter		
TransformationTechnology.hasInternalState		
Mapping Rule		**Mapping Type**
If an ISignal with dataTypePolicy set to transformingISignal - is received by this EcuInstance and - one of the TransformerTechnologys in the transformerChain has TransformationTechnology.has InternalState set to true and - the corresponding VariableDataPrototype is consumed via multiple RPorts then this XfrmVariableDataPrototypeInstanceRef shall be used.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_Xfrm_00011]

D Glossary

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

Artifact This is a Work Product Definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts ([55]).

At a high level, an artifact is represented as a single conceptual file.

AUTOSAR Tool This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an authoring tool, a converter tool, a processor tool or as a combination of those (see separate definitions).

AUTOSAR Authoring Tool An AUTOSAR Tool used to create and modify AUTOSAR XML Descriptions. Example: System Description Editor.

AUTOSAR Converter Tool An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files. Example: ECU Flattener

AUTOSAR Definition This is the definition of parameters which can have values. One could say that the parameter values are Instances of the definitions. But in the meta model hierarchy of AUTOSAR, definitions are also instances of the meta model and therefore considered as a description. Examples for AUTOSAR definitions are: `PortPrototype`, `PostBuildVariantCriterion`, `SwSystemconst`.

AUTOSAR XML Description In AUTOSAR this means “filled Template”. In fact an AUTOSAR XML description is the XML representation of an AUTOSAR model.

The AUTOSAR XML description can consist of several files. Each individual file represents an AUTOSAR partial model and shall validate successfully against the AUTOSAR XML schema.

AUTOSAR Meta-Model This is an UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR meta-model is an UML representation of the AUTOSAR templates. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes, UML tags and OCL expressions (object constraint language) are used for defining specific semantics and constraints.

AUTOSAR Meta-Model Tool The AUTOSAR Meta-Model Tool is the tool that generates different views (class tables, list of constraints, diagrams, XML Schema etc.) on the AUTOSAR meta-model.

AUTOSAR Model This is a representation of an AUTOSAR product. The AUTOSAR model represents aspects suitable to the intended use according to the AUTOSAR methodology.

Strictly speaking, this is an instance of the AUTOSAR meta-model. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR meta-model.

AUTOSAR Partial Model In AUTOSAR, the possible partitioning of models is marked in the meta-model by ≪atpSplitable≫. One partial model is represented in an AUTOSAR XML description by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.

AUTOSAR Processor Tool An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files. Example: RTE Generator

AUTOSAR Specification Element An AUTOSAR Specification Element is a named element that is part of an AUTOSAR specification. Examples: requirement, constraint, specification item, class or attribute in the meta model, methodology, deliverable, methodology activity, model element, bsw module etc.

AUTOSAR Template The term "Template" is used in AUTOSAR to describe the format different kinds of descriptions. The term template comes from the idea, that AUTOSAR defines a kind of form which shall be filled out in order to describe a model. The filled form is then called the description.

In fact the AUTOSAR templates are now defined as a meta-model.

AUTOSAR Validation Tool A specialized AUTOSAR Tool which is able to check an AUTOSAR model against the rules defined by a profile.

AUTOSAR XML Schema This is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR meta-model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.

Blueprint This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta model resp. types, this process is *not* an instantiation.

Instance Generally this is a particular exemplar of a model or of a type.

Life Cycle Life Cycle is the course of development/evolutionary stages of a model element during its life time.

Meta-Model This defines the building blocks of a model. In that sense, a Meta-Model represents the language for building models.

Meta-Data This includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc.

Model A Model is an simplified representation of reality. The model represents the aspects suitable for an intended purpose.

Partial Model This is a part of a model which is intended to be persisted in one particular artifact.

Pattern in GST This is an approach to simplify the definition of the meta model by applying a model transformation. This transformation creates an enhanced model out of an annotated model.

Property A property is a structural feature of an object. As an example a “connector” has the properties “receive port” and “send port”.

Properties are made variant by the ≪atpVariation≫.

Prototype This is the implementation of a role of a type within the definition of another type. In other words a type may contain Prototypes that in turn are typed by “Types”. Each one of these prototypes becomes an instance when this type is instantiated.

Type A type provides features that can appear in various roles of this type.

Value This is a particular value assigned to a “Definition”.

Variability Variability of a system is its quality to describe a set of variants. These variants are characterized by variant specific property settings and / or selections. As an example, such a system property selection manifests itself in a particular “receive port” for a connection.

This is implemented using the ≪atpVariation≫.

Variant A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no variability anymore with respect to the binding time.

This is implemented using EvaluatedVariantSet.

Variation Binding A variant is the result of a variation binding process that resolves the variability of the system by assigning particular values/selections to all the system’s properties.

This is implemented by VariationPoint.

Variation Binding Time The variation binding time determines the step in the methodology at which the variability given by a set of variable properties is resolved.

This is implemented by vh.latestBindingTime at the related properties.

Variation Definition Time The variation definition time determines the step in the methodology at which the variation points are defined.

Variation Point A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete variant.

This is implemented by VariationPoint.

E History of Constraints and Specification Items

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

E.1 Constraint and Specification Item History of this document according to AUTOSAR R4.0.1

E.1.1 Changed Constraints in R4.0.1

N/A

E.1.2 Added Constraints in R4.0.1

Number	Heading
[constr_3000]	valid SenderRecCompositeTypeMappings
[constr_3001]	valid ClientServerToSignalGroupMappings
[constr_3002]	valid SwcToImplMapping
[constr_3003]	Number of CAN channels
[constr_3004]	Clustering and separation must be exclusive
[constr_3005]	valid EcuResourceEstimation
[constr_3006]	valid EcuMapping
[constr_3007]	SelectorFieldCodes for dynamic part alternatives
[constr_3008]	EcuInstance subelements
[constr_3009]	Overlapping of ISignals is prohibited
[constr_3010]	ISignalIPdu shall not be exceeded
[constr_3011]	Overlapping of updateIndicationBits for ISignals is prohibited
[constr_3012]	Overlapping of Pdus is prohibited
[constr_3013]	Frame length shall not be exceeded
[constr_3014]	Overlapping of updateIndicationBits for Pdus is prohibited
[constr_3015]	Number of LIN channels
[constr_3016]	Number of Ethernet channels
[constr_3017]	Length of multiplexed Pdu shall not be exceeded
[constr_3018]	Number of FlexRay channels

Table E.1: Added Constraints in R4.0.1

E.1.3 Deleted Constraints in R4.0.1

N/A

E.2 Constraint and Specification Item History of this document according to AUTOSAR R4.0.2

E.2.1 Changed Constraints in R4.0.2

N/A

E.2.2 Added Constraints in R4.0.2

Number	Heading
[constr_3019]	In the flat ECU extract each required interface must be satisfied by connected provided interfaces

Table E.2: Added Constraints in R4.0.2

E.2.3 Deleted Constraints in R4.0.2

N/A

E.3 Constraint and Specification Item History of this document according to AUTOSAR R4.0.3

E.3.1 Changed Constraints in R4.0.3

N/A

E.3.2 Changed Specification Items in R4.0.3

N/A

E.3.3 Added Constraints in R4.0.3

Number	Heading
[constr_3020]	CommmunicationDirection of containedIPduGroups
[constr_3021]	Mapping of `SensorActuatorSwComponents` to SensorActuator `HwElements`
[constr_3024]	Usage of `triggeredWithoutRepetition` and `triggeredOnChangeWithoutRepetition` is not allowed for signal groups and group signals.
[constr_3025]	Usage of NPdus in TpConnections
[constr_3026]	valid `EmptySignalMappings`

Table E.3: Added Constraints in R4.0.3

E.3.4 Added Specification Items in R4.0.3

Number	Heading
[TPS_SYST_01000]	`FlatInstanceDescriptor` roles

Table E.4: Added Specification Items in 4.0.3

E.3.5 Deleted Constraints in R4.0.3

N/A

E.3.6 Deleted Specification Items in R4.0.3

N/A

E.4 Constraint and Specification Item History of this document according to AUTOSAR R4.1.1

E.4.1 Changed Constraints in R4.1.1

Number	Heading
[constr_3018]	Number of FlexRay channels

Table E.5: Changed Constraints in R4.1.1

E.4.2 Changed Specification Items in R4.1.1

N/A

E.4.3 Added Constraints in R4.1.1

Number	Heading
[constr_1198]	`TriggerToSignalMapping.systemSignal`s eligible for a `TriggerToSignalMapping`
[constr_1199]	`ISignal`s relating to `systemSignal`s eligible for a `TriggerToSignalMapping`
[constr_1206]	`DataMapping` to `PRPortPrototype`
[constr_1207]	Existence of the attribute `DataMapping`.communicationDirection in the context of a `SenderReceiverInterface` or `TriggerInterface`
[constr_1208]	Existence of the attribute `DataMapping`.communicationDirection in the context of a `ClientServerInterface`
[constr_1265]	`DoIpGidSynchronizationNeeds` can only exist once per `ECU_EXTRACT`

[constr_1266]	DoIpGidNeeds can only exist once per ECU_EXTRACT
[constr_1267]	DoIpActivationLineNeeds can only exist once per ECU_EXTRACT
[constr_3027]	Existence of ecuExtractVersion
[constr_3028]	FibexElements
[constr_3029]	Assign-Frame command usage
[constr_3030]	valid relationship between ECUMapping and EcuInstance
[constr_3031]	Complete System Description does not have outside ports
[constr_3032]	Combinations of SwcToEcuMapping targets
[constr_3033]	Criteria for primitive argument mapping
[constr_3034]	Values of LinSlaveConfig and LinSlave attributes
[constr_3035]	CanNm user data configuration in case NID/CBV are enabled
[constr_3036]	Pdus in CAN and LIN Frames
[constr_3037]	maximum Frame frameLength for CAN and LIN
[constr_3038]	maximum Frame frameLength for FlexRay
[constr_3039]	pncIdentifier range
[constr_3040]	Restriction of pncIdentifier values
[constr_3041]	pncVectorOffset range
[constr_3042]	pncVectorLength range
[constr_3043]	pncVector configuration in AUTOSAR Com
[constr_3044]	CBV configuration in case partial network is used
[constr_3045]	Signal content evaluation vs. Mode evaluation
[constr_3046]	Consistency of TransmissionModeCondition.iSignalInIPdu
[constr_3047]	Uniqueness of macMulticastAddresses
[constr_3048]	Range of vlanIdentifier
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs
[constr_3050]	J1939Cluster uses exactly one CanPhysicalChannel
[constr_3051]	Restriction of ISignalMapping references
[constr_3052]	Complete ISignalMapping of ISignalGroup signals
[constr_3053]	Complete ISignalMapping of target ISignalGroup
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3055]	SystemSignalGroup in a complete System Description
[constr_3056]	pduLength of the NmPdu
[constr_3057]	Maximal one BusspecificNmEcu per NmEcu and bus system is allowed to be defined
[constr_3058]	References from SenderRecArrayElementMapping and from SenderRecRecordElementMapping to SystemSignals are not allowed within a SenderReceiverCompositeElementToSignalMapping
[constr_3059]	Mandatory DataMapping on the receiver side for elements of a composite data type
[constr_3060]	Usage of networkRepresentationProps and physicalProps
[constr_3061]	CompuMethod specification in networkRepresentationProps
[constr_3062]	The EcuInstance that is referenced from a specific CouplingElement shall be connected to the same EthernetCluster as the specific CouplingElement
[constr_3063]	Usage of portNumber and dynamicallyAssigned with value “true” is mutually exclusive
[constr_3064]	Usage of DataMapping.serviceInstance, DataMapping.eventHandler and DataMapping.eventGroup references
[constr_3065]	Mapping of queued Triggers to SystemSignals is prohibited
[constr_3066]	Restriction of SenderComSpecs that refer to dataElements mapped to the same SystemSignal
[constr_3067]	initValue defined in the context of ISignal

[constr_3068]	DoIPPowerModeStatusNeeds in the category ECU_EXTRACT
[constr_3501]	Role of SystemSignal in 1:n communication
[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3505]	Criteria for primitive Data Mapping
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3508]	Value of nmReadySleepTime
[constr_3514]	No two ISignalToIPduMappings shall reference the identical ISignal

Table E.6: Added Constraints in R4.1.1

E.4.4 Added Specification Items in R4.1.1

Number	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01002]	System Category
[TPS_SYST_01003]	Standardized System Category Definitions
[TPS_SYST_01004]	Definition of AUTOSAR ECU
[TPS_SYST_01005]	Definition of EcuInstance
[TPS_SYST_01006]	Assign ECU type to EcuInstance
[TPS_SYST_01007]	Definition of CommunicationController
[TPS_SYST_01008]	Assign CommunicationController to the AUTOSAR Communication Peripheral
[TPS_SYST_01009]	Definition of CommunicationConnector
[TPS_SYST_01010]	Definition of CommunicationCluster
[TPS_SYST_01011]	Definition of PhysicalChannel
[TPS_SYST_01012]	Different Properties of LinMaster and LinSlave
[TPS_SYST_01013]	EcuInstance stands for its own
[TPS_SYST_01014]	Semantics of CommunicationControllerMapping
[TPS_SYST_01015]	Semantics of HwPortMapping
[TPS_SYST_01016]	System Extract, Ecu System Description and Ecu Extract may have ports
[TPS_SYST_01017]	The role of the top-level software composition
[TPS_SYST_01019]	Mapping of topology elements to elements of the ECU Resource Template
[TPS_SYST_01020]	Unconditional mapping of atomic Software Components
[TPS_SYST_01021]	Mapping of CompositionSwComponentType
[TPS_SYST_01022]	Prototype of a ParameterSwComponentType can be mapped to more than one ECU
[TPS_SYST_01023]	Prototype of an ServiceProxySwComponentType can be mapped to more than one ECU
[TPS_SYST_01024]	Component Clustering
[TPS_SYST_01025]	Clustering of Compositions
[TPS_SYST_01026]	Separation of Compositions
[TPS_SYST_01027]	Mapping of specific SW components to dedicated Ecus
[TPS_SYST_01028]	Task of the System Generator
[TPS_SYST_01029]	Mapping of specific SW components to exclusive Ecus
[TPS_SYST_01030]	Representation of VariableDataPrototypes and ClientServerOperations in System Description
[TPS_SYST_01032]	Independence of SystemSignals from CommunicationClusters
[TPS_SYST_01033]	DataMapping and SwConnector

[TPS_SYST_01034]	Data Mappings can be applied to compositions and atomic software components
[TPS_SYST_01035]	Transformation of Data Mappings during flattening
[TPS_SYST_01036]	No additional Data Mappings in composition substructure
[TPS_SYST_01037]	primitive Data Mapping of UINT8-Arrays
[TPS_SYST_01038]	Mapping of primitive arguments
[TPS_SYST_01039]	primitive Argument Mapping of UINT8-Arrays
[TPS_SYST_01040]	Mapping of composite arguments
[TPS_SYST_01041]	`CommonSignalPath` definition
[TPS_SYST_01042]	`ForbiddenSignalPath` definition
[TPS_SYST_01043]	`PermissibleSignalPath` definition
[TPS_SYST_01044]	`SeparateSignalPath` definition
[TPS_SYST_01045]	Component Seperation
[TPS_SYST_01046]	ShortNames of `LinSlaveConfig` and `LinSlave`
[TPS_SYST_01048]	Handling of large `IPdu`s
[TPS_SYST_01049]	Handling of `IPdu`s with dynamic signals
[TPS_SYST_01050]	`SystemSignal` in the System Extract and ECU Extract
[TPS_SYST_01051]	Handling of `DcmIPdu`s
[TPS_SYST_01052]	Routing of `UserDefinedPdu`s, `NmPdu`s, `NPdu`s
[TPS_SYST_01053]	Low-level routing of `NPdu`s
[TPS_SYST_01054]	Routing of `DcmIPdu`s
[TPS_SYST_01055]	Routing of `ISignalIPdu`s that are part of a `MultiplexedIPdu`
[TPS_SYST_01056]	Routing of `ISignalIPdu`s, `UserDefinedIPdu`s, `MultiplexedIPdu`s
[TPS_SYST_01057]	Routing of `NmPdu`s
[TPS_SYST_01058]	Pdu Gateway where an Ecu only routes a `PduTriggering` without being interested in the content
[TPS_SYST_01059]	Relationship between `FrameTriggering` and `CommConnectorPort`
[TPS_SYST_01060]	Relationship between `PduTriggering` and `CommConnectorPort`
[TPS_SYST_01061]	Relationship between `ISignalTriggering` and `CommConnectorPort`
[TPS_SYST_01062]	Network representation of an `ISignal`
[TPS_SYST_01063]	Context of network representation of an `ISignal`
[TPS_SYST_01064]	Transmit/Receive Semantics of Pdu Pools
[TPS_SYST_01065]	Mapping onto the of `ComSignalType` enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01068]	Bit Counting in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01070]	E2E Protection of `ISignalGroup`s
[TPS_SYST_01071]	E2E Protection of several `ISignalGroup`s in one `ISignalIPdu`
[TPS_SYST_01072]	Offset attributes of EndToEndDescription
[TPS_SYST_01073]	E2E Protection via COM Callouts
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_01075]	Signal content evaluation via `TransmissionModeCondition`
[TPS_SYST_01076]	Mode evaluation via `modeDrivenTrueCondition`
[TPS_SYST_01077]	Mapping of Com Transmission Modes to System Template elements
[TPS_SYST_01078]	Dynamic Part of a `MultiplexedIPdu`
[TPS_SYST_01079]	Static Part of a `MultiplexedIPdu`
[TPS_SYST_01080]	Sending or receiving of a `MultiplexedIPdu` in System Extract/ECU Extract
[TPS_SYST_01081]	Gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract
[TPS_SYST_01082]	Receiving and gatewaying of a `MultiplexedIPdu` in System Extract/ECU Extract

[TPS_SYST_01083]	A Frame represents a general design object that is used to describe the layout of the included Pdus as a reusable asset.
[TPS_SYST_01084]	FrameTriggering
[TPS_SYST_01085]	Transmission of a Frame multiple times within one communication cycle
[TPS_SYST_01086]	Number of Ethernet channels
[TPS_SYST_01087]	Role of SystemSignal in inter-ECU client server communication with clients located on the same ECU
[TPS_SYST_01088]	NetworkEndpoint priority
[TPS_SYST_01089]	ApplicationEndpoint priority
[TPS_SYST_01090]	valid NetworkEndpoint
[TPS_SYST_01091]	Definition of SoAdConfig
[TPS_SYST_01092]	Transmission of multiple Pdus over the same SocketConnection
[TPS_SYST_01093]	Activation/Deactivation of SoAdRoutingGroups
[TPS_SYST_01094]	allowed key/value CapabilityRecord combinations
[TPS_SYST_01095]	tagged VLANs
[TPS_SYST_01096]	untagged VLANs
[TPS_SYST_01097]	Assignment of CouplingPorts to a VLAN
[TPS_SYST_01098]	Assignment of CouplingPorts to an “untagged” VLAN
[TPS_SYST_01099]	Context of TpConfig
[TPS_SYST_01100]	TP routing using the same transport protocol
[TPS_SYST_01101]	TP routing using different transport protocols
[TPS_SYST_01102]	FlexrayTpConnectionControl reuse
[TPS_SYST_01103]	FlexrayTpConnection shall specify one txPduPool
[TPS_SYST_01104]	FlexrayTpConnection with several receivers
[TPS_SYST_01105]	CanTpConnection with several receivers
[TPS_SYST_01106]	Usage of additional directPdu in case of variable length sdu
[TPS_SYST_01107]	Definition of NmCoordinator
[TPS_SYST_01108]	ProvidedServiceInstance priority
[TPS_SYST_01109]	RTE fan-out support
[TPS_SYST_01110]	Com Signal Gateway fan-out support
[TPS_SYST_01111]	Pdu Router fan-out support
[TPS_SYST_01112]	FlexRay dual channel Pdu Router interaction
[TPS_SYST_01113]	FlexRay Interface fan-out support
[TPS_SYST_01114]	Frame fan-out support
[TPS_SYST_01115]	CDD communication support
[TPS_SYST_01116]	Frame Mapping is not supported by the AUTOSAR BSW
[TPS_SYST_01117]	Pdu Gateway support
[TPS_SYST_01118]	Support of Multicast Pdu routing
[TPS_SYST_01119]	Signal Gateway support
[TPS_SYST_01120]	Precedence of ISignalMappings
[TPS_SYST_01121]	Support of Mulicast signal routing
[TPS_SYST_01122]	partial routing between ISignalGroups
[TPS_SYST_01123]	System Extract may cover one or many EcuInstances
[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in
[TPS_SYST_01126]	Resource Consumption for RTE and basic software
[TPS_SYST_01127]	CDD Topology support
[TPS_SYST_01128]	Communication over FlexRay
[TPS_SYST_01129]	Communication over LIN
[TPS_SYST_01130]	Communication over CAN
[TPS_SYST_01131]	TCP/IP and UDP/IP communication over Ethernet
[TPS_SYST_01132]	Communication over SAE J1939
[TPS_SYST_01133]	Partial Network Clusters

[TPS_SYST_01134]	Abstract System Description
[TPS_SYST_01135]	Refactoring of an Abstract System Description into a project specific technical view of the software architecture
[TPS_SYST_01136]	ViewMapSet and ViewMap are used to trace the transformations between different models
[TPS_SYST_01137]	Several DataMappings may be defined for the same SystemSignal
[TPS_SYST_01138]	Low-level routing of XcpPdus
[TPS_SYST_01139]	Ecu Extract covers exactly one EcuInstance
[TPS_SYST_01140]	Ecu Extract contains only SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype
[TPS_SYST_01141]	Derivation of ComSignalType
[TPS_SYST_01142]	Rules for the creation of Triggerings and Ports on the sender side
[TPS_SYST_01143]	DataMapping on the sender side for elements of a composite data type
[TPS_SYST_01144]	Physical properties of a System Signal
[TPS_SYST_01145]	PortInterfaceMappings in the ECU Extract
[TPS_SYST_01146]	Generic CanTpConnections
[TPS_SYST_01147]	Generic J1939TpConnections
[TPS_SYST_01148]	Mapping of IN and INOUT ArgumentDataPrototypes to callSignals
[TPS_SYST_01149]	Mapping of OUT and INOUT ArgumentDataPrototypes to returnSignals
[TPS_SYST_01150]	Mapping of returnSignal and callSignal to COM Signal
[TPS_SYST_01151]	DataMapping reference to an EventHandler
[TPS_SYST_01152]	DataMapping reference to a ConsumedEventGroup
[TPS_SYST_01153]	Atomic transport of SystemSignalGroups
[TPS_SYST_01154]	CAN Controller support of CAN FD frames
[TPS_SYST_03000]	Co-existing System with category SYSTEM_DESCRIPTION and System with category SYSTEM_EXTRACT
[TPS_SYST_05000]	System Description doesn't use a complete Software Component Description
[TPS_SYST_05001]	Send a Trigger across a network
[TPS_SYST_05002]	The value of startPosition is irrelevant

Table E.7: Added Specification Items in 4.1.1

E.4.5 Deleted Constraints in R4.1.1

[constr_3016]	Number of Ethernet channels

Table E.8: Deleted Constraints in R4.1.1

E.4.6 Deleted Specification Items in R4.1.1

N/A

E.5 Constraint and Specification Item History of this document according to AUTOSAR R4.1.2

E.5.1 Changed Specification Items in R4.1.2

Number	Heading
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus
[TPS_SYST_01056]	Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus
[TPS_SYST_01138]	Low-level routing of XcpPdus

Table E.9: Added Specification Items in 4.1.2

E.5.2 Added Specification Items in R4.1.2

Number	Heading
[TPS_SYST_02001]	networkRepresentationProps are mandatory in case the dataTypePolicy is set to override or legacy
[TPS_SYST_02002]	SoAdRoutingGroup for Services with Methods
[TPS_SYST_02003]	SoAdRoutingGroups for Services with event groups
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus
[TPS_SYST_02006]	Usage of networkRepresentationFromComSpec
[TPS_SYST_02007]	Usage of SocketConnection attributes in the unicast server view
[TPS_SYST_02008]	Usage of SocketConnection attributes in the unicast client view
[TPS_SYST_02009]	Usage of SocketConnection attributes in the multicast server view
[TPS_SYST_02010]	Usage of SocketConnection attributes in the multicast client view

Table E.10: Added Specification Items in 4.1.2

E.5.3 Added Constraints in R4.1.2

Number	Heading
[constr_3069]	Allowed CanNmCluster.nmNidPosition values
[constr_3070]	Allowed CanNmCluster.nmCbvPosition values
[constr_3071]	CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value
[constr_3073]	nmVoteInformation only valid for FrNm
[constr_3074]	No TransmissionAcknowledgementRequest for multiple senders
[constr_3078]	Allowed UdpNmCluster.nmNidPosition values
[constr_3079]	Allowed UdpNmCluster.nmCbvPosition values
[constr_3080]	UdpNmCluster.nmCbvPosition and UdpNmCluster.nmNidPosition shall never have the same value
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3083]	Exactly one AtomicSwComponentType on an EcuInstance may use GeneralCallbackEventDataChanged / GeneralCallbackEventStatusChange
[constr_3084]	Service port in the role PowerTakeOff

[constr_3085]	Service port in the role CallbackDCMRequestServices

Table E.11: Added Constraints in R4.1.2

E.5.4 Changed Constraints in R4.1.2

Number	Heading
[constr_2025]	Uniqueness of `symbol` attributes

Table E.12: Changed Constraints in R4.1.2

E.5.5 Deleted Constraints in R4.1.2

[constr_3066]	Restriction of `SenderComSpec`s that refer to `dataElement`s mapped to the same `SystemSignal`

Table E.13: Deleted Constraints in R4.1.2

E.6 Constraint and Specification Item History of this document according to AUTOSAR R4.1.3

E.6.1 Changed Specification Items in R4.1.3

N/A

E.6.2 Added Specification Items in R4.1.3

Number	Heading
[TPS_SYST_01155]	Routing of `ISignalGroup`s
[TPS_SYST_01156]	Definition of `ISignalTriggering`s is allowed for `ISignalGroup`s and for `GroupSignal`s
[TPS_SYST_01157]	Allowed usage of attributes for `ISignal`s, `ISignalGroup`s and `GroupSignal`s
[TPS_SYST_02011]	`initValue`s of receivers that are mapped to the same Ecu
[TPS_SYST_02012]	`initValue` and `invalidValue` represent internal values

Table E.14: Added Specification Items in 4.1.3

E.6.3 Deleted Specification Items in R4.1.3

Number	Heading

[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in

Table E.15: Deleted Specification Items in 4.1.3

E.6.4 Added Constraints in R4.1.3

[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3087]	DataMapping to PRPortPrototype
[constr_3088]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3090]	TpSdu transmission on a PhysicalChannel
[constr_3094]	Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals

Table E.16: Added Constraints in R4.1.3

E.6.5 Changed Constraints in R4.1.3

[constr_3051]	Restriction of ISignalMapping references

Table E.17: Changed Constraints in R4.1.3

E.6.6 Deleted Constraints in R4.1.3

[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_1206]	DataMapping to PRPortPrototype
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description

Table E.18: Deleted Constraints in R4.1.3

E.7 Constraint and Specification Item History of this document according to AUTOSAR R4.2.1

E.7.1 Added Specification Items in R4.2.1

Id	Heading

[TPS_SYST_02013]	Usage of dataFilters on GroupSignals on receiver side
[TPS_SYST_02014]	ConsumedEventGroup priority
[TPS_SYST_02015]	LdCom: only one ISignal mapped to the ISignalIPdu
[TPS_SYST_02016]	LdCom: only Transformer output and UINT8_N or UINT8_DYN supported
[TPS_SYST_02017]	LdCom: Opaque ISignalToIPduMapping.packingByteOrder
[TPS_SYST_02018]	LdCom: ISignalToIPduMapping.startPosition shall be 0
[TPS_SYST_02019]	LdCom: ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition
[TPS_SYST_02020]	LdCom: No IPduTiming.minimumDelay defined
[TPS_SYST_02021]	LdCom: ISignalToIPduMapping.updateIndicationBitPosition shall not be defined
[TPS_SYST_02022]	LdCom: Only the transmissionModeTrueTiming defined
[TPS_SYST_02023]	LdCom: DataFilter "always" if TransmissionModeCondition defined
[TPS_SYST_02024]	LdCom: No ModeDrivenTransmissionModeCondition defined
[TPS_SYST_02025]	LdCom: Only EventControlledTiming defined
[TPS_SYST_02026]	LdCom: Only EventControlledTiming with no repetition defined
[TPS_SYST_02027]	LdCom: No ISignalPort.timeout reception timeout defined
[TPS_SYST_02028]	LdCom: No ISignalPort.dataFilter defined
[TPS_SYST_02029]	Multiple ParameterDataPrototype instances in an EcuExtract
[TPS_SYST_02030]	The DataTransformationSet contains all transformer chains
[TPS_SYST_02031]	A transformer is represented by a TransformationTechnology
[TPS_SYST_02032]	Transformer chains are ordered list of transformers
[TPS_SYST_02033]	Order of the transformers in the configuration represents the order on the sending side
[TPS_SYST_02034]	Order of the transformers on the receiving side is the reverse of the sending side
[TPS_SYST_02035]	protocol contains the human readable protocol identifier
[TPS_SYST_02036]	version contains the version of the protocol
[TPS_SYST_02037]	The attribute needsOriginalData configures a transformer's access to the original data
[TPS_SYST_02038]	Specification of transformer class
[TPS_SYST_02039]	Specification of transformer specific properties
[TPS_SYST_02040]	Specification of transformer buffer handling
[TPS_SYST_02041]	In-place buffer handling of transformers
[TPS_SYST_02042]	Header length to be considered by transformers
[TPS_SYST_02043]	Buffer computation of transformer
[TPS_SYST_02044]	Buffer computation of transformer
[TPS_SYST_02045]	SOME/IP Transformer configuration
[TPS_SYST_02046]	E2E Transformer configuration
[TPS_SYST_02047]	Custom transformer configuration
[TPS_SYST_02048]	ISignal specific transformation configuration
[TPS_SYST_02049]	Transformer specific TransformationISignalProps
[TPS_SYST_02050]	ISignal specific configuration of the SOME/IP Transformer
[TPS_SYST_02051]	ISignal specific configuration of the E2E Transformer
[TPS_SYST_02052]	ISignal specific configuration of custom transformers
[TPS_SYST_02053]	A reference from ISignal to DataTransformation in the role dataTransformation enables data transformation
[TPS_SYST_02054]	Definition of data which shall be transformed
[TPS_SYST_02055]	Alignment of SOME/IP
[TPS_SYST_02056]	Byte Order of SOME/IP
[TPS_SYST_02057]	Interface Version of SOME/IP
[TPS_SYST_02058]	Usage of COM Based Transformer
[TPS_SYST_02059]	Routing of SecuredIPdus

[TPS_SYST_02060]	SecuredIPdus
[TPS_SYST_02061]	Routing of IPdus that are part of a ContainerIPdu
[TPS_SYST_02062]	Allowed ContainedIPduProps.headerIdLongHeader and ContainedIPduProps.headerIdShortHeader values
[TPS_SYST_02063]	Byte order of ContainerIPdu header information
[TPS_SYST_02064]	Reception acceptance of contained IPdus
[TPS_SYST_02065]	Contained IPdu specific transmission timeout
[TPS_SYST_02066]	ContainerIPdu.thresholdSize
[TPS_SYST_02067]	E2E profile
[TPS_SYST_02068]	E2E header field representation in an ISignalGroup
[TPS_SYST_02069]	Recommended configuration settings for E2E Profile 1 configuration setting C
[TPS_SYST_02070]	Recommended configuration settings for E2E Profile 4 configuration setting A
[TPS_SYST_02071]	Recommended configuration settings for E2E Profile 4 configuration setting B
[TPS_SYST_02072]	profileName of EndToEndTransformationDescription
[TPS_SYST_02073]	EndToEndTransformationDescription.profileName
[TPS_SYST_02074]	Precedence of transformer configuration settings
[TPS_SYST_02075]	Mandatory attributes in transformer configuration elements
[TPS_SYST_03001]	LdCom: ISignalIPdu not part of any ISignalIPduGroup
[TPS_SYST_03002]	Keep behavior of DHCP clients
[TPS_SYST_03003]	Ethernet priority regeneration
[TPS_SYST_03004]	VLAN specific sending behavior
[TPS_SYST_03005]	VLAN re-tagging
[TPS_SYST_03006]	Ethernet switch egress port setup
[TPS_SYST_03007]	Ethernet port scheduler algorithm
[TPS_SYST_03008]	Ethernet port scheduler priority
[TPS_SYST_03009]	Ethernet port shaper idleSlope
[TPS_SYST_03010]	Ethernet switch packet to traffic class assignment
[TPS_SYST_03011]	Ethernet switch traffic class to FIFO assignment
[TPS_SYST_03013]	Semi-static DHCP server configuration
[TPS_SYST_03014]	Transmission triggering by the first contained IPdu put into a ContainerIPdu
[TPS_SYST_05003]	Usage of DiagnosticConnection in combination with a TP
[TPS_SYST_05004]	Usage of DiagnosticConnection in combination with UUDT
[TPS_SYST_05005]	Relation of GlobalTimeDomain to CommunicationCluster
[TPS_SYST_05006]	Chaining of GlobalTimeDomains
[TPS_SYST_05007]	separation of roles within a GlobalTimeDomain
[TPS_SYST_05008]	Semantics of a GlobalTimeGateway
[TPS_SYST_05009]	Pdu for transmitting global time information
[TPS_SYST_05010]	Pdu is not required on Ethernet
[TPS_SYST_05011]	Ownership of GlobalTimeGateway
[TPS_SYST_05013]	Semantics of GlobalTimeMaster.isSystemWideGlobalTimeMaster
[TPS_SYST_05014]	GlobalTimeMaster.isSystemWideGlobalTimeMaster
[TPS_SYST_05015]	Naming conventions

Table E.19: Added Traceables in R4.2.1

E.7.2 Changed Specification Items in R4.2.1

Id	Heading
[TPS_SYST_01024]	Component Clustering
[TPS_SYST_01025]	Clustering of Compositions
[TPS_SYST_01026]	Separation of Compositions
[TPS_SYST_01045]	Component Seperation
[TPS_SYST_01056]	Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus, ContainerIPdus
[TPS_SYST_01057]	Routing of NmPdus
[TPS_SYST_01088]	NetworkEndpoint priority
[TPS_SYST_01089]	ApplicationEndpoint priority
[TPS_SYST_01106]	Usage of additional directPdu in case of variable length sdu
[TPS_SYST_01108]	ProvidedServiceInstance priority
[TPS_SYST_01138]	Low-level routing of XcpPdus
[TPS_SYST_01157]	Allowed usage of attributes for ISignals, ISignalGroups and GroupSignals
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus

Table E.20: Changed Traceables in R4.2.1

E.7.3 Deleted Specification Items in R4.2.1

Id	Heading
[TPS_SYST_01038]	Mapping of primitive arguments
[TPS_SYST_01039]	primitive Argument Mapping of UINT8-Arrays
[TPS_SYST_01040]	Mapping of composite arguments
[TPS_SYST_01051]	Handling of DcmIPdus

Table E.21: Deleted Traceables in R4.2.1

E.7.4 Added Constraints in R4.2.1

Id	Heading
[constr_1367]	periodicResponseUudt.periodicResponseUudt shall only refer to a DcmIPdu
[constr_1368]	Limitation of the target of references from DiagnosticConnection
[constr_1369]	CommunicationConnectors shall be attached to the same CommunicationCluster
[constr_1370]	Consistency of GlobalTimeDomain
[constr_1371]	Consistency of attribute host
[constr_1372]	Consistency of attribute globalTimePdu
[constr_1373]	GlobalTimeMaster with attribute isSystemWideGlobalTimeMaster set to TRUE
[constr_1374]	Only fan-out possible for GlobalTimeGateway
[constr_3095]	canControllerFdAttributes and canControllerFdRequirements are mutually exclusive.
[constr_3096]	Allowed values for diagnosticMessageType
[constr_3097]	Overlapping of segments of one MultiplexedIPdu is not allowed
[constr_3098]	Defined segments of one MultiplexedIPdu shall not exceed the length of the MultiplexedIPdu

[constr_3099]	Defined segments in a DynamicPart shall not exceed the length of any DynamicPartAlternative.iPdu
[constr_3100]	Defined segments in a StaticPart shall not exceed the length of the StaticPart.iPdu
[constr_3101]	Signal representation of selector field for DynamicPartAlternative
[constr_3102]	Restriction on usage of J1939NodeName attributes
[constr_3103]	Range of ecuInstance
[constr_3104]	Range of function
[constr_3105]	Range of functionInstance
[constr_3106]	Range of identitiyNumber
[constr_3107]	Range of industryGroup
[constr_3108]	Range of manufacturerCode
[constr_3109]	Range of vehicleSystem
[constr_3110]	Range of vehicleSystemInstance
[constr_3111]	returnSignal in ClientServerToSignalMapping is mandatory
[constr_3112]	Invalidation support for partial mapping of a data element typed by composite data type
[constr_3113]	EthernetFrame shall not have a PduToFrameMapping
[constr_3114]	FlatInstanceDescriptors pointing to the same ParameterDataPrototype shall have different postBuildVariantConditions
[constr_3115]	FlatInstanceDescriptors pointing to the same ParameterDataPrototype instance
[constr_3116]	Overlap of ClientIdRanges in the context of the enclosing System
[constr_3117]	Allowed value of attribute clientId
[constr_3118]	Valid reference target for ClientIdDefinition.clientServerOperation.contextPort
[constr_3121]	The length of transformer chains is limited to 255 transformers
[constr_3122]	At most one transformer of each transformer class inside a transformer chain
[constr_3123]	Serializer transformer shall be the first in a chain
[constr_3124]	Applicability of needsOriginalData
[constr_3125]	Value of attribute inPlace for the first transformer in a chain
[constr_3126]	headerLength shall be less or equal output buffer size
[constr_3127]	Certain ISignals always need a reference to DataTransformation
[constr_3128]	SOME/IP transformer configuration
[constr_3129]	Byte Order of SOME/IP transformer
[constr_3130]	Range of Interface Version
[constr_3131]	Required first data transformation for comBasedSignalGroupTransformation
[constr_3132]	Required COM Based Transformation for comBasedSignalGroupTransformation
[constr_3133]	physicalLayerType of connected CouplingPorts
[constr_3134]	The connection of two CouplingPorts with connectionNegotiationBehavior set to master is forbidden
[constr_3135]	The connection of two CouplingPorts with connectionNegotiationBehavior set to slave is forbidden
[constr_3136]	Allowed payload of SecuredIPdus
[constr_3137]	IPduPort.rxSecurityVerification is configurable on the receiver side
[constr_3138]	IPduPort.rxSecurityVerification validness
[constr_3139]	Usage of IPduPort.rxSecurityVerification
[constr_3140]	No ByteOrderEnum.opaque allowed for System.containerIPduHeaderByteOrder
[constr_3141]	Only IPdus shall be part of a ContainerIPdu
[constr_3142]	Mandatory headerIdLongHeader for longHeader
[constr_3143]	Mandatory headerIdShortHeader for shortHeader

[constr_3144]	Mandatory IPdu.containedIPduProps for contained IPdus
[constr_3146]	Partial Networking timing constraint
[constr_3148]	executeDespiteDataUnavailability setting in case an E2E Transformer is used
[constr_3149]	TransformationTechnology settings for E2E Transformer
[constr_3150]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01 in case it is 0
[constr_3151]	BufferProperties.headerLength settings for an E2E transformer used in combination with a SOME/IP transformer
[constr_3152]	BufferProperties.headerLength settings for an E2E transformer used in combination with a COM Based transformer
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3154]	BufferProperties.bufferComputation setting for an E2E transformer
[constr_3155]	Allowed values for EndToEndTransformationDescription.upperHeaderBitsToShift
[constr_3156]	Allowed values for EndToEndTransformationISignalProps.dataId in PROFILE_01
[constr_3157]	Allowed values for EndToEndTransformationISignalProps.dataId in PROFILE_01 in case dataIdMode is set to lower12Bit
[constr_3158]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_01
[constr_3159]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_04
[constr_3160]	EndToEndTransformationISignalProps.dataId in PROFILE_02
[constr_3161]	EndToEndTransformationISignalProps.dataLength in PROFILE_01, PROFILE_02, PROFILE_05
[constr_3162]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_01, PROFILE_02, PROFILE_05
[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04 and PROFILE_06
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04 and PROFILE_06
[constr_3165]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01
[constr_3166]	EndToEndTransformationDescription.upperHeaderBitsToShift in PROFILE_02
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3169]	Attribute multiplicities and values in PROFILE_02
[constr_3171]	Value of EndToEndTransformationISignalProps.dataId shall be unique in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3172]	Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_01
[constr_3173]	Effect of EndToEndTransformationDescription.profileBehavior value in PROFILE_02
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3176]	Value range of windowSize
[constr_3177]	Dependency between maxErrorStateValid, maxErrorStateInit and maxErrorStateInvalid
[constr_3178]	Dependency between minOkStateValid, minOkStateInit and minOkStateInvalid

[constr_3179]	Dependency between minOkStateInit, maxErrorStateInit and windowSizeStateInit
[constr_3180]	Dependency between minOkStateValid, maxErrorStateValid and windowSizeStateValid
[constr_3181]	Dependency between minOkStateInvalid, maxErrorStateInvalid and windowSizeStateInvalid
[constr_3182]	Restriction on TransformationTechnology.transformationDescriptionVariationPoint
[constr_3183]	ISignalGroup with transformationISignalProps
[constr_3184]	Only one EndToEndTransformationISignalProps.dataId element in PROFILE_01
[constr_3185]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_01
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3187]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_01
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3189]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_01
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06
[constr_3191]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_01 and dataIdMode equal to lower12Bit
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05 and PROFILE_06 or dataIdMode different from lower12Bit
[constr_3193]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_01
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in Profiles different from PROFILE_01
[constr_3195]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02
[constr_3196]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_05
[constr_3197]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_06
[constr_3515]	Fully filled EthernetPriorityRegeneration table
[constr_3516]	limitation of Pdu.length for CAN L-PDUs
[constr_3517]	Consistent setting of ContainedIPduProps.collectionSemantics in the context of one ContainerIPdu
[constr_3518]	Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Table E.22: Added Constraints in R4.2.1

E.7.5 Changed Constraints in R4.2.1

Id	Heading
[constr_3010]	ISignalIPdu length shall not be exceeded
[constr_3011]	Overlapping of updateIndicationBits of ISignals is prohibited
[constr_3028]	FibexElements

[constr_3037]	maximum Frame frameLength for CAN and LIN
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup

Table E.23: Changed Constraints in R4.2.1

E.7.6 Deleted Constraints in R4.2.1

Id	Heading
[constr_1208]	Existence of the attribute DataMapping.communicationDirection in the context of a ClientServerInterface
[constr_3001]	valid ClientServerToSignalGroupMappings
[constr_3017]	Length of multiplexed Pdu shall not be exceeded.
[constr_3026]	valid EmptySignalMappings
[constr_3033]	Criteria for primitive argument mapping
[constr_3056]	pduLength of the NmPdu

Table E.24: Deleted Constraints in R4.2.1

E.8 Constraint and Specification Item History of this document according to AUTOSAR R4.2.2

E.8.1 Added Specification Items in R4.2.2

Id	Heading
[TPS_SYST_02076]	networkRepresentationProps in case the dataTypePolicy is set to transformingISignal
[TPS_SYST_02077]	Subscribers of a LinEventTriggeredFrame
[TPS_SYST_02078]	LinUnconditionalFrames associated with a LinEventTriggered-Frame
[TPS_SYST_02079]	Identification of ImplementationDataType for a given ISignal in an Ecu Extract
[TPS_SYST_02080]	Message type of SOME/IP
[TPS_SYST_02081]	PduTriggering that is used for ClientServer Communication
[TPS_SYST_02082]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE or BOOLEAN and a DataTypeMap exists
[TPS_SYST_02083]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and a Data-TypeMap exists
[TPS_SYST_02084]	SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and a DataTypeMap exists
[TPS_SYST_02085]	SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category ARRAY
[TPS_SYST_02086]	SenderReceiverInterface.dataElement is typed by an ImplementationDataType of category VALUE or TYPE_REFERENCE

[TPS_SYST_02087]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category BOOLEAN and no DataTypeMap exists
[TPS_SYST_02088]	SenderReceiverInterface.dataElement is typed by an ApplicationArrayDataType and no DataTypeMap exists
[TPS_SYST_02089]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category STRING and no DataTypeMap exists
[TPS_SYST_02090]	SenderReceiverInterface.dataElement is typed by an ApplicationPrimitiveDataType of category VALUE and no DataTypeMap exists
[TPS_SYST_02091]	Routing of GeneralPurposePdus with category SD and GeneralPurposePdus with category DoIP
[TPS_SYST_02092]	Size of Fixed-size Array Length Fields
[TPS_SYST_02093]	Size of Structure Length Fields
[TPS_SYST_02094]	Size of Union Length Fields
[TPS_SYST_03015]	Offset time domain requires synchronized time domain

Table E.25: Added Traceables in R4.2.2

E.8.2 Changed Specification Items in R4.2.2

Id	Heading
[TPS_SYST_01003]	Standardized System Category Definitions
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus with category GLOBAL_TIME
[TPS_SYST_01065]	Mapping onto the ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01077]	Mapping of Com Transmission Modes to System Template elements
[TPS_SYST_01113]	FlexRay Interface fan-out support
[TPS_SYST_01157]	Allowed usage of attributes for ISignals, ISignalGroups and GroupSignals
[TPS_SYST_02017]	LdCom: Opaque ISignalToIPduMapping.packingByteOrder
[TPS_SYST_02069]	Recommended configuration settings for E2E Profile 1 configuration setting C
[TPS_SYST_02070]	Recommended configuration settings for E2E Profile 4 configuration setting A
[TPS_SYST_02071]	Recommended configuration settings for E2E Profile 4 configuration setting B

Table E.26: Changed Traceables in R4.2.2

E.8.3 Deleted Specification Items in R4.2.2

none

E.8.4 Added Constraints in R4.2.2

Id	Heading
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_1387]	Transmission of Variable-Size Array Data Types by means of a Transformer
[constr_3198]	Uniqueness of PncMapping.shortLabel
[constr_3199]	ISignal that has dataTypePolicy set to transformingISignal shall reference a DataTransformation
[constr_3201]	eventGroupIdentifier in ConsumedEventGroups that are referenced by the same EventHandler
[constr_3202]	LinFrameTriggering to LinUnconditionalFrame reference restriction in LinEventTriggeredFrame context
[constr_3203]	LinFrameTriggering to LinSporadicFrame reference restriction in LinSporadicFrame context
[constr_3204]	LinUnconditionalFrames associated with a LinSporadicFrame
[constr_3205]	Existence of FramePort for a FrameTriggering that references a LinSporadicFrame
[constr_3206]	Existence of FramePort for a FrameTriggering that references a LinEventTriggeredFrame
[constr_3207]	Assignment of SocketConnectionIpduIdentifiers used for ClientServer Communication to SocketConnections
[constr_3208]	executeDespiteDataUnavailability usage restriction
[constr_3209]	CanFrameTriggerings with identical PGN
[constr_3210]	J1939TpPgs with identical pgn value
[constr_3211]	PduTriggerings with triggerIPduSendCondition
[constr_3212]	Limitation of DoIpTpConnection.tpSdu
[constr_3213]	TransformationISignalProps.csErrorReaction setting in case that the serializer transformerClass and Client/Server communication is used
[constr_3214]	TransformationISignalProps.csErrorReaction setting in case that a transformerClass different from serializer is used or the Client/Server communication is not used
[constr_3215]	TransformationTechnology.version and TransformationTechnology.protocol settings for request and response of a client/server communication
[constr_3216]	Usage of SOMEIPTransformationISignalProps.sessionHandlingSR
[constr_3218]	Range of Size of Fixed-size Array Length Fields
[constr_3219]	The existence of LinSlaves in the LinMaster EcuExtract
[constr_3220]	Range of Size of Structure Length Fields
[constr_3221]	Range of Size of Union Length Fields
[constr_3519]	Value of category of GlobalTimeDomain
[constr_3520]	Offset time domain shall be based on a synchronized time domain

Table E.27: Added Constraints in R4.2.2

E.8.5 Changed Constraints in R4.2.2

Id	Heading
[constr_1368]	Limitation of the target of references from DiagnosticConnection
[constr_1374]	Only fan-out possible for GlobalTimeGateway
[constr_3002]	valid swcToImplMapping
[constr_3003]	Number of CAN channels
[constr_3004]	Clustering and separation must be exclusive
[constr_3005]	valid EcuResourceEstimation

[constr_3006]	valid EcuMapping
[constr_3007]	selectorFieldCodes for dynamic part alternatives
[constr_3008]	EcuInstance subelements
[constr_3015]	Number of LIN channels
[constr_3018]	Number of FlexRay channels
[constr_3019]	In the flat ECU extract each required interface must be satisfied by connected provided interfaces
[constr_3020]	communicationDirection of containedIPduGroups
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3025]	Usage of NPdus in TpConnections
[constr_3027]	Existence of ecuExtractVersion
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3095]	canControllerFdAttributes and canControllerFdRequirements are mutually exclusive
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3179]	Dependency between minOkStateInit, maxErrorStateInit and windowSize
[constr_3180]	Dependency between minOkStateValid, maxErrorStateValid and windowSize
[constr_3181]	Dependency between minOkStateInvalid, maxErrorStateInvalid and windowSize
[constr_3501]	Role of SystemSignal in 1:n communication
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3518]	Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue

Table E.28: Changed Constraints in R4.2.2

E.8.6 Deleted Constraints in R4.2.2

Id	Heading
[constr_3131]	Required first data transformation for comBasedSignalGroupTransformation
[constr_3505]	Criteria for primitive Data Mapping

Table E.29: Deleted Constraints in R4.2.2

E.9 Constraint and Specification Item History of this document according to AUTOSAR R4.3.0

E.9.1 Added Specification Items in R4.3.0

Id	Heading
[TPS_SYST_02095]	LinFrameTriggering.linChecksum for LinUnconditionalFrames
[TPS_SYST_02096]	Sending of ANY finds for minor version
[TPS_SYST_02097]	Basic definition of contained IPdus
[TPS_SYST_02098]	Header id and header type of a contained IPdu
[TPS_SYST_02099]	Relation between ContainerIPdu and contained IPdus on sender side
[TPS_SYST_02100]	Relation between ContainerIPdu and contained IPdus on receiver side
[TPS_SYST_02101]	Usage of LinSlaveConfig in Ecu Extract
[TPS_SYST_02102]	FrameTriggering.pduTriggering references that shall be ignored
[TPS_SYST_02103]	Semantics of GlobalTimeDomain.domainId
[TPS_SYST_02104]	Triggerings on PhysicalChannel
[TPS_SYST_02105]	ISignalGroup and ISignal referenced from ISignalTriggering
[TPS_SYST_02106]	Rules for the creation of references to Ports (ecuCommPortInstance) with communicationDirection in on receiving Ecu
[TPS_SYST_02107]	Shared address space for J1939 routing relations
[TPS_SYST_02108]	Address proxying for J1939 routing relations
[TPS_SYST_02109]	Absence of J1939SharedAddressCluster.participatingJ1939Cluster to a J1939Cluster
[TPS_SYST_02110]	Default behavior for ISignal.iSignalType
[TPS_SYST_02111]	VariableDataPrototype in case ISignal.iSignalType is set to array
[TPS_SYST_02112]	Usage of EventHandler.applicationEndpoint reference
[TPS_SYST_02113]	Usage of ConsumedEventGroup.applicationEndpoint reference
[TPS_SYST_02114]	Mapping of SwComponentPrototypes onto SwcToEcuMapping targets
[TPS_SYST_02115]	Applicability of GlobalTimeDomain.globalTimeDomainProps
[TPS_SYST_02116]	Modeling of Service Discovery Pdus
[TPS_SYST_02117]	Length of GeneralPurposePdu with category SD
[TPS_SYST_02118]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD
[TPS_SYST_02119]	SocketConnectionBundles for GeneralPurposePdus with category SD
[TPS_SYST_02120]	runtimeIpAddressConfiguration and runtimePortConfiguration settings for SD SocketConnections
[TPS_SYST_02121]	Scope of DataPrototypeTransformationProps
[TPS_SYST_02123]	Size of a length field for a chosen fixed-size array
[TPS_SYST_02124]	Size of a length field for a chosen structure
[TPS_SYST_02125]	Size of a length field for a chosen union
[TPS_SYST_02126]	Alignment of a dynamic DataPrototype

[TPS_SYST_02127]	Usage of DataPrototypeTransformationProps in case of a VariableDataPrototype
[TPS_SYST_02128]	Usage of DataPrototypeTransformationProps in case of a ClientServerOperation
[TPS_SYST_02129]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype typed by an ApplicationDataType
[TPS_SYST_02130]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ApplicationDataType
[TPS_SYST_02131]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype typed by an ImplementationDataType
[TPS_SYST_02132]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ImplementationDataType
[TPS_SYST_02133]	BufferProperties.bufferComputation setting for a COM Based transformer
[TPS_SYST_02134]	Recommended configuration settings for E2E Profile 7 configuration setting A
[TPS_SYST_02135]	Recommended configuration settings for E2E Profile 7 configuration setting B
[TPS_SYST_02136]	Serialization based on the network representation
[TPS_SYST_02137]	Serialization based on the ImplementationDataTypes
[TPS_SYST_02138]	Definition of the network representation
[TPS_SYST_02139]	Applicability of the SwDataDefProps attributes for the network representation of the serialized data
[TPS_SYST_02140]	SocketConnectionBundle.udpChecksumHandling default value
[TPS_SYST_02141]	Semantics of udpChecksumHandling
[TPS_SYST_02142]	Reception of invalid checksum
[TPS_SYST_02143]	Support of Multisource Pdu routing
[TPS_SYST_02144]	ComTimeoutSubstitution does not apply for signal gateway operation
[TPS_SYST_02145]	Default behavior for not defined nmPncParticipation
[TPS_SYST_02146]	Explicit definition of pncVector at NmPdu
[TPS_SYST_02147]	Implicit definition of pncVector at NmPdu
[TPS_SYST_02148]	Meaning of useAsCryptographicIPdu that is not set or set to false
[TPS_SYST_02149]	Meaning of useAsCryptographicIPdu that is set to true
[TPS_SYST_02150]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that LdCom is used
[TPS_SYST_02151]	MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs if one SystemSignal per communication direction is used
[TPS_SYST_02152]	Security profile
[TPS_SYST_02153]	Standardized values for the attribute category of meta-class SecureCommunicationFreshnessProps
[TPS_SYST_02154]	Standardized values for the attribute category of meta-class SecureCommunicationAuthenticationProps
[TPS_SYST_02155]	Recommended configuration settings for E2E Profile 11 configuration setting C
[TPS_SYST_02156]	Length of GeneralPurposeIPdu with category SOMEIP_SEGMENTED_IPDU
[TPS_SYST_02157]	Default value for the attribute category of meta-class EthernetCommunicationConnector
[TPS_SYST_02158]	Default value for the attribute category of meta-class EthernetCommunicationController
[TPS_SYST_02159]	Default value for the attribute category of meta-class EthernetPhysicalChannel

[TPS_SYST_02160]	EthernetPhysicalChannels with different category values are not allowed within an EthernetCluster
[TPS_SYST_02161]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that Com is used
[TPS_SYST_03016]	Applicability of EthGlobalTimeManagedCouplingPort.pdelayRequestPeriod
[TPS_SYST_03017]	Reference to CouplingPort in the context of a GlobalTimeDomain
[TPS_SYST_03018]	Aggregation of PNCs at the hostPort
[TPS_SYST_03019]	Modeling of CouplingPorts for managed CouplingElement
[TPS_SYST_03020]	Default value for CouplingPort.couplingPortRole if not defined
[TPS_SYST_03021]	Routing of GeneralPurposePdus with category GLOBAL_TIME

Table E.30: Added Traceables in R4.3.0

E.9.2 Changed Specification Items in R4.3.0

Id	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus
[TPS_SYST_01065]	Mapping onto the ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01118]	Support of Multicast Pdu routing
[TPS_SYST_01142]	Rules for the creation of references to Ports (ecuCommPortInstance) with communicationDirection out on sending Ecu
[TPS_SYST_02002]	SoAdRoutingGroup for Services with Methods
[TPS_SYST_02003]	SoAdRoutingGroups for Services with event groups
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02033]	Order of the transformerChain references in the configuration represents the order on the sending side
[TPS_SYST_02044]	Buffer computation of transformer
[TPS_SYST_02073]	EndToEndTransformationDescription.profileName
[TPS_SYST_02092]	Size of Fixed-size Array Length Fields
[TPS_SYST_02093]	Size of Structure Length Fields
[TPS_SYST_02094]	Size of Union Length Fields
[TPS_SYST_05009]	GlobalTimeDomain.globalTimePduTriggering for transmitting global time information
[TPS_SYST_05010]	GlobalTimeDomain.globalTimePduTriggering is not required on Ethernet

Table E.31: Changed Traceables in R4.3.0

E.9.3 Deleted Specification Items in R4.3.0

Id	Heading
[TPS_SYST_01027]	Mapping of specific SW components to dedicated Ecus
[TPS_SYST_01028]	Task of the System Generator
[TPS_SYST_01029]	Mapping of specific SW components to exclusive Ecus
[TPS_SYST_01141]	Derivation of ComSignalType

Table E.32: Deleted Traceables in R4.3.0

E.9.4 Added Constraints in R4.3.0

Id	Heading
[constr_1441]	In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer
[constr_1463]	Applicable values for J1939Cluster.networkId
[constr_3222]	No ByteOrderEnum.opaque allowed for PduToFrameMapping.packingByteOrder
[constr_3223]	No ByteOrderEnum.opaque allowed for MultiplexedIPdu.selectorFieldByteOrder
[constr_3224]	No ByteOrderEnum.opaque allowed for SegmentPosition.segmentByteOrder.
[constr_3225]	LinFrameTriggering.linChecksum not allowed for LinSporadicFrames
[constr_3226]	LinFrameTriggering.linChecksum for LinEventTriggeredFrames
[constr_3227]	NmNode.nmPassiveModeEnabled setting
[constr_3229]	SwComponentPrototype mapped to an ApplicationPartition and EcuInstance
[constr_3230]	Usage of SenderRecRecordElementMapping.applicationRecordElement
[constr_3231]	Usage of IndexedArrayElement.applicationArrayElement
[constr_3232]	ApplicationPartition is allowed to be mapped to only one EcuPartition
[constr_3239]	Consistent mapping of software-component to J1939NmNode
[constr_3240]	Consistent mapping of J1939ControllerApplication to EcuInstance
[constr_3241]	Usage of AssignFrameId.messageId
[constr_3242]	Usage of UnassignFrameId.messageId
[constr_3243]	FrameTriggering.pduTriggering condition
[constr_3244]	Usage of SenderRecRecordElementMapping.implementationRecordElement
[constr_3245]	Usage of IndexedArrayElement.implementationArrayElement
[constr_3246]	Frame.packingByteOrder mix within a Frame is not allowed
[constr_3247]	Byte order mix within a MultiplexedIPdu is not allowed
[constr_3248]	Category of HwElement for ECUMapping
[constr_3249]	Category of HwElement for SwcToEcuMapping
[constr_3250]	PduTriggering.iSignalTriggering condition
[constr_3251]	Value of GlobalTimeDomain.domainId in subDomain chains
[constr_3252]	ISignalTriggering.iSignalPort reference condition
[constr_3253]	PduTriggering.iPduPort reference condition
[constr_3254]	FrameTriggering.framePort reference condition
[constr_3255]	FrameTriggering.pduTriggering reference condition with regard to the PhysicalChannel
[constr_3256]	PduTriggering.iSignalTriggering reference condition with regard to the PhysicalChannel
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3258]	Restriction on ISignal.length in case iSignalType is set to array
[constr_3259]	Allowed use of SdServerConfig.capabilityRecord
[constr_3260]	Allowed use of SdClientConfig.capabilityRecord
[constr_3261]	GlobalTimeDomain.globalTimePduTriggering category
[constr_3262]	ConsumedEventGroup.eventGroupIdentifier is mandatory
[constr_3263]	Restriction of usage of SwcToEcuMapping in a System

[constr_3264]	Server side ClientServerToSignalMappings in case of a n:1 inter-ECU client-server communication
[constr_3265]	TransformationTechnology.hasInternalState setting for an E2E transformer
[constr_3266]	TransformationTechnology.hasInternalState setting for a SOME/IP Transformer
[constr_3267]	PduTriggerings in Service Discovery SocketConnectionBundles
[constr_3268]	Service Discovery SocketConnectionBundle serverPort reference to a TpPort
[constr_3269]	Service Discovery SocketConnection clientPort reference to a TpPort
[constr_3270]	Service Discovery SocketConnection clientPort reference to an IP Address
[constr_3271]	clientIpAddrFromConnectionRequest and clientPortFromConnectionRequest settings for SD SocketConnections
[constr_3272]	SocketConnectionIpduIdentifier.headerId setting for SD SocketConnectionBundles
[constr_3273]	Service Discovery multicast SocketConnectionBundle's serverPort reference to an IP Address
[constr_3274]	Service Discovery unicast SocketConnectionBundle's serverPort reference to an IP Address
[constr_3275]	PduTriggering containment in different PdurIPduGroups of the same EcuInstance is not allowed
[constr_3276]	Prohibition of usage of allowedIPv6ExtHeaders in IPv4 SocketConnections
[constr_3277]	Restriction of usage of IPv6ExtHeaderFilterLists in IPv6 SocketConnections
[constr_3278]	Usage of SOMEIPTransformationProps.sizeOfArrayLengthField
[constr_3279]	Usage of SOMEIPTransformationProps.sizeOfStructLengthField
[constr_3280]	Usage of SOMEIPTransformationProps.sizeOfUnionLengthField
[constr_3281]	Usage of SOMEIPTransformationProps.alignment
[constr_3282]	SOME/IP Transformation settings for static size arrays in the context of an ISignal
[constr_3283]	SOME/IP Transformation settings for structures in the context of an ISignal
[constr_3284]	SOME/IP Transformation settings for unions in the context of an ISignal
[constr_3285]	Alignment of variable data length data elements in the context of an ISignal
[constr_3286]	ISignal.length shall be consistent to transformer configuration
[constr_3297]	Prohibition of usage of allowedTcpOptions in Udp SocketConnections
[constr_3298]	Ipv6Configuration.ipv6Address range in case of enableAnycast
[constr_3299]	SocketConnectionBundle.pathMtuDiscoveryEnabled setting dependency
[constr_3311]	Usage of SocketConnectionBundle.flowLabel
[constr_3312]	Consistency of vlanPriority and EthernetCommunicationConnector
[constr_3313]	E2E transformer configuration
[constr_3314]	BufferProperties.bufferComputation is mandatory
[constr_3315]	The value of V0 in BufferProperties.bufferComputation setting for a COM Based transformer
[constr_3316]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07
[constr_3317]	Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the ImplementationDataTypes
[constr_3318]	Allowed use of ISignal.networkRepresentationProps
[constr_3319]	Existence of DataPrototypeTransformationProps.networkRepresentationProps
[constr_3322]	Consistent setting of SocketConnectionIpduIdentifier.pduCollectionSemantics in the context of one SocketConnectionBundle
[constr_3323]	Relation between NmCluster.nmPncParticipation and PncMapping.pncGroup
[constr_3324]	Category of SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps

[constr_3325]	SecureCommunicationFreshnessProps and SecureCommunicationAuthenticationProps attribute values for predefined categories
[constr_3326]	Allowed values for EndToEndTransformationISignalProps.dataIdMode in PROFILE_11
[constr_3327]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_22
[constr_3328]	SomeipTpConnection.transportPdu reference restriction
[constr_3329]	SomeipTpConnection.tpSdu reference restriction
[constr_3330]	Same transportPdu shall not be used in different SomeipTpConnections
[constr_3331]	Standardized values for the attribute category of meta-class EthernetCommunicationConnector
[constr_3332]	Standardized values for the attribute category of meta-class EthernetCommunicationController
[constr_3333]	Standardized values for the attribute category of meta-class EthernetPhysicalChannel
[constr_3334]	Allowed references between EthernetPhysicalChannel and EthernetCommunicationConnector
[constr_3335]	Allowed references between EthernetCommunicationConnector and EthernetCommunicationController
[constr_3336]	EthernetPhysicalChannel.soAdConfig in case of WIRELESS EthernetPhysicalChannel
[constr_3337]	IPduPort.useAuthDataFreshness is configurable on the receiver side
[constr_3338]	IPduPort.useAuthDataFreshness validness
[constr_3339]	Relation between authDataFreshnessStartPosition, authDataFreshnessLength and useAuthDataFreshness
[constr_3521]	defaultVlan and vlanMembership
[constr_3522]	vlanModifier and vlanMembership
[constr_3523]	CouplingPort and PncMapping in the scope of an EthernetPhysicalChannel
[constr_3524]	Definition of couplingPortRole on CouplingPort for managed CouplingElement
[constr_3525]	Connection of CouplingPort with couplingPortRole set to upLinkPort

Table E.33: Added Constraints in R4.3.0

E.9.5 Changed Constraints in R4.3.0

Id	Heading
[constr_1372]	Consistency of attribute globalTimePduTriggering
[constr_3042]	pncVectorLength range
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs in case of networks other than Ethernet
[constr_3069]	Allowed CanNmCluster.nmNidPosition values
[constr_3070]	Allowed CanNmCluster.nmCbvPosition values
[constr_3078]	Allowed UdpNmCluster.nmNidPosition values
[constr_3079]	Allowed UdpNmCluster.nmCbvPosition values
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3113]	AbstractEthernetFrame shall not have a PduToFrameMapping
[constr_3121]	The length of transformer chains is limited to 255 transformers
[constr_3128]	SOME/IP transformer configuration
[constr_3136]	Allowed payload of SecuredIPdus

[constr_3149]	TransformationTechnology.needsOriginalData settings for E2E Transformer
[constr_3150]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01 and PROFILE_11 in case it is 0
[constr_3151]	BufferProperties.headerLength settings for an E2E transformer used in combination with a SOME/IP transformer
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3154]	BufferProperties.bufferComputation setting for an E2E transformer when used together with a Com-based transformer
[constr_3156]	Allowed values for EndToEndTransformationISignalProps.dataId in PROFILE_01 and PROFILE_11
[constr_3157]	Allowed values for EndToEndTransformationISignalProps.dataId in PROFILE_01 and PROFILE_11 in case dataIdMode is set to lower12Bit
[constr_3158]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_01 and PROFILE_11
[constr_3160]	EndToEndTransformationISignalProps.dataId in PROFILE_02 and PROFILE_22
[constr_3161]	EndToEndTransformationISignalProps.dataLength in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22
[constr_3162]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22
[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04, PROFILE_06, PROFILE_07
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07
[constr_3165]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01, PROFILE_11
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[constr_3169]	Attribute multiplicities and values in PROFILE_02 and PROFILE_22
[constr_3171]	Value of EndToEndTransformationISignalProps.dataId shall be unique in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22
[constr_3184]	Only one EndToEndTransformationISignalProps.dataId element in PROFILE_01 and PROFILE_11
[constr_3185]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_01 and PROFILE_11
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3187]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_01 and PROFILE_11
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3189]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_01 and PROFILE_11
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3191]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_01, PROFILE_11 and dataIdMode equal to lower12Bit

[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22 or dataIdMode different from lower12Bit
[constr_3193]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_01 and PROFILE_11
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_22
[constr_3195]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_02 and PROFILE_22
[constr_3514]	No two ISignalToIPduMappings shall reference the identical ISignal

Table E.34: Changed Constraints in R4.3.0

E.9.6 Deleted Constraints in R4.3.0

Id	Heading
[constr_3032]	Combinations of SwcToEcuMapping targets
[constr_3061]	CompuMethod specification in networkRepresentationProps
[constr_3207]	Assignment of SocketConnectionIpduIdentifiers used for ClientServer Communication to SocketConnections

Table E.35: Deleted Constraints in R4.3.0

E.10 Constraint and Specification Item History of this document according to AUTOSAR R4.3.1

E.10.1 Added Specification Items in R4.3.1

Number	Heading
[TPS_SYST_02162]	Routing of ISignals of ISignalGroups
[TPS_SYST_02163]	Applicability of syncLossTimeout
[TPS_SYST_02164]	LdCom: No ISignalPort.firstTimeout reception timeout defined
[TPS_SYST_02165]	Derivation of CanNmPnFilterMaskByte
[TPS_SYST_02166]	Derivation of UdpNmPnFilterMaskByte
[TPS_SYST_02167]	Derivation of FrNmPnFilterMaskByte
[TPS_SYST_02168]	MetaData support required if CanFrameTriggering.txMask is used
[TPS_SYST_02169]	MetaData support may be required if CanFrameTriggering.rxMask is used
[TPS_SYST_02170]	category of the GeneralPurposeConnection
[TPS_SYST_02171]	Secured Area in payload Pdu
[TPS_SYST_02172]	Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to false

▽

△

Number	Heading
[TPS_SYST_02173]	Modeling of SecuredIPdu in case useAsCryptographicIPdu is set to true
[TPS_SYST_02174]	Initial Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02175]	Repetition Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02176]	Main Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02177]	TTL for Offer Service Entries
[TPS_SYST_02178]	Servers RequestResponseDelay for received FindService entries
[TPS_SYST_02179]	Server Capability Records
[TPS_SYST_02180]	Usage of EventHandler.multicastThreshold
[TPS_SYST_02181]	TTL for SubscribeEventGroupAck Entries
[TPS_SYST_02182]	Servers RequestResponseDelay for received SubscribeEventGroup entries
[TPS_SYST_02183]	Initial Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02184]	Repetition Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02185]	TTL for Find Service Entries
[TPS_SYST_02186]	Client Capability Records
[TPS_SYST_02187]	SdClientConfig.ttl for SubscribeEventGroup Entries
[TPS_SYST_02188]	Clients RequestResponseDelay for received ServiceOffer entries
[TPS_SYST_02189]	Setting of useSecuredPduHeader attribute

Table E.36: Added Specification Items in R4.3.1

E.10.2 Changed Specification Items in R4.3.1

Number	Heading
[TPS_SYST_01120]	Precedence of ISignalMappings
[TPS_SYST_02098]	Header id and header type of a contained IPdu
[TPS_SYST_02100]	Relation between ContainerIPdu and contained IPdus on receiver side
[TPS_SYST_02112]	Usage of EventHandler.applicationEndpoint reference

Table E.37: Changed Specification Items in R4.3.1

E.10.3 Deleted Specification Items in R4.3.1

Number	Heading
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus

▽

△

Number	Heading
[TPS_SYST_02160]	EthernetPhysicalChannels with different category values are not allowed within an EthernetCluster

Table E.38: Deleted Specification Items in R4.3.1

E.10.4 Added Constraints in R4.3.1

Number	Heading
[constr_3364]	headerLength shall be a multiple of 8
[constr_3365]	EthernetPhysicalChannels with different category values are not allowed within an EthernetCluster
[constr_3373]	Limitation on the number of PhysicalChannels that are referencing a CommunicationConnector
[constr_3378]	Maximal one AliasNameAssignment allowed per FlatInstanceDescriptor
[constr_3379]	Multiple SocketAddress entries with the same IP Address, Protocol and Port in the context of a given EcuInstance
[constr_3383]	Standardized values for the attribute category of meta-class GeneralPurpose-Connection
[constr_3384]	PduTriggerings referenced by GeneralPurposeConnection shall be defined on the same PhysicalChannel
[constr_3385]	XcpChannel is allowed to reference exactly two PduTriggerings
[constr_3386]	XcpChannel is only allowed to reference PduTriggerings of GeneralPurpose-IPdus with category XCP
[constr_3399]	Existence of securedAreaOffset and securedAreaLength
[constr_3400]	Usage of SdClientConfig attributes in ConsumedServiceInstance and ConsumedEventGroup
[constr_3401]	Usage of SdServerConfig attributes in ProvidedServiceInstance and Event-Handler
[constr_3402]	Mandatory offset if noHeader is used
[constr_3403]	Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
[constr_3404]	Usage of ContainedIPduProps.updateIndicationBitPosition
[constr_3405]	Dynamic Length IPdu inside of a static configured ContainerIPdu
[constr_3406]	All signals before authDataFreshnessStartPosition shall have a static length
[constr_3407]	Freshness Value in Authentic IPdu is not allowed to be used in case of Container-IPdu with a dynamic layout

Table E.39: Added Constraints in R4.3.1

E.10.5 Changed Constraints in R4.3.1

Number	Heading
[constr_2025]	Uniqueness of `symbol` attributes
[constr_3052]	Complete `ISignalMapping` of `ISignalGroup` signals
[constr_3053]	Complete `ISignalMapping` of target `ISignalGroup`
[constr_3136]	Allowed payload of `SecuredIPdu`s

Table E.40: Changed Constraints in R4.3.1

E.10.6 Deleted Constraints in R4.3.1

Number	Heading
[constr_3139]	Usage of `IPduPort.rxSecurityVerification`

Table E.41: Deleted Constraints in R4.3.1

E.11 Constraint and Specification Item History of this document according to AUTOSAR R4.4.0

E.11.1 Added Specification Items in R4.4.0

Number	Heading
[TPS_SYST_02190]	`J1939TpConnection.transmitter` reference in case of broadcast connection
[TPS_SYST_02191]	`J1939TpConnection.transmitter` reference in case that the source is an unknown node
[TPS_SYST_02192]	`J1939TpConnection.receiver` reference in case that the destination is an unknown node
[TPS_SYST_02193]	`J1939TpConnection.receiver` reference in case that the destination is connected to a configured `J1939NmNode`
[TPS_SYST_02194]	Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07
[TPS_SYST_02195]	Applicable use cases for DataPrototypeInSystemRef
[TPS_SYST_02196]	`PduTriggering` is referenced by several `ContainerIPdu`s
[TPS_SYST_02197]	DRAFT
[TPS_SYST_02198]	Applicable `transferProperty` for `ISignal`
[TPS_SYST_02199]	Applicable `transferProperty` for `ISignalGroup` and no group signal has `transferProperty` defined
[TPS_SYST_02200]	Applicable `transferProperty` for `ISignalGroup` and group signals have `transferProperty` defined

▽

△

Number	Heading
[TPS_SYST_02201]	Existence of CanFrameTriggering.identifier
[TPS_SYST_02202]	Modeling of bus mirroring
[TPS_SYST_02203]	BusMirroring to CAN destination channel
[TPS_SYST_02204]	BusMirroring to FlexRay destination channel
[TPS_SYST_02205]	BusMirroring to Ethernet destination channel
[TPS_SYST_02206]	BusMirroring to UserDefined destination channel
[TPS_SYST_05016]	Assignment of TLV data ids
[TPS_SYST_05017]	Definition of the applicable wire type attribute SOMEIPTransformationISignalProps.isDynamicLengthFieldSize shall be used to define the applicable wire type
[TPS_SYST_05018]	Semantics of meta-class LinSlave
[TPS_SYST_05019]	Semantics of LinErrorResponse.responseError
[TPS_SYST_05020]	Semantics of CryptoServiceMapping
[TPS_SYST_05021]	Semantics of SecOcCryptoServiceMapping
[TPS_SYST_05022]	Semantics of PduTriggering.secOcCryptoMapping
[TPS_SYST_05023]	Semantics of CryptoServicePrimitive
[TPS_SYST_05024]	Semantics of CryptoServiceKey
[TPS_SYST_05025]	Standardized values of CryptoServicePrimitive.algorithmFamily and CryptoServiceKey.algorithmFamily
[TPS_SYST_05026]	Relation of CryptoServicePrimitive.algorithmFamily to CryptoServiceKey.algorithmFamily
[TPS_SYST_05027]	Standardized values of CryptoServicePrimitive.algorithmMode
[TPS_SYST_05028]	Semantics of CryptoServiceKey.keyStorageType
[TPS_SYST_05029]	Semantics of meta-class TlsCryptoServiceMapping
[TPS_SYST_05030]	Semantics of TlsCryptoCipherSuite
[TPS_SYST_05031]	Existence of TlsCryptoCipherSuite.keyExchange vs. TlsCryptoServiceMapping.keyExchange
[TPS_SYST_05032]	Semantics of CryptoServiceCertificate
[TPS_SYST_05033]	Existence of TlsCryptoCipherSuite.certificate in the *client* role

Table E.42: Added Specification Items in R4.4.0

E.11.2 Changed Specification Items in R4.4.0

Number	Heading
[TPS_SYST_01100]	TP routing without using transport protocol modules (low-level routing)
[TPS_SYST_01101]	TP routing using transport protocol modules
[TPS_SYST_01114]	Frame fan-out support
[TPS_SYST_02064]	Reception acceptance of contained IPdus

▽

△

Number	Heading
[TPS_SYST_02068]	Transformer header field representation in an ISignalGroup
[TPS_SYST_02099]	Relation between ContainerIPdu and contained PduTriggerings on sender side
[TPS_SYST_02113]	Usage of ConsumedEventGroup.applicationEndpoint reference
[TPS_SYST_02131]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype typed by an ImplementationDataType

Table E.43: Changed Specification Items in R4.4.0

E.11.3 Deleted Specification Items in R4.4.0

Number	Heading
[TPS_SYST_01046]	ShortNames of LinSlaveConfig and LinSlave
[TPS_SYST_01049]	Handling of IPdus with dynamic signals
[TPS_SYST_02112]	Usage of EventHandler.applicationEndpoint reference

Table E.44: Deleted Specification Items in R4.4.0

E.11.4 Added Constraints in R4.4.0

Number	Heading
[constr_1641]	Consistent assignment of TLV data ids to ApplicationRecordDataType
[constr_1642]	Consistent assignment of TLV data ids to ImplementationDataType or ImplementationDataTypeElement
[constr_1643]	Completeness of the existence of a set of TlvDataIdDefinition.tlvArguments
[constr_1644]	Completeness of the existence of a set of TlvDataIdDefinition.tlvRecordElements
[constr_1645]	Completeness of the existence of a set of TlvDataIdDefinition.tlvImplementationDataTypeElements
[constr_1646]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ArgumentDataPrototype
[constr_1647]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ApplicationRecordElement
[constr_1648]	Scope of the uniqueness of the value of TlvDataIdDefinition.id for references to ImplementationDataTypeElement
[constr_1649]	TlvDataIdDefinition referencing ArgumentDataPrototype
[constr_1650]	TlvDataIdDefinition referencing ApplicationRecordElement
[constr_1651]	TlvDataIdDefinition referencing ImplementationDataTypeElement
[constr_1652]	Definition of static length fields sizes in case of TLV usage

▽

△

Number	Heading
[constr_1653]	Identical values for length fields sizes in case of TLV usage
[constr_1654]	No definition of length field sizes on DataPrototype level in case of TLV usage
[constr_1655]	The mutual existence of LinMasters in the LinSlave EcuExtract
[constr_1656]	No application-level write access to LinErrorResponse.responseError on Lin slave
[constr_1657]	Existence of LinPhysicalChannel.scheduleTable
[constr_1669]	Existence of PduTriggering.secOcCryptoMapping
[constr_1670]	Prohibition of usage of tlsCryptoMapping in case of UDP socket connections
[constr_1671]	Supported values of TlsCryptoServiceMapping.category
[constr_1672]	Existence of TlsCryptoCipherSuite.certificate in the *server* role
[constr_3435]	Applicability of CouplingPort.macMulticastAddress
[constr_3436]	Value range of minimumTxContainerQueueSize and minimumRxContainer-QueueSize
[constr_3437]	invalidValue defined in the context of ISignal
[constr_3438]	timeoutSubstitutionValue defined in the context of ISignal
[constr_3448]	Restriction for usage of Pdu.hasDynamicLength
[constr_3453]	Uniqueness of header Id in case of acceptAll ContainerIPdu
[constr_3454]	Unique headerIdLongHeader for acceptConfigured
[constr_3455]	Unique headerIdShortHeader for acceptConfigured
[constr_3456]	Existence of ProvidedServiceInstance.loadBalancingPriority and ProvidedServiceInstance.loadBalancingWeight
[constr_3457]	Uniqueness of ConsumedEventGroup.eventGroupIdentifier in the scope of a ConsumedServiceInstance
[constr_3458]	DRAFT
[constr_3459]	Applicable transferProperty for group signal
[constr_3460]	Full definition of transferProperty for group signal
[constr_3461]	TransferProperty for group signals if ISignalGroup has transfer-Property=pending
[constr_3464]	Allowed Pdu type on BusMirrorChannelMapping.targetChannel
[constr_3465]	Identical BusMirrorChannel.busMirrorNetworkId for BusMirrorChannels referencing the same PhysicalChannel
[constr_3466]	Unique BusMirrorChannel.busMirrorNetworkIds for each specialization of PhysicalChannel
[constr_3467]	CanPhysicalChannel as destination channel of BusMirrorChannelMapping-Can
[constr_3468]	BusMirrorChannelMappingCan.targetPduTriggering restriction
[constr_3469]	CanFrameTriggering.txMask setting for the destination frame
[constr_3470]	PaddingValue used to transmit the Pdu on a Can-Fd destination bus
[constr_3471]	FlexrayPhysicalChannel as destination channel of BusMirrorChannel-MappingFlexray
[constr_3472]	Number of BusMirrorChannels derived for one FlexrayCluster

▽

△

Number	Heading
[constr_3473]	BusMirrorChannelMappingFlexray.targetPduTriggering restriction
[constr_3474]	EthernetPhysicalChannel as destination channel of BusMirrorChannelMappingIp
[constr_3475]	BusMirrorChannelMappingIp.targetPduTriggering restriction
[constr_3476]	UserDefinedPhysicalChannel as destination channel of BusMirrorChannelMappingUserDefined
[constr_3477]	BusMirrorChannelMappingUserDefined.targetPduTriggering restriction
[constr_3479]	PhysicalChannel is not allowed to be a managedPhysicalChannel and a managing PhysicalChannel
[constr_3480]	PhysicalChannel shall be referenced in the role managedPhysicalChannel only once
[constr_3481]	UdpNmCluster is not allowed to reference a managedPhysicalChannel in the role vlan
[constr_3482]	NmCluster is not allowed to reference a CommunicationCluster that aggregates a managedPhysicalChannel
[constr_3483]	The same PhysicalChannel is not allowed to be the source and the target of managedPhysicalChannel references
[constr_3484]	PncMapping that refers a managedPhysicalChannel shall also refer the managing PhysicalChannel
[constr_3488]	Value range of ContainedIPduProps.priority
[constr_3489]	ContainedIPduProps.priority is only applicable if a ContainerIPdu header is used
[constr_3490]	ContainedIPduProps.priority is only applicable if collectionSemantics is set to lastIsBest
[constr_3533]	EndToEndTransformationISignalProps.dataLength shall be a multiple of 8
[constr_3534]	EthernetPhysicalChannel shall only be referenced by one VlanMembership
[constr_3535]	EthernetCommunicationController shall aggregate at most one CouplingPort

Table E.45: Added Constraints in R4.4.0

E.11.5 Changed Constraints in R4.4.0

Number	Heading
[constr_3067]	initValue defined in the context of ISignal
[constr_3165]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01, PROFILE_11
[constr_3219]	The mutual existence of LinSlaves in the LinMaster EcuExtract
[constr_3323]	Relation between NmCluster.nmPncParticipation and PncMapping.pncGroup

▽

△

Number	Heading
[constr_3327]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_22

Table E.46: Changed Constraints in R4.4.0

E.11.6 Deleted Constraints in R4.4.0

Number	Heading
[constr_3024]	Usage of triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition is not allowed for signal groups and group signals.
[constr_3034]	Values of LinSlaveConfig and LinSlave attributes
[constr_3043]	pncVector configuration in AUTOSAR Com
[constr_3150]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_01 and PROFILE_11 in case it is 0
[constr_3171]	Value of EndToEndTransformationISignalProps.dataId shall be unique in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07

Table E.47: Deleted Constraints in R4.4.0

E.12 Constraint and Specification Item History of this document according to AUTOSAR R19-11

E.12.1 Added Specification Items in R19-11

Number	Heading
[TPS_SYST_02207]	Routing on the fly
[TPS_SYST_02208]	ISignalPort.handleInvalid defines the reception invalidation behavior
[TPS_SYST_02209]	Not defined ISignalPort.handleInvalid behavior
[TPS_SYST_02210]	Data invalidation in case the dataTypePolicy is set to override or legacy
[TPS_SYST_02211]	Reference from SOMEIPTransformationISignalProps to TlvDataIdDefinitionSet
[TPS_SYST_02212]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a ClientServerInterface typed by an ApplicationDataType
[TPS_SYST_02213]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a ClientServerInterface typed by an ApplicationDataType

▽

△

Number	Heading
[TPS_SYST_02214]	Assignment of SOMEIPTransformationProps to a root Autosar-DataPrototype in a ClientServerInterface typed by an ImplementationDataType
[TPS_SYST_02215]	Usage of portNumber with value 0
[TPS_SYST_02216]	Configuration of capabilityRecords
[TPS_SYST_02217]	SOME/IP Service offer
[TPS_SYST_02218]	ProvidedServiceInstance deployment
[TPS_SYST_02219]	Static configuration between ProvidedServiceInstance and ConsumedServiceInstance
[TPS_SYST_02220]	Maximal number of clients that may connect to the local server address
[TPS_SYST_02221]	ProvidedServiceInstance.localUnicastAddress reference target
[TPS_SYST_02222]	Usage of headerId
[TPS_SYST_02223]	Activation/Deactivation of PduActivationRoutingGroups
[TPS_SYST_02224]	Methods provided by a ProvidedServiceInstance
[TPS_SYST_02225]	Service methods provided over UDP
[TPS_SYST_02226]	Service methods provided over TCP
[TPS_SYST_02227]	Publishing of a SOME/IP Event group
[TPS_SYST_02228]	Transmission of events over UDP/TCP Port
[TPS_SYST_02229]	Event groups provided by a ProvidedServiceInstance
[TPS_SYST_02230]	PduActivationRoutingGroups for event groups
[TPS_SYST_02231]	PduActivationRoutingGroups for methods
[TPS_SYST_02232]	Events provided over UDP
[TPS_SYST_02233]	Events provided over TCP
[TPS_SYST_02234]	SOME/IP Service search
[TPS_SYST_02235]	ConsumedServiceInstance deployment
[TPS_SYST_02236]	Static configuration between ConsumedServiceInstance and ProvidedServiceInstance
[TPS_SYST_02237]	Maximal number of servers that may connect to the local client address
[TPS_SYST_02238]	ConsumedServiceInstance.localUnicastAddress reference target
[TPS_SYST_02239]	Methods consumed by a ConsumedServiceInstance
[TPS_SYST_02240]	Service methods consumed over UDP
[TPS_SYST_02241]	Service methods consumed over TCP
[TPS_SYST_02242]	Subscription to a SOME/IP Event group
[TPS_SYST_02243]	Reception of events over UDP/TCP Port in case of Service Discovery
[TPS_SYST_02244]	ConsumedServiceInstance without a defined localUnicastAddress
[TPS_SYST_02245]	Event groups consumed by a ConsumedServiceInstance
[TPS_SYST_02246]	PduActivationRoutingGroups for ConsumedEventGroups
[TPS_SYST_02247]	Events consumed over UDP
[TPS_SYST_02248]	Events consumed over TCP

▽

△

Number	Heading
[TPS_SYST_02249]	Service Discovery Message sending behavior on ProvidedServiceInstance
[TPS_SYST_02250]	Service Discovery Message sending behavior on ConsumedServiceInstance
[TPS_SYST_02251]	Non-SOME/IP data exchange between two communication endpoints
[TPS_SYST_02252]	Description of a TCP Client
[TPS_SYST_02253]	Description of a TCP Server
[TPS_SYST_02254]	Pdus transported over the StaticSocketConnection
[TPS_SYST_02255]	Frame.frameLength usage for FlexrayFrames and CanFrames
[TPS_SYST_02256]	Allowed CanFrame.frameLength settings
[TPS_SYST_02257]	Standardized values of LinCommunicationController.protocolVersion and LinSlaveConfig.protocolVersion
[TPS_SYST_02258]	Shared random timer for ProvidedServiceInstance service discovery
[TPS_SYST_02259]	Shared random timer for ProvidedServiceInstance service discovery
[TPS_SYST_02260]	Individual random timer for ProvidedServiceInstance service discovery
[TPS_SYST_02261]	Shared random timer for ConsumedServiceInstance service discovery
[TPS_SYST_02262]	Shared random timer for ConsumedServiceInstance service discovery
[TPS_SYST_02263]	Individual random timer for ConsumedServiceInstance service discovery
[TPS_SYST_02264]	Usage of DltLogChannel
[TPS_SYST_02265]	Configuration of IPsec
[TPS_SYST_02266]	Definition of IPSecRules
[TPS_SYST_02267]	IPsec policy
[TPS_SYST_02268]	IPsec mode
[TPS_SYST_02269]	IPsec AH and ESP protocol configuration
[TPS_SYST_02270]	Definition of general IPsec configuration settings
[TPS_SYST_02271]	IPsec AH and ESP CipherSuites
[TPS_SYST_02272]	IPsec Internet Key Exchange protocol configuration
[TPS_SYST_02273]	Protection of ProvidedServiceInstance by IPsec
[TPS_SYST_02274]	Protection of ConsumedServiceInstance by IPsec
[TPS_SYST_03022]	Provided service instance with translationStart requires *SdServerServiceAutoAvailable*
[TPS_SYST_03023]	Required service istance with translationStart requires *SdClientServiceAutoRequire*
[TPS_SYST_03024]	Required service instance with translationStart requires *SdConsumedEventGroupAutoRequire*
[TPS_SYST_03025]	Control of service instance in case of partialNetwork
[TPS_SYST_03026]	Monitoring of the *partial networks* status in case of partialNetwork for provided service instance
[TPS_SYST_03027]	Monitoring of the *partial networks* status in case of partialNetwork for required service instance
[TPS_SYST_03028]	Initial *service find* for a provided service instance with serviceControl

▽

△

Number	Heading
[TPS_SYST_03029]	Inital *offer* for a provided service instance with `serviceControl`
[TPS_SYST_03030]	*Stop offer* for a provided service instance with `serviceControl`
[TPS_SYST_03031]	Sufficient ASIL level of translation software component
[TPS_SYST_03032]	Data transmission in case of *E_OK* safe signal reception
[TPS_SYST_03033]	No data transmission in case of reception timeout
[TPS_SYST_03034]	Handling safe signal reception
[TPS_SYST_03036]	`PortAPIOption` for `safeTranslation RPortPrototype`
[TPS_SYST_03037]	`PortAPIOption` for `safeTranslation PPortPrototype`
[TPS_SYST_03038]	Definition of transmission triggers for translations with different sources
[TPS_SYST_03039]	Full translation before transmission triggering
[TPS_SYST_03040]	Transmission trigger for translations with different sources
[TPS_SYST_03041]	No transmission trigger for translations with different sources
[TPS_SYST_03042]	Periodic call in case of `ReceiverComSpec`.dataUpdatePeriod
[TPS_SYST_03043]	Periodic call in case of `SenderComSpec`.dataUpdatePeriod
[TPS_SYST_03044]	Handling of safe signal-service-translation in one software component
[TPS_SYST_03045]	Support for safe signal-service-translation and service-signal-translation
[TPS_SYST_03046]	Support for safe signal-service-translation with same or different E2E profiles
[TPS_SYST_03047]	1:n mapping for E2E protected data
[TPS_SYST_03048]	E2E protected target out of E2E protected sources
[TPS_SYST_03049]	No translation of not OK E2E protected composed data
[TPS_SYST_03050]	Usage of `ConsumedServiceInstance`.`blacklistedVersion`
[TPS_SYST_03051]	Data filter inside the signal-service-translation software component
[TPS_SYST_05034]	`DataMapping` of `ImplementationDataType` of category UNION, DATA_REFERENCE, or FUNCTION_REFERENCE

Table E.48: Added Specification Items in R19-11

E.12.2 Changed Specification Items in R19-11

Number	Heading
[TPS_SYST_01090]	valid `NetworkEndpoint`
[TPS_SYST_01143]	`DataMapping` on the sender side for elements of a composite data type
[TPS_SYST_02019]	LdCom: `ISignalToIPduMapping`.`transferProperty` shall be triggered or triggeredWithoutRepetition for sent `ISignal`s
[TPS_SYST_02020]	LdCom: No `IPduTiming`.`minimumDelay` defined
[TPS_SYST_02022]	LdCom: Only the `transmissionModeTrueTiming` defined
[TPS_SYST_02023]	LdCom: `DataFilter` "always" if `TransmissionModeCondition` defined
[TPS_SYST_02024]	LdCom: No `ModeDrivenTransmissionModeCondition` defined

▽

△

Number	Heading
[TPS_SYST_02119]	StaticSocketConnections for GeneralPurposePdus with category SD
[TPS_SYST_02129]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType
[TPS_SYST_02130]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype in a SenderReceiverInterface typed by an ApplicationDataType
[TPS_SYST_02131]	Assignment of SOMEIPTransformationProps to a root AutosarDataPrototype in a SenderReceiverInterface typed by an ImplementationDataType
[TPS_SYST_02132]	Assignment of SOMEIPTransformationProps to a subElement of a root AutosarDataPrototype typed by an ImplementationDataType
[TPS_SYST_02133]	BufferProperties.bufferComputation setting for a COM Based transformer
[TPS_SYST_02140]	SocketAddress.udpChecksumHandling default value
[TPS_SYST_02141]	Semantics of udpChecksumHandling
[TPS_SYST_02142]	Reception of invalid checksum
[TPS_SYST_02150]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs
[TPS_SYST_02151]	MetaData support required for inter-ECU client server communication over Ethernet with clients located on different ECUs
[TPS_SYST_02174]	Initial Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02175]	Repetition Wait Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02176]	Main Phase configuration for a ProvidedServiceInstance
[TPS_SYST_02177]	TTL for Offer Service Entries
[TPS_SYST_02178]	Servers RequestResponseDelay for received FindService entries
[TPS_SYST_02182]	Servers RequestResponseDelay for received SubscribeEventGroup entries
[TPS_SYST_02183]	Initial Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02184]	Repetition Wait Phase configuration for a ConsumedServiceInstance
[TPS_SYST_02185]	TTL for Find Service Entries
[TPS_SYST_02187]	SomeipSdClientEventGroupTimingConfig.timeToLive for SubscribeEventGroup Entries
[TPS_SYST_02188]	Clients RequestResponseDelay for received ServiceOffer entries
[TPS_SYST_02189]	Setting of useSecuredPduHeader attribute
[TPS_SYST_02195]	Applicable use cases for DataPrototypeReference
[TPS_SYST_05016]	Assignment of TLV data ids
[TPS_SYST_05033]	Existence of TlsCryptoCipherSuite.certificate and TlsCryptoCipherSuite.pskIdentity in the *client* role

Table E.49: Changed Specification Items in R19-11

E.12.3 Deleted Specification Items in R19-11

Number	Heading
[TPS_SYST_01092]	Transmission of multiple Pdus over the same SocketConnection
[TPS_SYST_01093]	Activation/Deactivation of SoAdRoutingGroups
[TPS_SYST_01151]	DataMapping reference to an EventHandler
[TPS_SYST_01152]	DataMapping reference to a ConsumedEventGroup
[TPS_SYST_02002]	SoAdRoutingGroup for Services with Methods
[TPS_SYST_02003]	SoAdRoutingGroups for Services with event groups
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02007]	Usage of SocketConnection attributes in the unicast server view
[TPS_SYST_02008]	Usage of SocketConnection attributes in the unicast client view
[TPS_SYST_02009]	Usage of SocketConnection attributes in the multicast server view
[TPS_SYST_02010]	Usage of SocketConnection attributes in the multicast client view
[TPS_SYST_02096]	Sending of ANY finds for minor version
[TPS_SYST_02113]	Usage of ConsumedEventGroup.applicationEndpoint reference
[TPS_SYST_02120]	runtimeIpAddressConfiguration and runtimePortConfiguration settings for SD SocketConnections
[TPS_SYST_02161]	Role of SystemSignal in inter-ECU client server communication over Ethernet with clients located on different ECUs in case that Com is used
[TPS_SYST_02179]	Server Capability Records
[TPS_SYST_02181]	TTL for SubscribeEventGroupAck Entries
[TPS_SYST_02186]	Client Capability Records
[TPS_SYST_05005]	Relation of GlobalTimeDomain to CommunicationCluster

Table E.50: Deleted Specification Items in R19-11

E.12.4 Added Constraints in R19-11

Number	Heading
[constr_3545]	Mandatory reference to a *Pnc* in case of partialNetwork
[constr_3546]	Mandatory reference to a ConsumedEventGroup in case of serviceControl
[constr_3547]	Mandatory reference to an EventHandler in case of serviceControl
[constr_3548]	EndToEnd profile for both ends of safeTranslation
[constr_3549]	Secure payload for both ends in case of secureTranslation
[constr_3559]	ConsumedServiceInstance.blacklistedVersion is restricted to the usage of minorVersion
[constr_3560]	minimumMinorVersion and ConsumedServiceInstance.minorVersion value
[constr_5029]	J1939NmCluster is not allowed to reference a TtcanCluster

▽

△

Number	Heading
[constr_5030]	Uniqueness of LinOrderedConfigurableFrame.index
[constr_5031]	Uniqueness of FramePid.index
[constr_5032]	Maximal one NmConfig per System is allowed to be defined
[constr_5049]	Ethernet switch packet to traffic class assignment restriction
[constr_5050]	VariableDataPrototype of COM Based Transformer
[constr_5051]	Existence of CanFrameTriggering.identifier in case of bus mirror target
[constr_5053]	Existence of ISignalPort.handleInvalid
[constr_5054]	externalReplacement not applicable for ISignalPort.handleInvalid
[constr_5055]	DataMapping of elements of PRPortPrototypes is not supported
[constr_5058]	Value range for CryptoServiceQueue.queueSize
[constr_5060]	Mapping of a SecuredIPdu into a LinFrame is not allowed
[constr_5061]	EthernetCommunicationConnectors and referencing SocketAddresses shall be in the same VLAN
[constr_5062]	SOME/IP ProvidedServiceInstances of the same serviceInterface on one EcuInstance
[constr_5063]	ProvidedServiceInstance.serviceIdentifier is mandatory
[constr_5064]	ProvidedServiceInstance.majorVersion is mandatory
[constr_5065]	ProvidedServiceInstance.minorVersion is mandatory
[constr_5066]	ProvidedServiceInstance.instanceIdentifier is mandatory
[constr_5067]	ProvidedServiceInstance shall be unique in respect of serviceIdentifier, instanceIdentifier, majorVersion
[constr_5068]	ProvidedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_5069]	ProvidedServiceInstance.remoteUnicastAddress shall be IP Unicast
[constr_5070]	headerIds of ProvidedServiceInstances shall be unique on a SocketAddress per communication direction
[constr_5071]	EventHandler.eventMulticastAddress reference target
[constr_5072]	EventHandler without defined eventMulticastAddress
[constr_5073]	PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast that is aggregated by an EventHandler
[constr_5074]	PduActivationRoutingGroup with eventGroupControlType set to activationMulticast that is aggregated by an EventHandler
[constr_5075]	Allowed references of SoConIPduIdentifiers by PduActivationRoutingGroup with eventGroupControlType set to activationMulticast and allowed SoConIPduIdentifier references
[constr_5076]	PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ProvidedServiceInstance
[constr_5077]	PduActivationRoutingGroup with iPduIdentifierUdp reference that is aggregated by a ProvidedServiceInstance
[constr_5078]	PduTriggerings referenced by a PduActivationRoutingGroup shall be on the same VLAN as the referencing PduActivationRoutingGroup
[constr_5079]	Service communication is restricted to one VLAN

▽

△

Number	Heading
[constr_5080]	ApplicationEndpoints referenced by EventHandlers and by the aggregating ProvidedServiceInstance shall be in the same VLAN
[constr_5081]	ConsumedServiceInstance.serviceIdentifier is mandatory
[constr_5082]	ConsumedServiceInstance.majorVersion is mandatory
[constr_5083]	ConsumedServiceInstance.minorVersion is mandatory
[constr_5084]	ConsumedServiceInstance.instanceIdentifier is mandatory
[constr_5085]	ConsumedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_5086]	ConsumedServiceInstance.remoteUnicastAddress shall be IP Unicast
[constr_5087]	PduActivationRoutingGroup with eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTrigger-Unicast that is referenced by a ConsumedEventGroup
[constr_5088]	PduActivationRoutingGroup with iPduIdentifierTcp reference that is aggregated by a ConsumedServiceInstance
[constr_5089]	PduActivationRoutingGroup with iPduIdentifierUdp reference that is aggregated by a ConsumedServiceInstance
[constr_5090]	ApplicationEndpoints referenced by ConsumedEventGroups and by the aggregating ConsumedServiceInstance shall be in the same VLAN
[constr_5091]	Relevance of tcpRole attribute
[constr_5092]	Relevance of tcpRole attribute
[constr_5093]	pncGatewayType and PhysicalChannel
[constr_5094]	pncGatewayType and ECU
[constr_5095]	Relationship between the timing behavior of the ConsumedEventGroup retry and the timing behavior of an Offer message
[constr_5096]	ConsumedEventGroup with value subscribeEventgroupRetryMax set to 255
[constr_5097]	DltLogChannel.txPduTriggering and DltLogChannel.rxPduTriggering shall point to GeneralPurposeIPdus of category DLT
[constr_5098]	Usage of DltArgument.networkRepresentation
[constr_5099]	Standardized values of DltMessage.messageTypeInfo
[constr_5100]	Compatibility of two MetaDataItemSets
[constr_5101]	Consistent Definition of meta-data
[constr_5104]	Assignment of a FlexrayFrame where allowDynamicLSduLength is set to true
[constr_5105]	Mapping of Pdu with dynamic length in a FlexrayFrame

Table E.51: Added Constraints in R19-11

E.12.5 Changed Constraints in R19-11

Number	Heading
[constr_1207]	Existence of the attribute DataMapping.communicationDirection in the context of a SenderReceiverInterface or TriggerInterface
[constr_1441]	In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer
[constr_1643]	Completeness of the existence of a set of TlvDataIdDefinition.tlvArguments
[constr_1644]	Completeness of the existence of a set of TlvDataIdDefinition.tlvRecordElements
[constr_1645]	Completeness of the existence of a set of TlvDataIdDefinition.tlvImplementationDataTypeElements
[constr_1672]	Existence of TlsCryptoCipherSuite.certificate and TlsCryptoCipherSuite.pskIdentity in the *server* role
[constr_3012]	Overlapping of Pdus is prohibited
[constr_3013]	FlexrayFrame length shall not be exceeded
[constr_3014]	Overlapping of updateIndicationBits for Pdus is prohibited
[constr_3267]	PduTriggerings in Service Discovery StaticSocketConnections
[constr_3268]	Service Discovery StaticSocketConnection aggregation by an ApplicationEndpoint
[constr_3269]	Service Discovery StaticSocketConnection remoteAddress reference to a TpPort
[constr_3270]	Service Discovery SocketConnection remoteAddress reference to an IP Address
[constr_3272]	SoConIPduIdentifier.headerId setting for SD StaticSocketConnections
[constr_3273]	Service Discovery multicast StaticSocketConnection's aggregation by an ApplicationEndpoint
[constr_3274]	Service Discovery unicast StaticSocketConnection's aggregation by an ApplicationEndpoint
[constr_3276]	Prohibition of usage of allowedIPv6ExtHeaders in IPv4 SocketAddress
[constr_3277]	Restriction of usage of IPv6ExtHeaderFilterLists in IPv6 SocketAddress
[constr_3297]	Prohibition of usage of allowedTcpOptions in Udp SocketAddress
[constr_3299]	SocketAddress.pathMtuDiscoveryEnabled setting dependency
[constr_3311]	Usage of SocketAddress.flowLabel
[constr_3315]	The value of V0 in BufferProperties.bufferComputation setting for a COM Based transformer
[constr_3322]	Consistent setting of SoConIPduIdentifier.pduCollectionSemantics in the context of one SocketAddress
[constr_3325]	SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute values for predefined categories
[constr_3379]	Multiple SocketAddress entries with the same IP Address, Protocol and Port in the context of a given EcuInstance
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup

▽

△

Number	Heading
[constr_3516]	limitation of Frame.frameLength for CAN L-PDUs

Table E.52: Changed Constraints in R19-11

E.12.6 Deleted Constraints in R19-11

Number	Heading
[constr_3055]	SystemSignalGroup in a complete System Description
[constr_3059]	Mandatory DataMapping on the receiver side for elements of a composite data type
[constr_3063]	Usage of portNumber and dynamicallyAssigned with value “true” is mutually exclusive
[constr_3064]	Usage of serviceInstance, eventHandler and eventGroup references
[constr_3087]	DataMapping to PRPortPrototype
[constr_3088]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3176]	Value range of windowSize
[constr_3177]	Dependency between maxErrorStateValid, maxErrorStateInit and maxErrorStateInvalid
[constr_3178]	Dependency between minOkStateValid, minOkStateInit and minOkStateInvalid
[constr_3179]	Dependency between minOkStateInit, maxErrorStateInit and windowSize
[constr_3180]	Dependency between minOkStateValid, maxErrorStateValid and windowSize
[constr_3181]	Dependency between minOkStateInvalid, maxErrorStateInvalid and windowSize
[constr_3201]	eventGroupIdentifier in ConsumedEventGroups that are referenced by the same EventHandler
[constr_3259]	Allowed use of SdServerConfig.capabilityRecord
[constr_3260]	Allowed use of SdClientConfig.capabilityRecord
[constr_3271]	clientIpAddrFromConnectionRequest and clientPortFromConnectionRequest settings for SD SocketConnections
[constr_3400]	Usage of SdClientConfig attributes in ConsumedServiceInstance and ConsumedEventGroup
[constr_3401]	Usage of SdServerConfig attributes in ProvidedServiceInstance and EventHandler

Table E.53: Deleted Constraints in R19-11

E.13 Constraint and Specification Item History of this document according to AUTOSAR R20-11

E.13.1 Added Specification Items in R20-11

Number	Heading
[TPS_SYST_02275]	Relation between EndToEndTransformationDescription and EndToEndTransformationComSpecProps
[TPS_SYST_02276]	Modeling of LIN master request frames
[TPS_SYST_02277]	Modeling of LIN slave response frames
[TPS_SYST_02278]	Existence of SystemSignals in a SystemSignalGroup that are not referenced by a SenderRecCompositeTypeMapping
[TPS_SYST_02279]	SenderReceiverInterface.dataElement is typed by a "new-world" variable-size ApplicationArrayDataType and a DataTypeMap exists
[TPS_SYST_02280]	SenderReceiverInterface.dataElement is typed by a "new-world" variable-size ImplementationDataType
[TPS_SYST_02281]	Definition of SecuredIPdu.authDataFreshnessStartPosition
[TPS_SYST_02282]	Definition of SecuredIPdu.messageLinkPosition
[TPS_SYST_02283]	Collection of ServiceInterface elements
[TPS_SYST_02284]	Event in a ServiceInterface
[TPS_SYST_02285]	Method in a ServiceInterface
[TPS_SYST_02286]	"fire & forget" method with data in a ServiceInterface
[TPS_SYST_02287]	"Fire & forget" method without data in a ServiceInterface
[TPS_SYST_02288]	"Fire & forget" method in a ServiceInterface
[TPS_SYST_02289]	Field in a ServiceInterface
[TPS_SYST_02290]	Field elements
[TPS_SYST_02291]	Field Notifier
[TPS_SYST_02292]	Field Getter
[TPS_SYST_02293]	Field Setter
[TPS_SYST_02294]	Serialization of ServiceInterfaces using ComBasedTransformer
[TPS_SYST_02295]	Serialization of ServiceInterfaces using SomeipTransformer
[TPS_SYST_02296]	eventGroupControlType of a unicast Event
[TPS_SYST_02297]	eventGroupControlType of a multicast Event
[TPS_SYST_02298]	eventGroupControlType of a unicast Field
[TPS_SYST_02299]	Modeling of 10Base-T1S networks
[TPS_SYST_02300]	Enabling of PLCA on a CouplingPort
[TPS_SYST_02301]	CSMA/CD configured nodes on a 10BASE-T1S network
[TPS_SYST_02302]	Definition of multicast only reception of an EventGroup
[TPS_SYST_02303]	Modeling of DoIpRoutingActivations

▽

△

Number	Heading
[TPS_SYST_02304]	Conversion of discrete parts of a CompuMethod on signal level in SenderReceiverToSignalMapping
[TPS_SYST_02305]	Relevance of attribute TextTableMapping.mappingDirection in an aggregation by SenderReceiverToSignalMapping
[TPS_SYST_02306]	Conversion of discrete parts of a CompuMethod on signal level in SenderRecRecordElementMapping and SenderRecArrayTypeMapping
[TPS_SYST_02307]	Relevance of attribute TextTableMapping.mappingDirection in an aggregation by SenderRecRecordElementMapping or SenderRecArrayTypeMapping
[TPS_SYST_02308]	TextTableMapping defined in the context of SenderRecArrayTypeMapping
[TPS_SYST_02309]	RTE fan-out support for a SystemSignalGroup
[TPS_SYST_02310]	Pdu routing with IPduMapping.pduMaxLength
[TPS_SYST_02311]	IPduMapping.pduMaxLength relying on the environment length configuration
[TPS_SYST_02312]	Ports for outermost composition of a SW_CLUSTER_SYSTEM_DESCRIPTION
[TPS_SYST_02313]	Ecu Extract derived from ECU_SYSTEM_DESCRIPTION covers an EcuInstance
[TPS_SYST_02314]	Ecu Extract derived from SW_CLUSTER_SYSTEM_DESCRIPTION covers a subset of an EcuInstance
[TPS_SYST_02315]	Definition of a software cluster on the *AUTOSAR classic platform*
[TPS_SYST_02316]	Semantics of meta-class SwComponentPrototypeAssignment
[TPS_SYST_02317]	References from CpSoftwareCluster to CompositionSwComponentType and SwComponentPrototype
[TPS_SYST_02318]	Membership in System
[TPS_SYST_02319]	Semantics of attribute CpSoftwareCluster.category
[TPS_SYST_02320]	Kinds of CpSoftwareClusterResources
[TPS_SYST_02321]	Assignment of CpSoftwareClusterCommunicationResources to CpSoftwareClusters in the context of a SwComponentPrototype
[TPS_SYST_02322]	PortElementToCommunicationResourceMapping aggregated by SystemMapping supersedes PortElementToCommunicationResourceMapping aggregated by CpSoftwareClusterMappingSet
[TPS_SYST_02323]	Assignment of CpSoftwareClusterServiceResources to CpSoftwareClusters
[TPS_SYST_02324]	CpSoftwareClusterServiceResource provided by the CpSoftwareCluster
[TPS_SYST_02325]	CpSoftwareClusterServiceResource required by the CpSoftwareCluster
[TPS_SYST_02326]	Aggregation possibilities of CpSoftwareClusterToResourceMapping
[TPS_SYST_02327]	Role of the Software Cluster Binary Manifest
[TPS_SYST_02328]	Semantics of meta-class CpSoftwareClusterBinaryManifestDescriptor
[TPS_SYST_02329]	Provison of a Software Cluster's ID

▽

△

Number	Heading
[TPS_SYST_02330]	Possible values of attribute CpSoftwareClusterBinaryManifest-Descriptor.category
[TPS_SYST_02331]	Definition of provided resource in the context of the Software Cluster Binary Manifest
[TPS_SYST_02332]	Definition of required resource in the context of the Software Cluster Binary Manifest
[TPS_SYST_02333]	Purpose of meta-class BinaryManifestResourceDefinition
[TPS_SYST_02334]	Semantics of meta-class BinaryManifestItem
[TPS_SYST_02335]	Semantics of aggregation BinaryManifestItem.auxiliaryField
[TPS_SYST_02336]	Semantics of meta-class BinaryManifestItemDefinition
[TPS_SYST_02337]	Semantics of aggregation BinaryManifestItemDefinition.auxiliary-FieldDefinition
[TPS_SYST_02338]	Relation between BinaryManifestItemDefinition and Binary-ManifestItem
[TPS_SYST_02339]	Standardized values of attribute BinaryManifestAddressable-Object.category
[TPS_SYST_02340]	Semantics of abstract meta-class BinaryManifestItemValue
[TPS_SYST_02341]	Semantics of the aggregation of meta-class BinaryManifestItem-PointerValue in the role defaultValue
[TPS_SYST_02342]	Semantics of meta-class BinaryManifestMetaDataField
[TPS_SYST_02343]	System with category SW_CLUSTER_SYSTEM_DESCRIPTION
[TPS_SYST_02344]	SW_CLUSTER_SYSTEM_DESCRIPTION content
[TPS_SYST_02345]	Assignment of CpSoftwareClusterCommunicationResources to CpSoftwareClusters in the context of a SwComponentType
[TPS_SYST_02346]	CpSoftwareClusterToResourceMapping aggregated by System-Mapping supersedes CpSoftwareClusterToResourceMapping aggregated by CpSoftwareClusterMappingSet
[TPS_SYST_02347]	Mapping of CpSoftwareClusterResource to ApplicationPartition
[TPS_SYST_02348]	Mapping of CpSoftwareCluster to EcuInstance
[TPS_SYST_02349]	Recommended configuration settings for E2E Profile 4m configuration setting A
[TPS_SYST_02350]	Recommended configuration settings for E2E Profile 7m configuration setting A
[TPS_SYST_02351]	Selector field signal initial values in case no application writing the selector field signal
[TPS_SYST_02352]	Triggering in case of application writing the selector field signal
[TPS_SYST_02353]	No support for trigger transmit in case of application writing the selector field signal
[TPS_SYST_02354]	No support for Just-In-Time update of dynamic parts in case of application writing the selector field signal
[TPS_SYST_02355]	TransmissionModeDeclaration in case of application writing the selector field signal

▽

△

Number	Heading
[TPS_SYST_02356]	Only one TransmissionModeCondition in case of application writing the selector field signal
[TPS_SYST_02357]	RTE fan-in support for a SystemSignal
[TPS_SYST_02358]	RTE fan-in support for a SystemSignalGroup
[TPS_SYST_02359]	Size of String Length Fields
[TPS_SYST_02360]	Size of a length field for a chosen string
[TPS_SYST_03052]	Enabling of wake-up and sleep mechanism
[TPS_SYST_03053]	Semantics of EthernetCommunicationController.slaveQualified-UnexpectedLinkDownTime
[TPS_SYST_03054]	Semantics of EthernetCluster.couplingPortStartupActiveTime
[TPS_SYST_03055]	Semantics of EthernetCluster.couplingPortSwitchoffDelay

Table E.54: Added Specification Items in R20-11

E.13.2 Changed Specification Items in R20-11

Number	Heading
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_01109]	RTE fan-out support for a SystemSignal
[TPS_SYST_01137]	Several DataMappings may be defined for the same SystemSignal
[TPS_SYST_01139]	Ecu Extract derived from System Description or System Extract covers exactly one EcuInstance
[TPS_SYST_02073]	EndToEndTransformationDescription.profileName
[TPS_SYST_02092]	Size of Array Length Fields
[TPS_SYST_02123]	Size of a length field for a chosen array
[TPS_SYST_02194]	Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, and PROFILE_44
[TPS_SYST_02197]	Assigning communication graphs to RTE Implementation Plug-Ins
[TPS_SYST_02234]	SOME/IP Service search
[TPS_SYST_02235]	ConsumedServiceInstance deployment
[TPS_SYST_02272]	IPsec Internet Key Exchange protocol configuration
[TPS_SYST_03038]	Definition of transmission triggers for translations with different sources
[TPS_SYST_05016]	Assignment of TLV data ids
[TPS_SYST_05025]	Standardized values of CryptoServicePrimitive.algorithmFamily and CryptoServiceKey.algorithmFamily

Table E.55: Changed Specification Items in R20-11

E.13.3 Deleted Specification Items in R20-11

Number	Heading
[TPS_SYST_02105]	ISignalGroup and ISignal referenced from ISignalTriggering

Table E.56: Deleted Specification Items in R20-11

E.13.4 Added Constraints in R20-11

Number	Heading
[constr_3600]	Setting of EthernetCommunicationController.slaveActAsPassive-CommunicationSlave
[constr_3601]	Mandatory attributes of EthernetWakeupSleepOnDatalineConfig
[constr_3602]	Existence of wakeupForwardLocalEnabled
[constr_3603]	Existence of wakeupLocalDurationTime
[constr_3604]	Existence of wakeupForwardRemoteEnabled
[constr_3605]	Existence of wakeupLocalDetectionTime
[constr_3606]	Values of wakeupLocalDurationTime and wakeupLocalDetectionTime
[constr_3607]	Existence of sleepRepetitionDelayOfSleepRequest
[constr_3608]	Existence of wakeupRepetitionDelayOfWakeupRequest
[constr_3609]	Values of wakeupLocalDurationTime in the context of a CouplingElement
[constr_3610]	Values of wakeupLocalDetectionTime in the context of a CouplingElement
[constr_3611]	Existence of EthernetCommunicationController.slaveQualified-UnexpectedLinkDownTime
[constr_3615]	Existence of EthernetCluster.couplingPortSwitchoffDelay
[constr_3616]	Value of EthernetCluster.couplingPortSwitchoffDelay
[constr_3617]	Existence of EthernetCluster.couplingPortStartupActiveTime
[constr_3618]	Value of EthernetCluster.couplingPortStartupActiveTime
[constr_3620]	GlobalTimeDomain.networkSegmentId only applicable to GlobalTime sub domains
[constr_3621]	Value range of GlobalTimeDomain.networkSegmentId
[constr_5106]	ISignalGroup and ISignal referenced from ISignalTriggering
[constr_5109]	Conditions for the explicit mapping of an ISignal to the pncVector
[constr_5110]	Search for a collection of ServiceInstances is not supported
[constr_5111]	Existence of references TlvDataIdDefinition.tlvArgument, TlvData-IdDefinition.tlvRecordElement, and TlvDataIdDefinition.tlv-ImplementationDataTypeElement
[constr_5112]	ImplementationDataType needs to be defined if a “new-world” variable-size ApplicationArrayDataType is mapped to a single SystemSignal
[constr_5113]	Mapping of “old-world” variable size arrays to a single SystemSignal is not supported.
[constr_5114]	Semantics of InterpolationRoutine.isDefault

▽

△

Number	Heading
[constr_5116]	Uniqueness of the symbols of software-components and BSW modules
[constr_5117]	Client-Server communication over Ethernet
[constr_5118]	Value range of UdpProps.udpTtl
[constr_5119]	Value range of TcpProps.tcpTtl
[constr_5120]	Value range of TcpProps.tcpDelayedAckTimeout
[constr_5121]	Value range of TcpProps.tcpSynMaxRtx
[constr_5122]	Value range of TcpProps.tcpMaxRtx
[constr_5123]	Value range of TcpProps.tcpKeepAliveProbesMax
[constr_5124]	Value range of TcpProps.tcpReceiveWindowMax
[constr_5125]	Value range of TcpIpIcmpv4Props.tcpIpIcmpV4Ttl
[constr_5126]	Value range of Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup
[constr_5127]	Value range of Ipv4FragmentationProps.tcpIpIpNumFragments
[constr_5128]	Value range of Ipv4FragmentationProps.tcpIpIpNumReassDgrams
[constr_5129]	Value range of Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount
[constr_5130]	Value range of Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize
[constr_5131]	Value range of Ipv6FragmentationProps.tcpIpIpReassemblyTimeout
[constr_5132]	Value range of Ipv6FragmentationProps.tcpIpIpReassemblySegment-Count
[constr_5133]	Value range of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount
[constr_5134]	Value range of Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize
[constr_5135]	Value range of Dhcpv6Props.tcpIpDhcpV6CnfDelayMin and Dhcpv6 Props.tcpIpDhcpV6CnfDelayMax
[constr_5136]	Value range of Dhcpv6Props.tcpIpDhcpV6InfDelayMin and Dhcpv6 Props.tcpIpDhcpV6InfDelayMax
[constr_5137]	Value range of Dhcpv6Props.tcpIpDhcpV6SolDelayMin and Dhcpv6 Props.tcpIpDhcpV6SolDelayMax
[constr_5138]	Value range of Ipv6NdpProps.tcpIpNdpSlaacDadNumberOfTransmissions
[constr_5139]	Value range of Ipv6NdpProps.tcpIpNdpSlaacDadRetransmissionDelay
[constr_5140]	Value range of Ipv6NdpProps.tcpIpNdpDefaultReachableTime
[constr_5141]	Value range of Ipv6NdpProps.tcpIpNdpDefaultRetransTimer
[constr_5142]	Value range of Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations
[constr_5143]	Value range of Ipv6NdpProps.tcpIpNdpNumMulticastSolicitations
[constr_5144]	Value range of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTime
[constr_5145]	Value range of Ipv6NdpProps.tcpIpNdpMinRandomFactor
[constr_5146]	Value range of Ipv6NdpProps.tcpIpNdpMaxRandomFactor
[constr_5147]	Value range of Ipv6NdpProps.tcpIpNdpDestinationCacheSize
[constr_5148]	Value range of Ipv6NdpProps.tcpIpNdpPrefixListSize
[constr_5149]	Value range of Ipv6NdpProps.tcpIpNdpDefaultRouterListSize
[constr_5151]	Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations

▽

△

Number	Heading
[constr_5152]	Value range of Ipv6NdpProps.tcpIpNdpMaxRtrSolicitationDelay
[constr_5153]	Value range of Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval
[constr_5154]	Value range of TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit
[constr_5157]	Mixing of Point-To-Point and Multi-Drop is not allowed in a CouplingPortConnection
[constr_5158]	Usage of plcaProps only allowed on 10BASE-T1S networks
[constr_5159]	Mandatory CouplingPortConnection settings if multi-drop feature is used
[constr_5160]	Mandatory PlcaProps settings if multi-drop feature is used
[constr_5162]	Valid TextTableMapping in the context of SenderRecRecordElementMapping
[constr_5163]	Existence of attribute IPSecRule.headerType
[constr_5164]	Existence of attribute IPSecRule.ipProtocol
[constr_5165]	Existence of attribute IPSecRule.policy
[constr_5166]	Existence of IPduMapping.pduMaxLength
[constr_5167]	pncGatewayType and ECU over the whole system
[constr_5168]	pncGatewayType passive and connected ECUs
[constr_5169]	pncGatewayType and (routing) paths
[constr_5170]	nmPassiveModeEnabled and dynamicPncToChannelMappingEnabled
[constr_5171]	Existence of the attribute DataMapping.communicationDirection in ClientServerToSignalMapping
[constr_5175]	RtePluginProps shall reference at least one EcucContainerValue representing a RteRipsPlugin
[constr_5176]	Existence of CpSoftwareCluster of category HOST_SOFTWARE_CLUSTER on one EcuInstance
[constr_5177]	Validity of reference CpSoftwareClusterToEcuInstanceMapping.swCluster
[constr_5178]	Existence of attribute CpSoftwareClusterResource.globalResourceId
[constr_5179]	Existence of attribute CpSoftwareClusterResource.isMandatory
[constr_5180]	Allowed values for CpSoftwareClusterResource.globalResourceId
[constr_5181]	Existence of attribute CpSoftwareClusterServiceResource.category
[constr_5182]	PRPortPrototypes are excluded as CpSoftwareCluster interfaces
[constr_5183]	PortElementToCommunicationResourceMapping shall reference exactly one element of a PortInterface
[constr_5184]	CpSoftwareClusterServiceResource can be provided only once on an EcuInstance
[constr_5185]	Existence of attribute BinaryManifestProvideResource.globalResourceId
[constr_5186]	Existence of attribute BinaryManifestProvideResource.resourceGuardValue
[constr_5187]	Existence of attribute BinaryManifestProvideResource.supportsMultipleNotifierSets
[constr_5188]	Existence of attribute BinaryManifestProvideResource.numberOfNotifierSets
[constr_5189]	Existence of reference BinaryManifestProvideResource.resourceDefinition

▽

△

Number	Heading
[constr_5190]	Existence of aggregation BinaryManifestProvideResource.item
[constr_5191]	Consequence of attribute BinaryManifestProvideResource.item.category
[constr_5192]	Existence of attribute BinaryManifestRequireResource.globalResourceId
[constr_5193]	Existence of attribute BinaryManifestRequireResource.resourceGuard-Value
[constr_5194]	Existence of reference BinaryManifestRequireResource.resource-Definition
[constr_5195]	Existence of aggregation BinaryManifestRequireResource.item
[constr_5196]	Consequence of attribute BinaryManifestRequireResource.item.category
[constr_5197]	Existence of aggregation BinaryManifestResourceDefinition.item-Definition
[constr_5198]	Allowed BinaryManifestResource.resourceDefinition
[constr_5199]	Consequence of attribute BinaryManifestItem.auxiliaryField.category
[constr_5200]	Existence of attribute BinaryManifestItemDefinition.category
[constr_5201]	Existence of attribute BinaryManifestItemDefinition.size
[constr_5202]	Existence of attribute BinaryManifestItemNumericalValue.value
[constr_5203]	Existence of attribute BinaryManifestItemPointerValue.symbol
[constr_5204]	Existence of attribute BinaryManifestMetaDataField.category
[constr_5205]	Existence of attribute BinaryManifestMetaDataField.size
[constr_5206]	Existence of attribute BinaryManifestMetaDataField.symbol
[constr_5207]	Existence of attribute BinaryManifestMetaDataField.address
[constr_5208]	Existence of System.swCluster
[constr_5209]	Existence of reference CpSoftwareCluster.swComponentAssignmentswCom-ponent
[constr_5210]	Existence of reference SystemMapping.portElementToComResourceMapping
[constr_5211]	Existence of reference PortElementToCommunicationResource-Mapping.communicationResource
[constr_5212]	Existence of reference SystemMapping.resourceToApplicationPartition-Mapping
[constr_5213]	Existence of reference CpSoftwareClusterResourceToApplication-PartitionMapping.applicationPartition
[constr_5214]	Existence of reference CpSoftwareClusterResourceToApplication-PartitionMapping.resource
[constr_5215]	Existence of reference CpSoftwareClusterToResourceMapping.service-Resource
[constr_5216]	Existence of reference CpSoftwareClusterToResourceMapping.requester and/or provider
[constr_5217]	Existence of attribute BinaryManifestMetaDataField.value
[constr_5218]	Existence of attribute BinaryManifestItemPointerValue.address
[constr_5219]	CpSoftwareCluster shall only be mapped to one EcuInstance
[constr_5220]	Multiplicity of EndToEndTransformationISignalProps.sourceId in PRO-FILE_04m and PROFILE_07m

▽

△

Number	Heading
[constr_5221]	Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, and PROFILE_22
[constr_5222]	Mandatory elements of UdpNmCluster
[constr_5223]	Mandatory elements of UdpNmNode
[constr_5224]	UdpNmNode.nmMsgCycleOffset < UdpNmCluster.nmMsgCycleTime
[constr_5225]	UdpNmCluster.nmNetworkTimeout multiple of UdpNmCluster.nmMsgCycleTime
[constr_5226]	UdpNmCluster.nmRepeatMessageTime multiple of UdpNmCluster.nmMsgCycleTime
[constr_5229]	Existence of attribute E2EProfileCompatibilityProps.transitToInvalidExtended is mandatory for each EndToEndTransformationDescription
[constr_5231]	Allowed values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment
[constr_5232]	Triggering in case of application writing the selector field signal
[constr_5233]	Usage of invalidValue in case of application writing the selector field signal
[constr_5235]	Maximum Frame.frameLength of the used bus protocol shall not be exceeded
[constr_5236]	Restriction of IPduMapping.pduMaxLength
[constr_5244]	Value of attribute SOMEIPTransformationISignalProps.sizeOfArrayLengthFields
[constr_5245]	Value of attribute SOMEIPTransformationISignalProps.sizeOfStringLengthFields
[constr_5246]	SOME/IP Transformation settings for strings in the context of an ISignal
[constr_5247]	Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField
[constr_5248]	Value of attribute DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField

Table E.57: Added Constraints in R20-11

E.13.5 Changed Constraints in R20-11

Number	Heading
[constr_1652]	Definition of static length fields sizes in case of TLV usage
[constr_1653]	Identical values for length fields sizes in case of TLV usage
[constr_3000]	valid SenderRecCompositeTypeMappings
[constr_3004]	Clustering and separation shall be exclusive
[constr_3039]	pncIdentifier range
[constr_3041]	pncVectorOffset range
[constr_3042]	pncVectorLength range

▽

△

Number	Heading
[constr_3069]	Allowed CanNmCluster.nmNidPosition values
[constr_3070]	Allowed CanNmCluster.nmCbvPosition values
[constr_3078]	Allowed UdpNmCluster.nmNidPosition values
[constr_3079]	Allowed UdpNmCluster.nmCbvPosition values
[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3151]	BufferProperties.headerLength settings for an E2E transformer used in combination with a SOME/IP transformer
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3159]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_04, PROFILE_04m and PROFILE_44
[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, and PROFILE_44
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m and PROFILE_44
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m and PROFILE_44
[constr_3169]	EndToEndTransformationDescription.offset value in PROFILE_02 and PROFILE_22
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m and PROFILE_44
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, and PROFILE_44
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, and PROFILE_44
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, and PROFILE_44
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, and PROFILE_44 or dataIdMode different from lower12Bit
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, and PROFILE_44
[constr_3212]	Limitation of DoIpTpConnection.tpSdu
[constr_3218]	Range of Size of Array Length Fields
[constr_3282]	SOME/IP Transformation settings for arrays in the context of an ISignal

▽

△

Number	Heading
[constr_3316]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07, PROFILE_08 and PROFILE_07m
[constr_5092]	Local and remoteAddress of a StaticSocketConnection shall define the same transport protocol

Table E.58: Changed Constraints in R20-11

E.13.6 Deleted Constraints in R20-11

Number	Heading
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs in case of networks other than Ethernet
[constr_3453]	Uniqueness of header Id in case of acceptAll ContainerIPdu
[constr_3483]	The same PhysicalChannel is not allowed to be the source and the target of managedPhysicalChannel references

Table E.59: Deleted Constraints in R20-11

E.14 Constraint and Specification Item History of this document according to AUTOSAR R21-11

E.14.1 Added Specification Items in R21-11

Number	Heading
[TPS_SYST_02361]	PduR Fan-out of SecuredIPdu with useAsCryptographicIPdu = true
[TPS_SYST_02362]	Relevance of attribute EthernetCluster.baudrate
[TPS_SYST_02363]	messageId of AssignFrameId and UnassignFrameId
[TPS_SYST_02364]	Scope of the System
[TPS_SYST_02365]	No support of Com Based Transformer for Trigger communication
[TPS_SYST_02366]	NID/CBV signals shall be ignored by Ecuc tools
[TPS_SYST_02367]	Execution Order of RTEEvents on a EcuInstance
[TPS_SYST_02368]	RTEEvent pairing constraint in Software Composition context
[TPS_SYST_02369]	RTEEvent separation constraint in Software Composition context
[TPS_SYST_02370]	RTEEvent pairing constraint in System context
[TPS_SYST_02371]	RTEEvent separation constraint in System context
[TPS_SYST_02372]	Precedence of ContainedIPduProps settings
[TPS_SYST_02373]	Assignment of a Dlt Ecu Identifier to an EcuInstance

▽

△

Number	Heading
[TPS_SYST_02374]	Assignment of DltMessage to DltLogChannels
[TPS_SYST_02375]	Definition of DltLogChannels source
[TPS_SYST_02376]	Pdu Router fan-in support
[TPS_SYST_02377]	Consistent setting of Service Interface Version
[TPS_SYST_02378]	Optional method arguments
[TPS_SYST_03056]	Monitoring of the released *partial networks* status in case of partialNetwork for required service instance
[TPS_SYST_03057]	Monitoring of the released *partial networks* status in case of partialNetwork for provided service instance
[TPS_SYST_03058]	*Auto require* for ConsumedServiceInstance in case of service instance with serviceControl
[TPS_SYST_03059]	At most one queued source input in case of signal/service translation from several sources
[TPS_SYST_03060]	Source input with queued semantics shall have transmissionTrigger set to true
[TPS_SYST_03061]	No support for queued reception semantics in combination with periodic communication
[TPS_SYST_03062]	Definition of a primitive target for SignalServiceTranslationElementProps
[TPS_SYST_03063]	Definition of a composite target for SignalServiceTranslationElementProps
[TPS_SYST_03064]	Enabling of multicast subscription
[TPS_SYST_03065]	Static definition of multicast subscription at the server
[TPS_SYST_03066]	Mix of static definition consisting of multicast subscription clients and unicast subscription clients at the server
[TPS_SYST_03067]	Definition of pncVectorOffset
[TPS_SYST_03068]	Definition of pncVectorLength
[TPS_SYST_03069]	User data shall be defined within empty space of the NmPdu
[TPS_SYST_03070]	User data shall be before the PncBitVector or after the PncBitVector
[TPS_SYST_03071]	Available space of user data with PncBitVector
[TPS_SYST_03072]	Available space of user data without PncBitVector
[TPS_SYST_03073]	Derivation of NmPnFilterMaskByte

Table E.60: Added Specification Items in R21-11

E.14.2 Changed Specification Items in R21-11

Number	Heading
[TPS_SYST_01005]	Definition of `EcuInstance`
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_02044]	Buffer computation of transformer
[TPS_SYST_02064]	Reception acceptance of contained `IPdu`s
[TPS_SYST_02073]	`EndToEndTransformationDescription.profileName`
[TPS_SYST_02097]	Basic definition of contained `IPdu`s
[TPS_SYST_02099]	Relation between `ContainerIPdu` and contained `PduTriggering`s on sender side
[TPS_SYST_02117]	Length of `GeneralPurposePdu` with category SD
[TPS_SYST_02136]	Serialization based on the network representation
[TPS_SYST_02172]	Modeling of `SecuredIPdu` in case `useAsCryptographicIPdu` is set to false
[TPS_SYST_02173]	Modeling of `SecuredIPdu` in case `useAsCryptographicIPdu` is set to true
[TPS_SYST_02194]	Identification of E2E protected data in case of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m
[TPS_SYST_02196]	`PduTriggering` is referenced by several `ContainerIPdu`s
[TPS_SYST_02264]	Usage of `DltLogChannel`
[TPS_SYST_02283]	Collection of `ServiceInterface` elements
[TPS_SYST_02288]	“Fire & forget” method in a `ServiceInterface`
[TPS_SYST_02289]	`Field` in a `ServiceInterface`
[TPS_SYST_02339]	Standardized values of attribute `BinaryManifestAddressableObject.category`
[TPS_SYST_03022]	`autoAvailable` setting for provided service instance with `translationStart`
[TPS_SYST_03023]	`autoRequire` setting for required service instance with `translationStart`
[TPS_SYST_03024]	`autoRequire` setting for required event groups of required service instance with `translationStart`
[TPS_SYST_03026]	Monitoring of the `requested` *partial networks* status in case of partialNetwork for provided service instance
[TPS_SYST_03027]	Monitoring of the `requested` *partial networks* status in case of partialNetwork for required service instance
[TPS_SYST_03028]	*Auto require* for `controlConsumedEventGroup` in case of service instance with `serviceControl`
[TPS_SYST_03029]	*Offer* for a provided translated service instance with `serviceControl`
[TPS_SYST_03030]	*Stop offer* for a provided service instance with `serviceControl`
[TPS_SYST_03032]	Data transmission in case of *E_OK* safe signal reception

▽

△

Number	Heading
[TPS_SYST_03039]	Full translation before transmission triggering
[TPS_SYST_03042]	Periodic call in case of ReceiverComSpec.ReceiverComSpec.receptionProps.dataUpdatePeriod
[TPS_SYST_03043]	Periodic call in case of SenderComSpec.transmissionProps.dataUpdatePeriod
[TPS_SYST_03044]	Handling of safe signal/service translation in one software component
[TPS_SYST_03045]	Support for safe signal/service translation
[TPS_SYST_03046]	Support for safe signal/service translation with same or different E2E profiles
[TPS_SYST_03051]	Data filter inside the signal/service translation software component
[TPS_SYST_05029]	Semantics of meta-class TlsCryptoServiceMapping
[TPS_SYST_05031]	Existence of TlsCryptoCipherSuite.keyExchange vs. TlsCryptoServiceMapping.keyExchange
[TPS_SYST_05033]	Existence of TlsCryptoCipherSuite.certificate and TlsCryptoCipherSuite.pskIdentity in the *client* role

Table E.61: Changed Specification Items in R21-11

E.14.3 Deleted Specification Items in R21-11

Number	Heading
[TPS_SYST_02043]	Buffer computation of transformer
[TPS_SYST_02133]	BufferProperties.bufferComputation setting for a COM Based transformer
[TPS_SYST_02147]	Implicit definition of pncVector at NmPdu
[TPS_SYST_02165]	Derivation of CanNmPnFilterMaskByte
[TPS_SYST_02166]	Derivation of UdpNmPnFilterMaskByte
[TPS_SYST_02167]	Derivation of FrNmPnFilterMaskByte

Table E.62: Deleted Specification Items in R21-11

E.14.4 Added Constraints in R21-11

Number	Heading
[constr_1001]	Value of dataId shall be unique
[constr_3651]	No element in case translationTarget is primitive
[constr_3652]	Allowed sub-classes of DataPrototypeReference in the context of signal/service translation
[constr_3653]	Consistent translationTarget and element in case ApplicationDataType is used
[constr_3654]	Consistent translationTarget and element in case ImplementationDataType is used
[constr_3655]	Supported filter types for primitive SignalServiceTranslationElementProps
[constr_3656]	Supported filter types for composite SignalServiceTranslationElementProps
[constr_3668]	Existence of TlsCryptoCipherSuite.cipherSuiteShortLabel
[constr_3669]	eventMulticastSubscriptionAddress shall refer to a multicast address
[constr_3670]	No support for parallel localUnicastAddress and eventMulticastSubscriptionAddress
[constr_3671]	remoteMulticastSubscriptionAddress shall refer to a multicast address
[constr_3672]	No support for methods in multicast subscription at the client
[constr_3673]	No support for methods in multicast subscription at the server static configuration
[constr_3685]	Allowed values for each element of pncFilterArrayMask
[constr_3686]	Allowed number of entries for pncFilterArrayMask
[constr_3687]	Limited value range for NmCluster.pncClusterVectorLength
[constr_4000]	Local communication of mode switches
[constr_5249]	Existence of Pdu.length
[constr_5251]	CouplingPort.connectionNegotiationBehavior shall exist
[constr_5252]	LinSlaveConfig.protocolVersion shall exist
[constr_5253]	Value range of ISignal.length
[constr_5254]	Value range of MultiplexedIPdu.selectorFieldLength
[constr_5258]	TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping in case DataTransformation is used
[constr_5259]	PduTriggerings and FrameTriggerings of SecuredIPdu with useAsCryptographicIPdu = true
[constr_5262]	SystemSignal used for Trigger communication shall not be part of any SystemSignalGroup
[constr_5263]	NetworkEndpoint.networkEndpointAddress restriction for IPv4
[constr_5264]	NetworkEndpoint.networkEndpointAddress restriction for IPv6
[constr_5265]	NetworkEndpoint.networkEndpointAddress restriction
[constr_5266]	VariableDataPrototype of NvDataInterface shall not be mapped to a SystemSignal



△

Number	Heading
[constr_5267]	VariableDataPrototype of NvDataInterface shall not be mapped to a SystemSignal via a delegation to a PortPrototype with a SenderReceiverInterface
[constr_5268]	Existence of ContainedIPduProps.containedPduTriggering reference
[constr_5269]	Exclusion of ContainedIPduProps.containedPduTriggering reference
[constr_5270]	Exclusive usage of ContainerIPdu.containedPduTriggering and ContainerIPdu.containedIPduTriggeringProps
[constr_5271]	Existence of attribute BinaryManifestItem.isUnused
[constr_5272]	Value of attribute BinaryManifestItem.isUnused
[constr_5273]	One ISignalTriggering pair allowed per EthernetPhysicalChannel for a ClientServerOperation
[constr_5274]	ISignalTriggerings that represent the callSignal and returnSignal of the same ClientServerOperation on a PhysicalChannel shall be referenced by the same ClientServerToSignalMapping
[constr_5306]	Restriction of DltLogChannel.logChannelId attribute value
[constr_5307]	Existence of DltLogChannel.logChannelId
[constr_5308]	Existence of DltLogChannel.nonVerboseMode
[constr_5309]	Existence of DltConfig.sessionIdSupport
[constr_5310]	Existence of DltConfig.timestampSupport
[constr_5311]	Existence of DltLogChannel.logTraceDefaultLogThreshold
[constr_5312]	Existence of DltLogChannel.defaultTraceState
[constr_5313]	Existence of DltLogChannel.txPduTriggering
[constr_5314]	DltLogChannel txPduTriggering and rxPduTriggering shall be on the same network
[constr_5315]	FlexrayArTpConnections within the same FlexrayArTpChannel not allowed to have the same address information
[constr_5319]	TCP endpoint using TLS_SERVER role can only serve provided service instances
[constr_5320]	TCP endpoint using TLS_CLIENT role can only serve consumed service instances
[constr_5321]	Value range of Pdu.length
[constr_5322]	Value range of ISignalToIPduMapping.startPosition
[constr_5323]	Value range of ISignalToIPduMapping.updateIndicationBitPosition

Table E.63: Added Constraints in R21-11

E.14.5 Changed Constraints in R21-11

Number	Heading
[constr_1198]	TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping in case no DataTransformation is used
[constr_1199]	ISignals relating to systemSignals eligible for a TriggerToSignalMapping shall use update bit in case no DataTransformation is used

▽

△

Number	Heading
[constr_3141]	Only IPdus shall be part of a ContainerIPdu
[constr_3142]	Mandatory headerIdLongHeader for longHeader
[constr_3143]	Mandatory headerIdShortHeader for shortHeader
[constr_3151]	BufferProperties.headerLength settings for an E2E transformer used in combination with a SOME/IP transformer
[constr_3152]	BufferProperties.headerLength settings for any transformer used in combination with a COM Based transformer
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3159]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_04, PROFILE_04m PROFILE_44 and PROFILE_44m
[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m or dataIdMode different from lower12Bit

▽

△

Number	Heading
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m
[constr_3316]	Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07, PROFILE_08, PROFILE_07m and PROFILE_08m
[constr_3402]	Mandatory offset if noHeader is used
[constr_3404]	Usage of ContainedIPduProps.updateIndicationBitPosition
[constr_3454]	Unique headerIdLongHeader for acceptConfigured
[constr_3455]	Unique headerIdShortHeader for acceptConfigured
[constr_5191]	Consequence of attribute BinaryManifestProvideResource.item.category
[constr_5220]	Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_04m, PROFILE_07m, PROFILE_08m and PROFILE_44m

Table E.64: Changed Constraints in R21-11

E.14.6 Deleted Constraints in R21-11

Number	Heading
[constr_1207]	Existence of the attribute DataMapping.communicationDirection in the context of a SenderReceiverInterface or TriggerInterface
[constr_3035]	CanNm user data configuration in case NID/CBV are enabled
[constr_3041]	pncVectorOffset range
[constr_3042]	pncVectorLength range
[constr_3065]	Mapping of queued Triggers to SystemSignals is prohibited
[constr_3126]	headerLength shall be less or equal output buffer size
[constr_3154]	BufferProperties.bufferComputation setting for an E2E transformer when used together with a Com-based transformer
[constr_3241]	Usage of AssignFrameId.messageId
[constr_3242]	Usage of UnassignFrameId.messageId
[constr_3275]	PduTriggering containment in different PdurIPduGroups of the same EcuInstance is not allowed
[constr_3286]	ISignal.length shall be consistent to transformer configuration
[constr_3314]	BufferProperties.bufferComputation is mandatory
[constr_3315]	The value of V0 in BufferProperties.bufferComputation setting for a COM Based transformer
[constr_3547]	Mandatory reference to an EventHandler in case of serviceControl
[constr_5070]	headerIds of ProvidedServiceInstances shall be unique on a SocketAddress per communication direction

▽

△

Number	Heading
[constr_5098]	Usage of DltArgument.networkRepresentation
[constr_5099]	Standardized values of `DltMessage.messageTypeInfo`
[constr_5109]	Conditions for the explicit mapping of an `ISignal` to the `pncVector`
[constr_5171]	Existence of the attribute `DataMapping.communicationDirection` in `ClientServerToSignalMapping`

Table E.65: Deleted Constraints in R21-11

E.15 Constraint and Specification Item History of this document according to AUTOSAR R22-11

E.15.1 Added Specification Items in R22-11

Number	Heading
[TPS_SYST_02379]	Recommended configuration settings for E2E Profile 5 configuration setting
[TPS_SYST_02380]	Monitoring of the requested partial networks status in case of `allPartialNetworksActive` for provided service instance
[TPS_SYST_02381]	Monitoring of the requested partial networks status in case of `allPartialNetworksActive` for required service instance
[TPS_SYST_02382]	Monitoring of the `released` *partial networks* status in case of `allPartialNetworksActive` for required service instance
[TPS_SYST_02383]	Monitoring of the `released` *partial networks* status in case of `allPartialNetworksActive` for provided service instance
[TPS_SYST_02384]	Sending inner Ports may contain the superset of data provided on the outer delegation Port
[TPS_SYST_02385]	Receiving outer Ports may contain the superset of data delegated to the inner Ports
[TPS_SYST_02386]	MACsec configuration
[TPS_SYST_02387]	MAC Security Key Agreement Entity configuration
[TPS_SYST_02388]	Standardized values for the attribute `cipherSuite` of meta-class `MacSecCipherSuiteConfig`
[TPS_SYST_02389]	Semantics of `MacSecCipherSuiteConfig.cipherSuitePriority`
[TPS_SYST_03074]	CAN Controller support of CAN XL frames
[TPS_SYST_03075]	Communication over CAN XL
[TPS_SYST_03076]	Definition of CAN XL frame triggering attributes
[TPS_SYST_03077]	Managed channel in case of Ethernet tunneling through CAN XL
[TPS_SYST_03078]	RTE signal fan-out support for a stand-alone `SystemSignal` out of a `SystemSignalGroup`
[TPS_SYST_03079]	No RTE fan-in support for stand-alone `SystemSignal` and the same `SystemSignal` as part of a `SystemSignalGroup`

▽

△

Number	Heading
[TPS_SYST_03080]	`PhysicalChannel` involved in a particular PNC
[TPS_SYST_03081]	`EcuInstance` involved in a particular PNC
[TPS_SYST_03082]	Definition of top level PNC-Coordinator
[TPS_SYST_03083]	Creation of a *PNC paths tree*
[TPS_SYST_03084]	Acyclic PNC graph definition
[TPS_SYST_03085]	`Pdu` qualifies as dynamic length

Table E.66: Added Specification Items in R22-11

E.15.2 Changed Specification Items in R22-11

Number	Heading
[TPS_SYST_01109]	RTE signal fan-out support for a `SystemSignal`
[TPS_SYST_02185]	TTL for Find Service Entries
[TPS_SYST_02199]	Applicable `transferProperty` for `ISignalGroup` and all group signals have `transferProperty` not defined or `pending` defined
[TPS_SYST_02200]	Applicable `transferProperty` for `ISignalGroup` and group signals have `transferProperty` defined
[TPS_SYST_02228]	Transmission of events over UDP/TCP Port
[TPS_SYST_02251]	Data exchange not regulated by the Service Discovery protocol between two communication endpoints
[TPS_SYST_02309]	RTE signal fan-out support for a `SystemSignalGroup`
[TPS_SYST_02317]	References from `CpSoftwareCluster` to `CompositionSwComponentType` and `SwComponentPrototype`
[TPS_SYST_02320]	Kinds of `CpSoftwareClusterResource`s
[TPS_SYST_03025]	Control of service instance in case of `anyPartialNetworkActive` or `allPartialNetworksActive`
[TPS_SYST_03026]	Monitoring of the `requested` *partial networks* status in case of `anyPartialNetworkActive` for provided service instance
[TPS_SYST_03027]	Monitoring of the `requested` *partial networks* status in case of `anyPartialNetworkActive` for required service instance
[TPS_SYST_03050]	Usage of `ConsumedServiceInstance.blocklistedVersion`
[TPS_SYST_03056]	Monitoring of the `released` *partial networks* status in case of `anyPartialNetworkActive` for required service instance
[TPS_SYST_03057]	Monitoring of the `released` *partial networks* status in case of `anyPartialNetworkActive` for provided service instance

Table E.67: Changed Specification Items in R22-11

E.15.3 Deleted Specification Items in R22-11

Number	Heading
[TPS_SYST_01004]	Definition of AUTOSAR ECU
[TPS_SYST_01156]	Definition of ISignalTriggerings is allowed for ISignalGroups and for GroupSignals

Table E.68: Deleted Specification Items in R22-11

E.15.4 Added Constraints in R22-11

Number	Heading
[constr_3695]	canControllerXlAttributes and canControllerXlRequirements are mutually exclusive
[constr_3696]	Mandatory attributes of CanControllerXlConfiguration
[constr_3697]	Latest existence time of CanControllerXlConfiguration and CanControllerXlConfigurationRequirements
[constr_3698]	Value of errorSignalingEnabled
[constr_3699]	Existence of pwmL
[constr_3700]	Existence of pwmO
[constr_3701]	Existence of pwmS
[constr_3702]	Relevant attributes of EthernetCommunicationController for CAN XL
[constr_3703]	Reference to CanControllerXlConfiguration in case of category CAN_XL
[constr_3704]	Existence of CanXlFrameTriggeringProps
[constr_3705]	Allowed values for priorityId
[constr_3706]	Allowed values for sduType
[constr_3707]	Allowed values for vcid
[constr_3708]	No UDP network management in case of Ethernet tunneling through CAN XL
[constr_3713]	Allowed values for acceptanceField
[constr_3714]	Only one top level PNC-Coordinator per PNC
[constr_3716]	SecuredIPdu.dynamicRuntimeLengthHandling for dynamic length Pdus
[constr_3717]	SecuredIPdu.dynamicRuntimeLengthHandling for gateway operation with IPduMapping.pduMaxLength defined
[constr_3718]	Minimum length of SecuredIPdus
[constr_3726]	Upper multiplicity of aggregation in the role CouplingPort.macSecProps
[constr_5326]	Each local SocketAddress of an EcuInstance shall reference an EthernetCommunicationConnector in the role connector or multicastConnector
[constr_5327]	Existence of attribute CpSoftwareCluster.category
[constr_5328]	Ecu Extract shall only contain outerPort DataMappings
[constr_5329]	SynchronousServerCallPoints for cross cluster communication are not supported



△

Number	Heading
[constr_5330]	`ServiceInterface` elements shall belong to exactly one `ServiceInterface`
[constr_5331]	No IP multicast in case of TCP
[constr_5334]	Supported values for `CryptoServiceKey.length`
[constr_5335]	`CpSoftwareCluster.softwareClusterId` shall be unique in the scope of an `EcuInstance`
[constr_5336]	Existence of `CpSoftwareCluster.softwareClusterId`
[constr_5337]	All `CpSoftwareClusterToEcuInstanceMapping`s that are referencing the same `EcuInstance` shall define the same `machineId`
[constr_5344]	Applicable `transferProperty` for GroupSignal and `ISignalGroup`
[constr_5359]	`CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId`
[constr_5360]	Cross cluster communication involving `NvBlockSwComponentType` is not supported
[constr_5361]	MACsec configuration is allowed only on switch ports

Table E.69: Added Constraints in R22-11

E.15.5 Changed Constraints in R22-11

Number	Heading
[constr_3060]	Allowed Attributes for `networkRepresentationProps` and `physicalProps`
[constr_3268]	Service Discovery `StaticSocketConnection` aggregation by a `SocketAddress`
[constr_3325]	`SecureCommunicationFreshnessProps`, `SecureCommunicationAuthenticationProps` and `CryptoServicePrimitive` attribute value settings for standardized AUTOSAR security profiles
[constr_3331]	Standardized values for the attribute `category` of meta-class `EthernetCommunicationConnector`
[constr_3332]	Standardized values for the attribute `category` of meta-class `EthernetCommunicationController`
[constr_3516]	limitation of `Frame.frameLength` for CAN L-PDUs
[constr_3545]	Mandatory reference to a *Pnc* in case of `anyPartialNetworkActive` or `allPartialNetworksActive`
[constr_3559]	`ConsumedServiceInstance.blocklistedVersion` is restricted to the usage of `minorVersion`
[constr_5168]	pncGatewayType passive and connected ECUs

Table E.70: Changed Constraints in R22-11

E.15.6 Deleted Constraints in R22-11

Number	Heading
[constr_3459]	Applicable `transferProperty` for group signal

Table E.71: Deleted Constraints in R22-11

E.15.7 Added Advisories in R22-11

Number	Heading
[advisory_02004]	Check for reachable PNC nodes
[advisory_03000]	`initValue` defined in the context of `ISignal` that references a `SystemSignal` with a `CompuMethod` of `category` `TEXTTABLE` or `BITFIELD_TEXTTABLE`
[advisory_03001]	`invalidValue` defined in the context of `ISignal` that references a `SystemSignal` with a `CompuMethod` of `category` `TEXTTABLE` or `BITFIELD_TEXTTABLE`
[advisory_03002]	`timeoutSubstitutionValue` defined in the context of `ISignal` that references a `SystemSignal` with a `CompuMethod` of `category` `TEXTTABLE` or `BITFIELD_TEXTTABLE`

Table E.72: Added Advisories in R22-11

E.15.8 Changed Advisories in R22-11

none

E.15.9 Deleted Advisories in R22-11

none

E.16 Constraint and Specification Item History of this document according to AUTOSAR R23-11

E.16.1 Added Specification Items in R23-11

Number	Heading
[TPS_SYST_02390]	Relevance of LdCom spec items for the sender and for the receiver side.
[TPS_SYST_02391]	Mapping of MOST FIBEX elements to AUTOSAR elements
[TPS_SYST_02393]	Semantics of `CpSoftwareClusterToApplicationPartitionMapping`
[TPS_SYST_02394]	Aggregation possibilities of `CpSoftwareClusterToApplicationPartitionMapping`
[TPS_SYST_02395]	`CpSoftwareClusterToApplicationPartitionMapping` aggregated by `SystemMapping` supersedes `CpSoftwareClusterToApplicationPartitionMapping` aggregated by `CpSoftwareClusterMappingSet`
[TPS_SYST_02396]	Summary of supported configuration options between ApplicationEndpoints
[TPS_SYST_02397]	Assignment of `CpSoftwareClusterCommunicationResource`s to a `SystemSignal`
[TPS_SYST_02398]	Assignment of `CpSoftwareClusterCommunicationResource`s to a `SystemSignalGroup`
[TPS_SYST_02399]	`mirroringProtocol` for CAN destination channel with CAN 2.0 protocol
[TPS_SYST_02400]	`mirroringProtocol` for Ethernet destination channel
[TPS_SYST_02401]	ACL check definition for a `ProvidedServiceInstance`
[TPS_SYST_02402]	ACL check definition for a `ConsumedServiceInstance`
[TPS_SYST_03086]	DdsDomain
[TPS_SYST_03087]	DdsTopic
[TPS_SYST_03088]	DdsPartition
[TPS_SYST_03089]	DdsQoSProfile
[TPS_SYST_03090]	DDS Methods Request/Reply Topic
[TPS_SYST_03091]	DDS Fields Request/Reply Topic
[TPS_SYST_03093]	DDS Operation
[TPS_SYST_03094]	DdsService QoS Profile
[TPS_SYST_03095]	DDS Event QoSProfile
[TPS_SYST_03096]	Firewall on a Host ECU
[TPS_SYST_03097]	Firewall on an Ethernet Switch
[TPS_SYST_03098]	`IEEE1722Tp` Stream Id definition
[TPS_SYST_03099]	`IEEE1722Tp` Stream Version definition
[TPS_SYST_03100]	`IEEE1722Tp` destination MAC address definition
[TPS_SYST_03101]	Collection of several control messages in one `IEEE1722TpAcfConnection`

▽

△

Number	Heading
[TPS_SYST_03102]	Values of IEEE1722TpAcfBus.busId per bus system
[TPS_SYST_03103]	Abbreviated CAN message transport
[TPS_SYST_03104]	Triggering the immediate transmission of an IEEE1722Tp ACF message
[TPS_SYST_03105]	Definition of LIN message transport in IEEE1722Tp ACF stream
[TPS_SYST_03106]	Definition of LIN Id in IEEE1722Tp ACF stream
[TPS_SYST_03107]	Definition of CAN message transport in IEEE1722Tp ACF stream
[TPS_SYST_03108]	Definition of CAN Id in IEEE1722Tp ACF stream
[TPS_SYST_03109]	Transmission of an IEEE1722TpConnection
[TPS_SYST_03110]	Reception of an IEEE1722TpConnection
[TPS_SYST_03111]	Ethernet switch CouplingPortAsynchronousTrafficShaper
[TPS_SYST_03112]	ingressPort definition for SwitchStreamIdentification
[TPS_SYST_03113]	egressPort definition for SwitchStreamIdentification
[TPS_SYST_03114]	Actions for SwitchStreamIdentification
[TPS_SYST_03115]	SwitchStreamFilterEntry for SwitchStreamIdentification

Table E.73: Added Specification Items in R23-11

E.16.2 Changed Specification Items in R23-11

Number	Heading
[TPS_SYST_01065]	Mapping onto the ComSignalType enumeration
[TPS_SYST_01157]	Allowed usage of attributes for ISignals, ISignalGroups and GroupSignals
[TPS_SYST_02070]	Recommended configuration settings for E2E Profile 4 configuration setting A
[TPS_SYST_02103]	Semantics of GlobalTimeDomain
[TPS_SYST_02134]	Recommended configuration settings for E2E Profile 7 configuration setting A
[TPS_SYST_02177]	TTL for Offer Service Entries
[TPS_SYST_02344]	SW_CLUSTER_SYSTEM_DESCRIPTION content
[TPS_SYST_03009]	Ethernet switch CouplingPortCreditBasedShaper
[TPS_SYST_05027]	Standardized values of CryptoServicePrimitive.algorithmMode

Table E.74: Changed Specification Items in R23-11

E.16.3 Deleted Specification Items in R23-11

Number	Heading
[TPS_SYST_02011]	initValues of receivers that are mapped to the same Ecu
[TPS_SYST_02099]	Relation between ContainerIPdu and contained PduTriggerings on sender side

Table E.75: Deleted Specification Items in R23-11

E.16.4 Added Constraints in R23-11

Number	Heading
[constr_3735]	Existence of DdsCpServiceInstance.ddsServiceQosProfile
[constr_3736]	ISignal that has dataTypePolicy set to ddsSignal shall be referenced by a DdsCpISignalToDdsTopicMapping
[constr_3737]	ISignal referenced from DdsCpISignalToDdsTopicMapping
[constr_3738]	ISignal that has dataTypePolicy set to ddsSignal or to ddsService shall not reference a DataTransformation
[constr_3739]	Value of ISignal.dataTypePolicy for all ISignals associated with a DdsCpServiceInstance
[constr_3740]	Existence of DdsCpServiceInstanceEvent.ddsEventTopic
[constr_3741]	Exclusive setting of channelSynchronousWakeup or pncSynchronousWakeup
[constr_3742]	Value for createEcuWakeupSource in the conntext of a CommunicationCluster
[constr_3743]	Allowed values for IEEE1722TpConnection.uniqueStreamId
[constr_3744]	Allowed values for IEEE1722TpConnection.version
[constr_3745]	category of GeneralPurposePdu referenced in the role IEEE1722TpConnection.pdu
[constr_3746]	category of GeneralPurposePdu referenced in the role IEEE1722TpAvConnection.sdu
[constr_3747]	Existence of attribute IEEE1722TpConnection.uniqueStreamId
[constr_3748]	Existence of attribute IEEE1722TpConnection.macAddressStreamId
[constr_3749]	Existence of attribute IEEE1722TpConnection.version
[constr_3750]	Existence of attribute IEEE1722TpConnection.pdu
[constr_3751]	Allowed values for IEEE1722TpAcfBus.busId
[constr_3752]	Existence of attribute IEEE1722TpAcfBus.busId
[constr_3753]	Existence of attribute IEEE1722TpAcfLinPart.linIdentifier
[constr_3754]	Existence of attribute IEEE1722TpAcfCan.messageType
[constr_3755]	Consistend aggregation of IEEE1722TpAcfCanPart
[constr_3756]	Consistend aggregation of IEEE1722TpAcfLinPart
[constr_3757]	Allowed values for IEEE1722TpAcfLinPart.linIdentifier
[constr_3758]	Allowed values for IEEE1722TpAcfCanPart.canIdentifier



△

Number	Heading
[constr_3759]	Existence of attribute IEEE1722TpAcfCanPart.canIdentifier for IEEE1722Tp ACF stream transmission
[constr_3760]	Existence of attribute IEEE1722TpAcfCanPart.canIdentifierRange or canIdentifierMask for IEEE1722Tp ACF stream reception
[constr_3761]	Identical EthernetPhysicalChannel owning PduTriggerings referenced by IEEE1722TpConnection.pdu and IEEE1722TpAvConnection.sdu
[constr_3762]	Usage of CouplingElementSwitchDetails only on an Ethernet switch
[constr_5362]	Relation between ContainerIPdu and contained PduTriggerings on sender side
[constr_5369]	Consistency between SwcToApplicationPartitionMapping and CpSoftwareClusterToApplicationPartitionMapping
[constr_5370]	Restriction for SystemSignalToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignal
[constr_5371]	Restriction for SystemSignalGroupToCommunicationResourceMapping in case a DataMapping is defined for the mapped SystemSignalGroup
[constr_5374]	IPdu shall only be referenced once from a FlexrayTpConnection in the role directTpSdu or reversedTpSdu on a FlexrayCluster
[constr_5375]	IPdu shall only be referenced once from a FlexrayArTpConnection in the role directTpSdu or reversedTpSdu on a FlexrayCluster
[constr_5376]	IPdu shall only be referenced once from a CanTpConnection in the role tpSdu on a CanCluster
[constr_5377]	IPdu shall only be referenced once from a LinTpConnection in the role linTpNSdu on a LinCluster
[constr_5378]	PduTriggering shall only be referenced once from a SomeipTpConnection in the role tpSdu
[constr_5379]	IPdu shall only be referenced once from a J1939TpPg in the role sdu on a J1939Cluster
[constr_5380]	Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to to a value greater than 1 in the context of an EcuInstance
[constr_5382]	Relation between the value of attributes offerCyclicDelay and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig
[constr_5383]	Relation between the value of attributes initialRepetitionsBaseDelay and initialRepetitionsMax and serviceOfferTimeToLive in the context of a SomeipSdServerServiceInstanceConfig
[constr_5384]	Existence of BusMirrorChannelMapping.mirroringProtocol
[constr_5385]	Reception of UserData inside of a NmPdu by Applications is not supported
[constr_5389]	Dependency between globalTimeTxPeriod and globalTimePortRole
[constr_5390]	The globalTimePortRole shall not be configured to timeSlave several times in the same GlobalTimeDomain
[constr_5391]	ConsumedServiceInstance.allowedServiceProvider reference restriction
[constr_5393]	Existence of clientId
[constr_5394]	Existence of clientServerOperation

▽

△

Number	Heading
[constr_5395]	Existence of physicalChannel
[constr_5396]	Existence of ClientIdRange.lowerLimit
[constr_5397]	Existence of ClientIdRange.upperLimit
[constr_5398]	Existence of CommunicationConnector.commController
[constr_5399]	Existence of ecu
[constr_5400]	Existence of ecuInstance
[constr_5401]	Existence of communicationController
[constr_5402]	Existence of hwCommunicationController
[constr_5403]	Existence of communicationConnector
[constr_5404]	Existence of hwCommunicationPort
[constr_5405]	Existence of actionPointOffset
[constr_5406]	Existence of bit
[constr_5407]	Existence of casRxLowMax
[constr_5408]	Existence of coldStartAttempts
[constr_5409]	Existence of cycle
[constr_5410]	Existence of cycleCountMax
[constr_5412]	Existence of dynamicSlotIdlePhase
[constr_5414]	Existence of listenNoise
[constr_5415]	Existence of macroPerCycle
[constr_5416]	Existence of macrotickDuration
[constr_5417]	Existence of maxWithoutClockCorrectionFatal
[constr_5418]	Existence of maxWithoutClockCorrectionPassive
[constr_5419]	Existence of minislotActionPointOffset
[constr_5420]	Existence of minislotDuration
[constr_5421]	Existence of networkIdleTime
[constr_5422]	Existence of networkManagementVectorLength
[constr_5423]	Existence of numberOfMinislots
[constr_5424]	Existence of numberOfStaticSlots
[constr_5425]	Existence of offsetCorrectionStart
[constr_5426]	Existence of payloadLengthStatic
[constr_5428]	Existence of staticSlotDuration
[constr_5429]	Existence of symbolWindow
[constr_5431]	Existence of syncFrameIdCountMax
[constr_5432]	Existence of transmissionStartSequenceDuration
[constr_5433]	Existence of wakeupRxIdle
[constr_5434]	Existence of wakeupRxLow
[constr_5435]	Existence of wakeupRxWindow
[constr_5436]	Existence of wakeupTxActive
[constr_5437]	Existence of wakeupTxIdle

▽

△

Number	Heading
[constr_5438]	Existence of sampleClockPeriod
[constr_5439]	Existence of admitWithoutMessageId
[constr_5440]	Existence of baseCycle
[constr_5441]	Existence of cycleRepetition
[constr_5442]	Existence of fifoDepth
[constr_5443]	Existence of msgIdMask
[constr_5444]	Existence of msgIdMatch
[constr_5445]	Existence of fifoRange
[constr_5446]	Existence of rangeMax
[constr_5447]	Existence of rangeMin
[constr_5448]	Existence of channelName
[constr_5449]	LinCommunicationController.protocolVersion shall exist
[constr_5450]	Existence of index
[constr_5451]	Existence of LinOrderedConfigurableFrame.frame reference
[constr_5452]	Existence of LinConfigurableFrame.frame reference
[constr_5453]	Existence of macMulticastAddress
[constr_5454]	Existence of vlanIdentifier
[constr_5455]	Existence of couplingType
[constr_5456]	Existence of communicationCluster
[constr_5457]	Existence of defaultPriority
[constr_5458]	Existence of vlan
[constr_5459]	Existence of dataLength
[constr_5460]	Existence of policyAction
[constr_5461]	Existence of timeInterval
[constr_5462]	Existence of ingressPriority
[constr_5463]	Existence of regeneratedPriority
[constr_5464]	Existence of trafficClass
[constr_5465]	Existence of softwareComposition
[constr_5466]	Existence of SenderReceiverToSignalMapping.dataElement
[constr_5467]	Existence of SenderReceiverToSignalMapping.systemSignal
[constr_5468]	Existence of SenderReceiverToSignalGroupMapping.dataElement
[constr_5469]	Existence of SenderReceiverToSignalGroupMapping.signalGroup
[constr_5470]	Existence of SenderReceiverToSignalGroupMapping.typeMapping
[constr_5471]	Existence of SenderRecArrayElementMapping.indexedArrayElement
[constr_5472]	Existence of IndexedArrayElement.index
[constr_5473]	Existence of ClientServerToSignalMapping.callSignal
[constr_5474]	Existence of ClientServerToSignalMapping.clientServerOperation
[constr_5475]	Existence of SenderReceiverCompositeElementToSignalMapping.systemSignal
[constr_5476]	Existence of SenderReceiverCompositeElementToSignalMapping.typeMapping

▽

△

Number	Heading
[constr_5477]	Existence of TriggerToSignalMapping.systemSignal
[constr_5478]	Existence of TriggerToSignalMapping.trigger
[constr_5479]	Existence of PncMapping.pncIdentifier
[constr_5480]	Existence of EcuResourceEstimation.ecuInstance
[constr_5481]	Existence of SwcToSwcSignal.dataElement
[constr_5482]	Existence of SwcToSwcOperationArguments.direction
[constr_5483]	Existence of SwcToSwcOperationArguments.operation
[constr_5484]	Existence of ForbiddenSignalPath.physicalChannel
[constr_5485]	Existence of PermissibleSignalPath.physicalChannel
[constr_5486]	Existence of SwcToEcuMapping.component
[constr_5487]	Existence of SwcToEcuMapping.ecuInstance
[constr_5488]	Existence of SwcToImplMapping.component
[constr_5489]	Existence of SwcToImplMapping.componentImplementation
[constr_5491]	Existence of ComponentClustering.clusteredComponent
[constr_5492]	Existence of ComponentSeparation.separatedComponent
[constr_5493]	Existence of J1939ControllerApplication.functionId
[constr_5494]	Existence of BusMirrorChannel.busMirrorNetworkId
[constr_5495]	Existence of BusMirrorCanIdRangeMapping.destinationBaseId
[constr_5496]	Existence of BusMirrorCanIdRangeMapping.sourceCanIdCode
[constr_5497]	Existence of BusMirrorCanIdRangeMapping.sourceCanIdMask
[constr_5498]	Existence of BusMirrorCanIdToCanIdMapping.remappedCanId
[constr_5499]	Existence of BusMirrorLinPidToCanIdMapping.remappedCanId
[constr_9100]	Existence of CanFrameTriggering.canAddressingMode
[constr_9101]	Existence of RxIdentifierRange.lowerCanId
[constr_9102]	Existence of RxIdentifierRange.upperCanId
[constr_9103]	Existence of communicationDirection
[constr_9105]	Existence of DoIpTpConfig.tpConnection
[constr_9106]	Existence of DoIpTpConnection.doIpSourceAddress
[constr_9107]	Existence of DoIpTpConnection.doIpTargetAddress
[constr_9108]	Existence of DoIpTpConnection.tpSdu
[constr_9109]	Existence of IPv6ExtHeaderFilterList.allowedIPv6ExtHeader
[constr_9110]	Existence of TcpOptionFilterList.allowedTcpOption
[constr_9111]	Existence of ApplicationEndpoint.networkEndpoint
[constr_9112]	Existence of GenericTp.tpTechnology
[constr_9113]	Existence of UdpTp.udpTpPort
[constr_9114]	Existence of TcpTp.tcpTpPort
[constr_9115]	Existence of RtpTp.ssrc
[constr_9116]	Existence of RtpTp.tcpUdpConfig
[constr_9119]	Existence of Ieee1722Tp.streamIdentifier

▽

△

Number	Heading
[constr_9120]	Existence of HttpTp.protocolVersion
[constr_9121]	Existence of HttpTp.tcpTpConfig
[constr_9122]	Existence of NetworkEndpoint.networkEndpointAddress
[constr_9123]	Existence of MacMulticastConfiguration.macMulticastGroup
[constr_9124]	Existence of FlexrayFrameTriggering.allowDynamicLSduLength
[constr_9125]	Existence of FlexrayFrameTriggering.payloadPreambleIndicator
[constr_9126]	Existence of FlexrayAbsolutelyScheduledTiming.slotID
[constr_9127]	Existence of FlexrayAbsolutelyScheduledTiming.communicationCycle
[constr_9128]	Existence of CycleCounter.CycleCounter
[constr_9129]	Existence of CycleRepetition.BaseCycle
[constr_9130]	Existence of CycleRepetition.CycleRepetition
[constr_9131]	Existence of FrameTriggering.frame
[constr_9132]	Existence of LinSporadicFrame.substitutedFrame
[constr_9133]	Existence of LinEventTriggeredFrame.linUnconditionalFrame
[constr_9134]	Existence of ScheduleTableEntry.delay
[constr_9135]	Existence of ScheduleTableEntry.positionInTable
[constr_9136]	Existence of ApplicationEntry.frameTriggering
[constr_9137]	Existence of AssignFrameId.assignedFrameTriggering
[constr_9138]	Existence of UnassignFrameId.unassignedFrameTriggering
[constr_9139]	Existence of AssignFrameIdRange.startIndex
[constr_9140]	Existence of FramePid.index
[constr_9141]	Existence of FramePid.pid
[constr_9142]	Existence of AssignNad.newNad
[constr_9143]	Existence of ConditionalChangeNad.byte
[constr_9144]	Existence of ConditionalChangeNad.id
[constr_9145]	Existence of ConditionalChangeNad.invert
[constr_9146]	Existence of ConditionalChangeNad.mask
[constr_9147]	Existence of ConditionalChangeNad.newNad
[constr_9148]	Existence of DataDumpEntry.byteValue
[constr_9149]	Existence of FreeFormat.byteValue
[constr_9150]	Existence of NmEcu.ecuInstance
[constr_9151]	Existence of nmDataCycle
[constr_9152]	Existence of nmRemoteSleepIndicationTime
[constr_9153]	Existence of nmRepeatMessageTime
[constr_9154]	Existence of nmRepetitionCycle
[constr_9155]	Existence of nmVotingCycle
[constr_9156]	Existence of nmScheduleVariant
[constr_9157]	Existence of nmBusloadReductionActive
[constr_9158]	Existence of nmImmediateNmTransmissions

▽

△

Number	Heading
[constr_9159]	Existence of nmMessageTimeoutTime
[constr_9160]	Existence of nmMsgCycleTime
[constr_9161]	Existence of nmNetworkTimeout
[constr_9162]	Existence of nmRemoteSleepIndicationTime
[constr_9163]	Existence of nmRepeatMessageTime
[constr_9164]	Existence of nmWaitBusSleepTime
[constr_9165]	Existence of nmBusloadReductionEnabled
[constr_9166]	Existence of nmImmediateRestartEnabled
[constr_9167]	Existence of J1939NodeName.arbitraryAddressCapable
[constr_9168]	Existence of J1939NodeName.ecuInstance
[constr_9169]	Existence of J1939NodeName.function
[constr_9170]	Existence of J1939NodeName.functionInstance
[constr_9171]	Existence of J1939NodeName.identitiyNumber
[constr_9172]	Existence of J1939NodeName.industryGroup
[constr_9173]	Existence of J1939NodeName.manufacturerCode
[constr_9174]	Existence of J1939NodeName.vehicleSystem
[constr_9175]	Existence of J1939NodeName.vehicleSystemInstance
[constr_9176]	Existence of StaticPart.iPdu
[constr_9177]	Existence of DynamicPartAlternative.initialDynamicPart
[constr_9178]	Existence of DynamicPartAlternative.initialDynamicPart
[constr_9179]	Existence of DynamicPartAlternative.iPdu
[constr_9180]	Existence of DynamicPartAlternative.selectorFieldCode
[constr_9181]	Existence of MultiplexedPart.segmentPosition
[constr_9182]	Existence of SegmentPosition.segmentByteOrder
[constr_9183]	Existence of SegmentPosition.segmentLength
[constr_9184]	Existence of SegmentPosition.segmentPosition
[constr_9185]	Existence of TransmissionModeCondition.dataFilter
[constr_9186]	Existence of TransmissionModeCondition.iSignalInIPdu
[constr_9187]	Existence of ModeDrivenTransmissionModeCondition.modeDeclaration
[constr_9188]	Existence of ModeDrivenTransmissionModeCondition.timePeriod
[constr_9189]	Existence of EventControlledTiming.numberOfRepetitions
[constr_9190]	Existence of TimeRangeType.value
[constr_9191]	Existence of RelativeTolerance.relative
[constr_9192]	Existence of AbsoluteTolerance.absolute
[constr_9193]	Existence of TriggerIPduSendCondition.modeDeclaration
[constr_9194]	Existence of DcmIPdu.diagPduType
[constr_9195]	Existence of PduToFrameMapping.packingByteOrder
[constr_9196]	Existence of PduToFrameMapping.startPosition
[constr_9197]	Existence of PduToFrameMapping.pdu

▽

△

Number	Heading
[constr_9198]	Existence of PduTriggering.iPdu
[constr_9199]	Existence of ISignalIPduGroup.communicationDirection
[constr_9200]	Existence of ContainerIPdu.headerType
[constr_9201]	Existence of ContainerIPdu.rxAcceptContainedIPdu
[constr_9202]	Existence of ContainedIPduProps.collectionSemantics
[constr_9203]	Existence of SecuredIPdu.payload
[constr_9204]	Existence of SecuredIPdu.secureCommunicationProps
[constr_9205]	Existence of SecureCommunicationProps.dataId
[constr_9206]	Existence of CryptoServiceKey.length
[constr_9207]	Existence of EndToEndProtectionISignalIPdu.iSignalIPdu
[constr_9208]	Existence of EndToEndProtectionISignalIPdu.iSignalGroup
[constr_9209]	Existence of EndToEndProtectionISignalIPdu.dataOffset
[constr_9210]	Existence of InitialSdDelayConfig.initialDelayMaxValue aggregated by SomeipSdClientServiceInstanceConfig
[constr_9211]	Existence of InitialSdDelayConfig.initialDelayMinValue aggregated by SomeipSdClientServiceInstanceConfig
[constr_9212]	Existence of SomeipSdClientEventGroupTimingConfig.timeToLive
[constr_9213]	Existence of RequestResponseDelay.minValue aggregated by SomeipSdClientEventGroupTimingConfig
[constr_9214]	Existence of RequestResponseDelay.maxValue aggregated by SomeipSdClientEventGroupTimingConfig
[constr_9215]	Existence of InitialSdDelayConfig.initialDelayMaxValue aggregated by SomeipSdServerServiceInstanceConfig
[constr_9216]	Existence of InitialSdDelayConfig.initialDelayMinValue aggregated by SomeipSdServerServiceInstanceConfig
[constr_9217]	Existence of SomeipSdServerServiceInstanceConfig.serviceOfferTimeToLive
[constr_9218]	Existence of RequestResponseDelay.minValue aggregated by SomeipSdServerServiceInstanceConfig
[constr_9219]	Existence of RequestResponseDelay.maxValue aggregated by SomeipSdServerServiceInstanceConfig
[constr_9220]	Existence of RequestResponseDelay.minValue aggregated by SomeipSdServerEventGroupTimingConfig
[constr_9221]	Existence of RequestResponseDelay.maxValue aggregated by SomeipSdServerEventGroupTimingConfig
[constr_9222]	Existence of ISignal.dataTypePolicy
[constr_9223]	Existence of ISignal.length
[constr_9224]	Existence of ISignal.systemSignal
[constr_9225]	Existence of ISignalGroup.systemSignalGroup
[constr_9226]	Existence of TpConfig.communicationCluster
[constr_9227]	Existence of TpAddress.tpAddress
[constr_9228]	Existence of FlexrayTpConfig.pduPool

▽

△

Number	Heading
[constr_9229]	Existence of FlexrayTpConfig.tpAddress
[constr_9230]	Existence of FlexrayTpConfig.tpEcu
[constr_9231]	Existence of FlexrayTpConnection.directTpSdu
[constr_9233]	Existence of FlexrayTpConnection.receiver
[constr_9234]	Existence of FlexrayTpConnection.tpConnectionControl
[constr_9235]	Existence of FlexrayTpConnection.transmitter
[constr_9236]	Existence of FlexrayTpEcu.ecuInstance
[constr_9237]	Existence of FlexrayTpEcu.fullDuplexEnabled
[constr_9238]	Existence of FlexrayArTpChannel.ackType
[constr_9239]	Existence of FlexrayArTpChannel.extendedAddressing
[constr_9240]	Existence of FlexrayArTpChannel.maximumMessageLength
[constr_9241]	Existence of FlexrayArTpChannel.minimumSeparationTime
[constr_9242]	Existence of FlexrayArTpChannel.multicastSegmentation
[constr_9243]	Existence of FlexrayArTpChannel.tpConnection
[constr_9244]	Existence of FlexrayArTpConnection.directTpSdu
[constr_9245]	Existence of FlexrayArTpConnection.source
[constr_9246]	Existence of FlexrayArTpConnection.target
[constr_9247]	Existence of CanTpConfig.tpAddress
[constr_9248]	Existence of CanTpConfig.tpChannel
[constr_9249]	Existence of CanTpConfig.tpConnection
[constr_9250]	Existence of CanTpConfig.tpEcu
[constr_9251]	Existence of CanTpConfig.tpNode
[constr_9252]	Existence of CanTpConnection.addressingFormat
[constr_9253]	Existence of CanTpConnection.canTpChannel
[constr_9254]	Existence of CanTpConnection.dataPdu
[constr_9255]	Existence of CanTpConnection.paddingActivation
[constr_9256]	Existence of CanTpConnection.tpSdu
[constr_9257]	Existence of CanTpAddress.tpAddress
[constr_9258]	Existence of CanTpEcu.ecuInstance
[constr_9259]	Existence of LinTpConfig.tpAddress
[constr_9260]	Existence of LinTpConnection.dataPdu
[constr_9261]	Existence of LinTpConnection.linTpNSdu
[constr_9262]	Existence of LinTpConnection.receiver
[constr_9263]	Existence of LinTpConnection.transmitter
[constr_9264]	Existence of J1939TpConfig.tpAddress
[constr_9265]	Existence of J1939TpConfig.tpConnection
[constr_9266]	Existence of J1939TpConfig.tpNode
[constr_9267]	Existence of J1939TpConnection.broadcast
[constr_9268]	Existence of J1939TpConnection.dataPdu

▽

△

Number	Heading
[constr_9269]	Existence of J1939TpConnection.flowControlPdu
[constr_9270]	Existence of TlsCryptoCipherSuite.version
[constr_9271]	Existence of TlsPskIdentity.pskIdentity
[constr_9272]	Existence of TlsPskIdentity.preSharedKey
[constr_9273]	Existence of DataTransformation.executeDespiteDataUnavailability
[constr_9274]	Existence of DataTransformation.transformerChain
[constr_9275]	Existence of TransformationTechnology.bufferProperties
[constr_9276]	Existence of TransformationTechnology.protocol
[constr_9277]	Existence of TransformationTechnology.transformerClass
[constr_9278]	Existence of TransformationTechnology.version
[constr_9279]	Existence of BufferProperties.headerLength
[constr_9280]	Existence of BufferProperties.inPlace
[constr_9281]	Existence of TransformationISignalProps.dataPdu
[constr_9282]	Existence of SOMEIPTransformationDescription.alignment
[constr_9283]	Existence of SOMEIPTransformationDescription.byteOrder
[constr_9284]	Existence of SOMEIPTransformationDescription.interfaceVersion
[constr_9285]	Existence of DataPrototypeInSenderReceiverInterfaceInstance-Ref.targetDataPrototypeInSr
[constr_9286]	Existence of DataPrototypeInClientServerInterfaceInstance-Ref.targetDataPrototypeInCs
[constr_9287]	Existence of EndToEndTransformationDescription.profileName
[constr_9288]	Existence of TlvDataIdDefinition.id
[constr_9289]	Existence of FrameMapping.sourceFrame
[constr_9290]	Existence of FrameMapping.targetFrame
[constr_9291]	Existence of Gateway.ecu
[constr_9292]	Existence of IPduMapping.sourceIPdu
[constr_9293]	Existence of IPduMapping.targetIPdu
[constr_9294]	Existence of TargetIPduRef.targetIPdu
[constr_9295]	Existence of PduMappingDefaultValue.defaultValueElement
[constr_9296]	Existence of DefaultValueElement.elementPosition
[constr_9297]	Existence of DefaultValueElement.elementByteValue
[constr_9298]	Existence of ISignalMapping.sourceSignal
[constr_9299]	Existence of ISignalMapping.targetSignal
[constr_9300]	Existence of FlatMap.instance
[constr_9301]	Existence of AliasNameAssignment.shortLabel
[constr_9302]	Existence of GlobalTimeDomain.domainId
[constr_9303]	Existence of GlobalTimeMaster.communicationConnector
[constr_9304]	Existence of GlobalTimeMaster.isSystemWideGlobalTimeMaster
[constr_9305]	Existence of GlobalTimeMaster.syncPeriod

▽

△

Number	Heading
[constr_9306]	Existence of GlobalTimeSlave.communicationConnector
[constr_9307]	Existence of GlobalTimeGateway.master
[constr_9308]	Existence of GlobalTimeGateway.slave
[constr_9309]	Existence of GlobalTimeCanMaster.syncConfirmationTimeout
[constr_9310]	Existence of GlobalTimeCanSlave.crcValidated
[constr_9311]	Existence of EthGlobalTimeDomainProps.messageCompliance
[constr_9312]	Existence of EthGlobalTimeManagedCouplingPort.pdelayResponseEnabled
[constr_9313]	Existence of GlobalTimeCouplingPortProps.propagationDelay
[constr_9314]	Existence of GlobalTimeFrMaster.crcSecured
[constr_9315]	Existence of GlobalTimeFrSlave.crcValidated

Table E.76: Added Constraints in R23-11

E.16.5 Changed Constraints in R23-11

Number	Heading
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3111]	returnSignal in ClientServerToSignalMapping is mandatory
[constr_3266]	TransformationTechnology.hasInternalState setting for a SOME/IP Transformer
[constr_3601]	Mandatory attributes of EthernetWakeupSleepOnDatalineConfig
[constr_5144]	Value range of Ipv6NdpProps.tcpIpNdpDelayFirstProbeTimeValue
[constr_5252]	LinSlaveConfig.protocolVersion shall exist
[constr_5307]	Existence of DltLogChannel.logChannelId
[constr_5308]	Existence of DltLogChannel.nonVerboseMode
[constr_5309]	Existence of DltConfig.sessionIdSupport
[constr_5310]	Existence of DltConfig.timestampSupport
[constr_5311]	Existence of DltLogChannel.logTraceDefaultLogThreshold
[constr_5312]	Existence of DltLogChannel.defaultTraceState
[constr_5313]	Existence of DltLogChannel.txPduTriggering

Table E.77: Changed Constraints in R23-11

E.16.6 Deleted Constraints in R23-11

Number	Heading
[constr_3216]	Usage of SOMEIPTransformationISignalProps.sessionHandlingSR
[constr_3251]	Value of GlobalTimeDomain.domainId in globalTimeSubDomain chains
[constr_3520]	Offset time domain shall be based on a synchronized time domain

Table E.78: Deleted Constraints in R23-11

E.16.7 Added Advisories in R23-11

none

E.16.8 Changed Advisories in R23-11

none

E.16.9 Deleted Advisories in R23-11

none

E.17 Constraint and Specification Item History of this document according to AUTOSAR R24-11

E.17.1 Added Specification Items in R24-11

Number	Heading
[TPS_SYST_02403]	Usage of an identical GlobalTimeDomain.domainId in the GlobalTimeMaster role on one PhysicalChannel
[TPS_SYST_02404]	Relevance of BusMirrorChannelMapping.transmissionDeadline
[TPS_SYST_02405]	TransformationISignalPropsIdent.shortName usage
[TPS_SYST_02406]	DataPrototypeTransformationPropsIdent.shortName usage
[TPS_SYST_02407]	MetaData support is required if J1939TpConnection.acceptVariableSA is used
[TPS_SYST_02408]	MetaData support is required if J1939TpConnection.acceptVariableDA is used
[TPS_SYST_02409]	MetaData support is required if J1939TpConnection.useVariableSA is used
[TPS_SYST_02410]	MetaData support is required if J1939TpConnection.useVariableDA is used

▽

△

Number	Heading
[TPS_SYST_02411]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are transmitted by an EcuInstance
[TPS_SYST_02412]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are received via IP unicast by an EcuInstance
[TPS_SYST_02413]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD that are received via IP mutlicast by an EcuInstance
[TPS_SYST_02414]	StaticSocketConnection for IP unicast/IP multicast transmission and IP unicast reception of GeneralPurposePdus with category SD
[TPS_SYST_02415]	StaticSocketConnection for IP multicast reception of GeneralPurposePdus with category SD
[TPS_SYST_02416]	Semantics of J1939ProtectedIPdu
[TPS_SYST_02417]	Content of ISignalGroup located in J1939ProtectedIPdu.payload
[TPS_SYST_02418]	Definition of wildcard EthernetVlanTranslationTable
[TPS_SYST_02419]	Semantics of E2EProfileCompatibilityProps.combinedNoDataInitCount
[TPS_SYST_03116]	Definition of Time-Synchronous Control Format
[TPS_SYST_03117]	Only one PduTriggering for the payload of a ContainerIPdu or SecuredIPdu in case of fan-out
[TPS_SYST_03118]	Only one ISignalTriggering in case of PDU fan-out
[TPS_SYST_03119]	Definition of MAC address learning mode for an Ethernet switch
[TPS_SYST_03120]	Definition of MAC multicast addresses in case of SwitchMacAddressLearningEnum.sharedVlanLearning
[TPS_SYST_03121]	Definition of MAC multicast addresses in combination with VLANs in case of SwitchMacAddressLearningEnum.independentVlanLearning
[TPS_SYST_03122]	ISignal.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received"
[TPS_SYST_03123]	ISignal.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received" shall only cover non-optional members
[TPS_SYST_03124]	ISignal.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received" shall be defined in SenderReceiverInterface.dataElement order
[TPS_SYST_03125]	Enabling sending of GlobalTimeDomain Integrity Check Value (ICV)
[TPS_SYST_03126]	Enabling verification of GlobalTimeDomain Integrity Check Value (ICV)
[TPS_SYST_03127]	Support for CouplingPortDetails.framePreemptionSupport
[TPS_SYST_03128]	Traffic class specific enabling of egress frame preemption
[TPS_SYST_03129]	Definition of a defaultTrafficClass per CouplingPort
[TPS_SYST_03130]	Frame Mapping support

Table E.79: Added Specification Items in R24-11

E.17.2 Changed Specification Items in R24-11

Number	Heading
[TPS_SYST_01034]	Data Mappings can be applied to compositions and atomic software components
[TPS_SYST_01035]	Transformation of Data Mappings during flattening
[TPS_SYST_01070]	E2E Protection of ISignalGroups
[TPS_SYST_01071]	E2E Protection of several ISignalGroups in one ISignalIPdu
[TPS_SYST_01072]	Offset attributes of EndToEndDescription
[TPS_SYST_01073]	E2E Protection via COM Callouts
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_01137]	Several DataMappings may be defined for the same SystemSignal
[TPS_SYST_02073]	EndToEndTransformationDescription.profileName
[TPS_SYST_03007]	Ethernet port scheduler algorithm
[TPS_SYST_03010]	Ethernet switch packet to traffic class assignment
[TPS_SYST_03011]	Ethernet switch traffic class to FIFO assignment
[TPS_SYST_03085]	Pdu qualifies as dynamic length

Table E.80: Changed Specification Items in R24-11

E.17.3 Deleted Specification Items in R24-11

Number	Heading
[TPS_SYST_01116]	Frame Mapping is not supported by the AUTOSAR BSW
[TPS_SYST_02116]	Modeling of Service Discovery Pdus
[TPS_SYST_02119]	StaticSocketConnections for GeneralPurposePdus with category SD
[TPS_SYST_02391]	Mapping of MOST FIBEX elements to AUTOSAR elements
[TPS_SYST_03005]	VLAN re-tagging
[TPS_SYST_03015]	Offset time domain requires synchronized time domain

Table E.81: Deleted Specification Items in R24-11

E.17.4 Added Constraints in R24-11

Number	Heading
[constr_3763]	Allowed value for maxDeltaCounter in the context of a profileName
[constr_3764]	Applicability of CouplingPort.macAddressVlanAssignment
[constr_3765]	Applicability of MacAddressVlanMembership.vlan

▽

△

Number	Heading
[constr_3766]	Valid MacAddressVlanMembership.vlan target EthernetPhysicalChannel
[constr_3767]	NmNode.nmVariant setting to slavePassive
[constr_3768]	NmNode.nmVariant setting to slaveActive
[constr_3769]	NmNode.nmVariant setting to full
[constr_3770]	NmNode.nmVariant setting to passive
[constr_3771]	Range of NmCluster.nmLightTimeout
[constr_3779]	Number of ISignal.receptionDefaultValue elements
[constr_3780]	ISignal.receptionDefaultValue definition in case that the SOME/IP Serializer receives less data than expected
[constr_3781]	Each PNC assigned to multiple PhysicalChannels shall have a top level PNC-Coordinator
[constr_3782]	Consistent framePreemptionSupport setting in the scope of one CouplingPortConnection
[constr_3783]	Definition of CouplingPortFifo.trafficClassPreemptionSupport only in context of an Ethernet switch
[constr_3784]	Applicable CouplingPortFifo as predecessor for portScheduler = enhancedTrafficShaper
[constr_3785]	Exclusive definition of etsAvailableBandwidthInPercent or etsAvailableBandwidthInWeightValue
[constr_3786]	Consistent usage of either etsAvailableBandwidthInPercent or etsAvailableBandwidthInWeightValue for portScheduler = enhancedTrafficShaper
[constr_3787]	Existence of CouplingPortTrafficClassAssignment.trafficClass
[constr_3788]	Existence of CouplingPortFifo.assignedTrafficClass
[constr_3789]	Allowed values for CouplingPortFifo.assignedTrafficClass
[constr_3790]	Existence of CouplingPortDetails.defaultTrafficClass
[constr_3791]	Allowed values for CouplingPortDetails.defaultTrafficClass
[constr_3792]	FrameMapping between identical bus systems
[constr_9317]	StateDependentFirewall.firewallStateModeDeclaration reference restriction
[constr_9318]	Reception of CanFrameTriggerings with the same identifier by an EcuInstance
[constr_9319]	Value of BusMirrorChannelMappingCan.mirroringProtocol
[constr_9320]	Value of BusMirrorChannelMappingFlexray.mirroringProtocol
[constr_9321]	Same time base for all BusMirrorChannelMappings of one EcuInstance
[constr_9326]	Exclusive existence of ISignalTriggering.iSignal and ISignalTriggering.iSignalGroup
[constr_9330]	Derivation of network representation in case that several DataMappings are defined that map the same SystemSignal to different VariableDataPrototypes
[constr_9331]	E2E protection of a ClientServerOperation
[constr_9332]	Existence of J1939TpConnection.tpProtocolType

▽

△

Number	Heading
[constr_9333]	FibexElements in ECU_EXTRACT
[constr_9343]	Allowed J1939ProtectedIPdu.payload reference target
[constr_9346]	Existence of EthernetVlanTranslationTable.translatedVlanId
[constr_9347]	Range of EthernetVlanTranslationTable.ingressVlanId and EthernetVlanTranslationTable.translatedVlanId
[constr_9348]	EthernetVlanTranslationTable.translatedVlanId and vlanMembership

Table E.82: Added Constraints in R24-11

E.17.5 Changed Constraints in R24-11

Number	Heading
[constr_1001]	Value of dataId shall be unique
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_3153]	E2E header field reservation required by COM Based transformer
[constr_3163]	EndToEndTransformationISignalProps.minDataLength and EndToEndTransformationISignalProps.maxDataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76
[constr_3164]	EndToEndTransformationISignalProps.dataLength in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, and PROFILE_76
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76
[constr_3169]	EndToEndTransformationDescription.offset value in PROFILE_02, PROFILE_22 and PROFILE_76
[constr_3174]	EndToEndTransformationDescription settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76
[constr_3186]	Multiplicity of EndToEndTransformationDescription.dataIdMode in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3188]	Multiplicity of EndToEndTransformationDescription.counterOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76

▽

△

Number	Heading
[constr_3190]	Multiplicity of EndToEndTransformationDescription.crcOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3192]	Multiplicity of EndToEndTransformationDescription.dataIdNibbleOffset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m and PROFILE_76 or dataIdMode different from lower12Bit
[constr_3194]	Multiplicity of EndToEndTransformationDescription.offset in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, PROFILE_44m, PROFILE_76
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3267]	PduTriggerings in Service Discovery StaticSocketConnections
[constr_3268]	Service Discovery StaticSocketConnection aggregation by a SocketAddress
[constr_3269]	Service Discovery StaticSocketConnection remoteAddress reference to a TpPort
[constr_3270]	Service Discovery SocketConnection remoteAddress reference to an IP Address
[constr_3272]	SoConIPduIdentifier.headerId setting for SD StaticSocketConnections
[constr_3273]	Service Discovery multicast StaticSocketConnection's aggregation by an ApplicationEndpoint
[constr_3274]	Service Discovery unicast StaticSocketConnection's aggregation by an ApplicationEndpoint
[constr_3519]	Value of category of GlobalTimeDomain
[constr_3600]	Setting of EthernetCommunicationController.slaveActAsPassive-CommunicationSlave
[constr_3697]	Latest existence time of CanControllerXlConfiguration and CanControllerXlConfigurationRequirements
[constr_3714]	Multiple top level PNC-coordinators shall be allowed
[constr_3716]	SecuredIPdu.dynamicRuntimeLengthHandling for dynamic length Pdus
[constr_5049]	Ethernet switch packet to traffic class assignment restriction
[constr_5091]	Relevance of tcpRole attribute
[constr_5105]	Mapping of Pdu with dynamic length in a FlexrayFrame
[constr_5168]	pncGatewayType passive and connected ECUs
[constr_5221]	Multiplicity of EndToEndTransformationISignalProps.sourceId in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, PROFILE_22, and PROFILE_76
[constr_5273]	One ISignalTriggering pair allowed per EthernetPhysicalChannel for a ClientServerOperation
[constr_5380]	Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of an EcuInstance

▽

△

Number	Heading
[constr_9112]	Existence of GenericTp.tpTechnology
[constr_9115]	Existence of RtpTp.ssrc
[constr_9116]	Existence of RtpTp.tcpUdpConfig
[constr_9119]	Existence of Ieee1722Tp.streamIdentifier
[constr_9120]	Existence of HttpTp.protocolVersion
[constr_9121]	Existence of HttpTp.tcpTpConfig
[constr_9207]	Existence of `EndToEndProtectionISignalIPdu.iSignalIPdu`
[constr_9208]	Existence of `EndToEndProtectionISignalIPdu.iSignalGroup`
[constr_9209]	Existence of `EndToEndProtectionISignalIPdu.dataOffset`
[constr_9267]	Existence of `J1939TpConnection.broadcast`

Table E.83: Changed Constraints in R24-11

E.17.6 Deleted Constraints in R24-11

Number	Heading
[constr_3522]	`vlanModifier` and `vlanMembership`
[constr_5169]	pncGatewayType and (routing) paths
[constr_5391]	`ConsumedServiceInstance.allowedServiceProvider` reference restriction

Table E.84: Deleted Constraints in R24-11

E.17.7 Added Advisories in R24-11

none

E.17.8 Changed Advisories in R24-11

none

E.17.9 Deleted Advisories in R24-11

none

E.18 Constraint and Specification Item History of this document according to AUTOSAR R25-11

E.18.1 Added Specification Items in R25-11

Number	Heading
[TPS_SYST_02420]	Semantic of EcuPartitionToCoreMapping
[TPS_SYST_02421]	Semantics of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation
[TPS_SYST_02422]	Applicability of Ethernet model elements
[TPS_SYST_02423]	Compatibility of VariableDataPrototype typed by primitive data type of category VALUE in a PPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02424]	Compatibility of VariableDataPrototype typed by primitive data type of category VALUE in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02425]	Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category VALUE in a PPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping
[TPS_SYST_02426]	Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category VALUE in a RPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping
[TPS_SYST_02427]	Compatibility of VariableDataPrototype typed by primitive data type of category BOOLEAN in a PPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02428]	Compatibility of VariableDataPrototype typed by primitive data type of category BOOLEAN in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02429]	Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category BOOLEAN in a PPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping
[TPS_SYST_02430]	Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category BOOLEAN in a RPortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping



△

Number	Heading
[TPS_SYST_02431]	Compatibility of VariableDataPrototype typed by primitive data type of category STRING in a PortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02432]	Compatibility of ApplicationCompositeElementDataPrototype typed by primitive data type of category STRING in a PortPrototype and a SystemSignal to which the ApplicationCompositeElementDataPrototype is mapped by SenderReceiverToSignalGroupMapping
[TPS_SYST_02433]	Compatibility of VariableDataPrototype typed by an ARRAY in a PPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02434]	Compatibility of VariableDataPrototype typed by an ARRAY in a RPortPrototype and a SystemSignal to which the VariableDataPrototype is mapped by SenderReceiverToSignalMapping
[TPS_SYST_02435]	Semantics of J1939Node
[TPS_SYST_02436]	Attributes of GlobalTimeCorrectionProps applicable for GlobalTimeMaster
[TPS_SYST_02437]	Attributes of GlobalTimeCorrectionProps applicable for GlobalTimeSlave
[TPS_SYST_02438]	Types of OsTaskProxy
[TPS_SYST_02439]	OsTaskProxy offset is optional
[TPS_SYST_02440]	Semantic of CyclicHandlingComDataToOsTaskProxyMapping
[TPS_SYST_02441]	Mapping of VariableDataPrototypes to the same OsTaskProxy
[TPS_SYST_02442]	Concurrent processing of upper layer IPdus by the SOME/IP TP segmenter
[TPS_SYST_03131]	"Fire & Forget" method assignment to AbstractServiceInstances
[TPS_SYST_03132]	category of IEEE1722TpIidcConnection
[TPS_SYST_03133]	Pass-through DDS transformer
[TPS_SYST_03134]	DdsCpProvidedServiceInstance deployment
[TPS_SYST_03135]	Static unicast configuration between DdsCpProvidedServiceInstance and DdsCpConsumedServiceInstance
[TPS_SYST_03136]	Maximal number of unicast clients that may connect to the local unicast server address
[TPS_SYST_03137]	DdsCpProvidedServiceInstance Unicast Address UDP port configuration rules
[TPS_SYST_03138]	PduTriggering that is used for DDS ClientServer Communication
[TPS_SYST_03139]	DdsCpConsumedServiceInstance deployment
[TPS_SYST_03140]	Static unicast configuration between DdsCpConsumedServiceInstance and DdsCpProvidedServiceInstance
[TPS_SYST_03141]	Maximal number of unicast servers that may connect to the local unicast client address

▽

△

Number	Heading
[TPS_SYST_03142]	DdsCpConsumedServiceInstance Unicast Address UDP port configuration rules
[TPS_SYST_03143]	channelSynchronousWakeup is set to FALSE and no explicitWakeupChannel reference
[TPS_SYST_03144]	channelSynchronousWakeup is set to FALSE and at least one explicitWakeupChannel reference
[TPS_SYST_03145]	channelSynchronousWakeup is set to TRUE
[TPS_SYST_03146]	pncSynchronousWakeup is set to FALSE and no explicitWakeupPnc reference
[TPS_SYST_03147]	pncSynchronousWakeup is set to FALSE and at least one explicitWakeupPnc reference
[TPS_SYST_03148]	pncSynchronousWakeup is set to TRUE and no explicitWakeupPnc reference
[TPS_SYST_03149]	pncSynchronousWakeup is set to TRUE and at least one explicitWakeupPnc reference

Table E.85: Added Specification Items in R25-11

E.18.2 Changed Specification Items in R25-11

Number	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01034]	Data Mappings can be applied to compositions and atomic software components
[TPS_SYST_02015]	Large Data Path in COM: only one ISignal mapped to the ISignalIPdu
[TPS_SYST_02016]	Large Data Path in COM: only Transformer output and UINT8_N or UINT8_DYN supported
[TPS_SYST_02017]	Large Data Path in COM: Opaque ISignalToIPduMapping.packingByteOrder
[TPS_SYST_02018]	Large Data Path in COM: ISignalToIPduMapping.startPosition shall be 0
[TPS_SYST_02019]	Large Data Path in COM: ISignalToIPduMapping.transferProperty shall be triggered or triggeredWithoutRepetition for sent ISignals
[TPS_SYST_02021]	Large Data Path in COM: ISignalToIPduMapping.updateIndicationBitPosition shall not be defined
[TPS_SYST_02022]	Large Data Path in COM: Only the transmissionModeTrueTiming defined
[TPS_SYST_02023]	Large Data Path in COM: DataFilter "always" if TransmissionModeCondition defined
[TPS_SYST_02024]	Large Data Path in COM: No ModeDrivenTransmissionModeCondition defined
[TPS_SYST_02025]	Large Data Path in COM: Only EventControlledTiming defined

▽

△

Number	Heading
[TPS_SYST_02026]	Large Data Path in COM: Only `EventControlledTiming` with no repetition defined
[TPS_SYST_02028]	Large Data Path in COM: No `ISignalPort.dataFilter` defined
[TPS_SYST_02114]	Mapping of `SwComponentPrototype`s onto `EcuInstance`s
[TPS_SYST_02390]	Relevance of Large Data Path in COM spec items for the sender and for the receiver side.
[TPS_SYST_03009]	Ethernet switch `CouplingPortCreditBasedShaper`
[TPS_SYST_03034]	Handling safe signal reception
[TPS_SYST_03085]	`Pdu` qualifies as dynamic length
[TPS_SYST_03090]	DDS Methods Request/Reply Topic
[TPS_SYST_03091]	DDS Fields Request/Reply Topic
[TPS_SYST_03093]	DDS Operation
[TPS_SYST_03095]	DDS Event QoSProfile
[TPS_SYST_03111]	Ethernet switch `SwitchAtsInstanceEntry`
[TPS_SYST_05003]	Usage of `DiagnosticConnection` in combination with a TP
[TPS_SYST_05022]	Semantics of `PduTriggering.secOcCryptoServiceMapping`

Table E.86: Changed Specification Items in R25-11

E.18.3 Deleted Specification Items in R25-11

Number	Heading
[TPS_SYST_01070]	E2E Protection of `ISignalGroup`s
[TPS_SYST_01071]	E2E Protection of several `ISignalGroup`s in one `ISignalIPdu`
[TPS_SYST_01072]	Offset attributes of `EndToEndDescription`
[TPS_SYST_01073]	E2E Protection via COM Callouts
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_02020]	LdCom: No `IPduTiming.minimumDelay` defined
[TPS_SYST_02027]	LdCom: No `ISignalPort.timeout` reception timeout defined
[TPS_SYST_02164]	LdCom: No `ISignalPort.firstTimeout` reception timeout defined
[TPS_SYST_03001]	LdCom: `ISignalIPdu` not part of any `ISignalIPduGroup`
[TPS_SYST_03122]	`ISignal`.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received"
[TPS_SYST_03123]	`ISignal`.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received" shall only cover non-optional members
[TPS_SYST_03124]	`ISignal`.receptionDefaultValue configured for SOME/IP Serializer "less data than expected received" shall be defined in `SenderReceiverInterface.dataElement` order

Table E.87: Deleted Specification Items in R25-11

E.18.4 Added Constraints in R25-11

Number	Heading
[constr_3796]	Applicable attributes of IEEE1722TpIidcConnection for category IIDC
[constr_3797]	Applicable attributes of IEEE1722TpIidcConnection for category IEC61883
[constr_3799]	DDS Fire & Forget Method definition
[constr_3800]	Existence of DdsCpServiceInstance.discoveryType
[constr_3801]	Existence of DdsCpServiceInstance.serviceInterfaceId
[constr_3802]	Existence of DdsCpServiceInstance.majorVersion
[constr_3803]	Existence of DdsCpServiceInstance.serviceMinorVersion
[constr_3804]	Existence of DdsCpServiceInstance.transportProtocol
[constr_3805]	Existence of DdsServiceInstanceEventCp.eventTriggering
[constr_3806]	Existence of DdsServiceInstanceOperationCp. operationRequestTriggering
[constr_3807]	Existence of DdsServiceInstanceFieldCp reference
[constr_3808]	Existence of DdsCpProvidedServiceInstance. providedServiceInstanceId
[constr_3809]	Existence of DdsCpConsumedServiceInstance. requiredServiceInstanceId
[constr_3810]	Value range of DdsCpProvidedServiceInstance. providedServiceInstanceId
[constr_3811]	DdsCpProvidedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_3812]	DdsCpProvidedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast
[constr_3813]	DdsCpProvidedServiceInstance unicast address references target
[constr_3814]	Multiple SocketAddress entries with the same IP Address and Port in the context of a given EcuInstance
[constr_3815]	DDS Service communication is restricted to one VLAN
[constr_3816]	Value range of DdsCpConsumedServiceInstance. requiredServiceInstanceId
[constr_3817]	DdsCpConsumedServiceInstance.localUnicastAddress shall be IP Unicast
[constr_3818]	DdsCpConsumedServiceInstance.staticRemoteUnicastAddress shall be IP Unicast
[constr_3819]	DdsCpConsumedServiceInstance unicast address references target
[constr_3820]	DDS Service communication is restricted to one VLAN
[constr_3821]	DDS transformer configuration
[constr_3822]	No further transfomers in scope of the DDS transformer
[constr_3824]	No explicitWakeupChannel reference for channelSynchronousWakeup = TRUE
[constr_9349]	An ISignal shall not be used for SOME/IP and DDS transport at the same time
[constr_9354]	Definition of unconditional IPdu Timing
[constr_9355]	Category of HwElement for EcuPartitionToCoreMapping

▽

△

Number	Heading
[constr_9358]	ISignal that references a DataTransformation shall have the dataTypePolicy set to transformingISignal
[constr_9359]	Different EcuPartitionToCoreMapping.coreId attribute values shall not be assigned to the same processingUnit
[constr_9360]	DataMapping not available for elements of PortInterfaces with isService = true
[constr_9361]	FlatInstanceDescriptor.dataPrototype reference restriction
[constr_9362]	Mutually exclusive existence of FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation vs. FlatInstanceDescriptor.ecuExtractReference
[constr_9363]	FlatInstanceDescriptor.dataPrototype and FlatInstanceDescriptor.bswImplementation
[constr_9366]	GlobalTimeSlave.followUpTimeoutValue not applicable to GlobalTimeFrSlave
[constr_9367]	Definition of several local ApplicationEndpoints with identical properties (IP, Port, Protocol) is forbidden
[constr_9368]	Applicable attributes for an Ethernet switch
[constr_9369]	Applicable attributes for an Ethernet node on the classic platform
[constr_9371]	Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a RPortPrototype mapped to the SystemSignal
[constr_9372]	Compatibility of CompuMethod defined in the context of a SystemSignal and CompuMethod defined in the context of the DataPrototype in a PPortPrototype mapped to the SystemSignal
[constr_9373]	Compatibility of DataConstr between SystemSignal and DataPrototype in a PPortPrototype
[constr_9374]	Compatibility of DataConstr between SystemSignal and DataPrototype in a RPortPrototype
[constr_9375]	Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a PPortPrototype mapped to the SystemSignal
[constr_9376]	Compatibility of CompuScales defined in the context of a SystemSignal and CompuScales defined in the context of a DataPrototype of a RPortPrototype mapped to the SystemSignal
[constr_9377]	J1939Node.pduTriggering reference restriction
[constr_9378]	J1939Nodes on the same EcuInstance shall not share PduTriggerings
[constr_9379]	Consistent mapping of software-component to J1939Node
[constr_9380]	Consistent mapping of J1939ControllerApplication to EcuInstance
[constr_9381]	J1939Node shall be referenced by a single J1939ControllerApplicationToJ1939NodeMapping
[constr_9382]	AppOsTaskProxyToEcuTaskProxyMapping.appTaskProxy shall only target AppTaskProxy
[constr_9383]	AppOsTaskProxyToEcuTaskProxyMapping.ecuTaskProxy shall only target EcuTaskProxy

▽

△

Number	Heading
[constr_9384]	Relation between both CyclicHandlingComDataToOsTaskProxyMapping.rxCycleProcessTime and CyclicHandlingComDataToOsTaskProxyMapping.offset and OsTaskProxy.period
[constr_9385]	Existence of CyclicHandlingComDataToOsTaskProxyMapping.osTaskProxy
[constr_9386]	Existence of CyclicHandlingComDataToOsTaskProxyMapping.variableDataPrototype
[constr_9387]	One CyclicHandlingComDataToOsTaskProxyMapping for every VariableDataPrototype with ReceptionComSpecProps.comHandlerTaskMappingEnabled = TRUE
[constr_9388]	CanNmNode.txNmPdu reference restriction
[constr_9389]	UdpNmNode.txNmPdu reference restriction
[constr_9390]	J1939NmNode.txNmPdu reference restriction
[constr_9391]	FlexrayNmNode.txNmPdu reference restriction
[constr_9395]	Each call ISignal shall have exactly one corresponding return ISignal that is associated with the same ClientServerOperation and PhysicalChannel
[constr_9396]	SomeipTpConnection.tpConcurrentProcessingSdu reference restriction
[constr_9397]	Exclusive usage of tpSdu or tpConcurrentProcessingSdu

Table E.88: Added Constraints in R25-11

E.18.5 Changed Constraints in R25-11

Number	Heading
[constr_1669]	Existence of PduTriggering.secOcCryptoServiceMapping
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3060]	Allowed Attributes for networkRepresentationProps and physicalProps
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3096]	Allowed values for diagnosticMessageType
[constr_3102]	Restriction on usage of J1939NodeName attributes
[constr_3215]	TransformationTechnology.version and TransformationTechnology.protocol settings for request and response of a client/server communication on the same TransformationTechnology
[constr_3249]	Category of HwElement for SwcToEcuMapping
[constr_3263]	Restriction of usage of SwcToEcuMapping in a System
[constr_3266]	TransformationTechnology.hasInternalState setting for a SOME/IP Transformer
[constr_3739]	Value of ISignal.dataTypePolicy for all ISignals associated with a DdsCpServiceInstance
[constr_3740]	Existence of DdsServiceInstanceEventCp.ddsEventTopic

▽

△

Number	Heading
[constr_5273]	One ISignalTriggering pair allowed per EthernetPhysicalChannel for a ClientServerOperation
[constr_5378]	PduTriggering shall only be referenced once from a SomeipTpConnection in the role tpSdu or tpConcurrentProcessingSdu

Table E.89: Changed Constraints in R25-11

E.18.6 Deleted Constraints in R25-11

Number	Heading
[constr_1001]	Value of dataId shall be unique
[constr_1002]	End-to-end protection does not support n:1 communication
[constr_1463]	Applicable values for J1939Cluster.networkId
[constr_3239]	Consistent mapping of software-component to J1939NmNode
[constr_3240]	Consistent mapping of J1939ControllerApplication to EcuInstance
[constr_3257]	TimeSyncTechnology of servers and clients in a time synchronized network.
[constr_3299]	SocketAddress.pathMtuDiscoveryEnabled setting dependency
[constr_3435]	Applicability of CouplingPort.macMulticastAddress
[constr_3738]	ISignal that has dataTypePolicy set to ddsSignal or to ddsService shall not reference a DataTransformation
[constr_3779]	Number of ISignal.receptionDefaultValue elements
[constr_3780]	ISignal.receptionDefaultValue definition in case that the SOME/IP Serializer receives less data than expected
[constr_5029]	J1939NmCluster is not allowed to reference a TtcanCluster
[constr_5459]	Existence of dataLength
[constr_5460]	Existence of policyAction
[constr_5461]	Existence of timeInterval
[constr_9112]	Existence of GenericTp.tpTechnolog
[constr_9115]	Existence of RtpTp.ssrc
[constr_9116]	Existence of RtpTp.tcpUdpConfig
[constr_9119]	Existence of Ieee1722Tp.streamIdentifier
[constr_9120]	Existence of HttpTp.protocolVersion
[constr_9121]	Existence of HttpTp.tcpTpConfig
[constr_9207]	Existence of EndToEndProtectionISignalIPdu.iSignalIPdu
[constr_9208]	Existence of EndToEndProtectionISignalIPdu.iSignalGroup
[constr_9209]	Existence of EndToEndProtectionISignalIPdu.dataOffset
[constr_9267]	Existence of J1939TpConnection.broadcast
[constr_9272]	Existence of TlsPskIdentity.preSharedKey

▽

△

Number	Heading
[constr_9309]	Existence of GlobalTimeCanMaster.syncConfirmationTimeout

Table E.90: Deleted Constraints in R25-11

E.18.7 Added Advisories in R25-11

none

E.18.8 Changed Advisories in R25-11

none

E.18.9 Deleted Advisories in R25-11

none

E.19 Constraint and Specification Item History of this document according to AUTOSAR R26-11

E.19.1 Added Specification Items in R26-11

Number	Heading
[TPS_SYST_01159]	J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress in case tpProtocolType is fpp
[TPS_SYST_01160]	Broadcast transmission in case tpProtocolType is fpp and J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress is omitted
[TPS_SYST_01161]	J1939TpConnection.receiver or J1939TpConnection.receiver.tpAddress in case tpProtocolType is bam or fd_bam
[TPS_SYST_02116]	Modeling of Service Discovery Pdus
[TPS_SYST_02443]	Usage of portNumber with value 0 on a local port
[TPS_SYST_02444]	eventGroupControlType of a unicast and multicast Event
[TPS_SYST_02445]	SecuredIPdus with encryption
[TPS_SYST_03150]	Mixed Large Data Path in COM and Traditional COM for Client Server communication
[TPS_SYST_03151]	Semantics of GlobalTimeDomain.clonedFromTimeDomain

Table E.91: Added Specification Items in R26-11

E.19.2 Changed Specification Items in R26-11

Number	Heading
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01106]	Usage of additional directPdu in case of variable length sdu
[TPS_SYST_01133]	Partial Network Clusters
[TPS_SYST_02064]	Reception acceptance of contained IPdus
[TPS_SYST_02097]	Basic definition of contained IPdus
[TPS_SYST_02146]	Explicit definition of PNC Request Bit Vector at NmPdu
[TPS_SYST_02153]	Standardized values for the attribute category of meta-class SecureCommunicationFreshnessProps
[TPS_SYST_02154]	Standardized values for the attribute category of meta-class SecureCommunicationAuthenticationProps
[TPS_SYST_02196]	PduTriggering is referenced by several ContainerIPdus
[TPS_SYST_02215]	Usage of portNumber with value 0 on a remote port
[TPS_SYST_02230]	PduActivationRoutingGroups for event groups
[TPS_SYST_02246]	PduActivationRoutingGroups for ConsumedEventGroups
[TPS_SYST_02349]	Recommended configuration settings for E2E Profile 4m configuration setting A
[TPS_SYST_02350]	Recommended configuration settings for E2E Profile 7m configuration setting A
[TPS_SYST_03067]	Definition of pncVectorOffset
[TPS_SYST_03068]	Definition of pncVectorLength
[TPS_SYST_03069]	User data shall be defined within empty space of the NmPdu
[TPS_SYST_03070]	User data shall be before the PNC Request Bit Vector or after the PNC Request Bit Vector
[TPS_SYST_03071]	Available space of user data with PNC Request Bit Vector
[TPS_SYST_03073]	Derivation of NmPnFilterMaskByte
[TPS_SYST_05027]	Standardized values of CryptoServicePrimitive.algorithmMode

Table E.92: Changed Specification Items in R26-11

E.19.3 Deleted Specification Items in R26-11

Number	Heading
[TPS_SYST_01143]	DataMapping on the sender side for elements of a composite data type
[TPS_SYST_02076]	networkRepresentationProps in case the dataTypePolicy is set to transformingISignal
[TPS_SYST_02118]	Rules for the creation of references to IPduPorts from PduTriggerings related to GeneralPurposePdus with category SD

▽

△

Number	Heading
[TPS_SYST_02372]	Precedence of ContainedIPduProps settings

Table E.93: Deleted Specification Items in R26-11

E.19.4 Added Constraints in R26-11

Number	Heading
[constr_3826]	DdsCpTopic.ddsPartition reference restriction
[constr_3827]	Existence of IkePqPresharedKey.pqPresharedKeyId
[constr_3828]	Existence of IkePqPresharedKey.pqPresharedKey
[constr_3829]	CommunicationCluster.comMLightTimeout not applicable for EthernetCluster
[constr_9398]	dataTypePolicy = transformingISignal not applicable for ISignals that are transformed by the ComBasedTransformer
[constr_9399]	Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if no data transformation is used
[constr_9400]	Mapping of VariableDataPrototype typed by a composite data type to SystemSignals in SystemSignalGroup by SenderReceiverToSignalGroupMapping if comBasedSignalGroup Transformation is used
[constr_9401]	Mapping of VariableDataPrototype typed by a array data type consisting of UINT8 elements to exactly one SystemSignal
[constr_9402]	PduActivationRoutingGroup with PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or activationUnicastAndMulticast that is aggregated by a ConsumedEventGroup
[constr_9403]	J1939TpProtocolTypeEnum values fd_bam, fd_bam_cmdt, and fd_cmdt shall only be used on CAN FD clusters
[constr_9404]	J1939Cluster.usesMultiPg attribute shall only be set to a value on CAN FD clusters
[constr_9407]	Standardized values for the attribute category of meta-class PortElementToCommunicationResourceMapping
[constr_9408]	J1939TpConnection.tpPg in case J1939TpConnection.tpProtocolType is fpp
[constr_9421]	Exactly two MacSecConnectivityAssociation.couplingPort references for a switched Ethernet connection
[constr_9422]	Consistent reference to a VLAN using AbstractMacSecCommunication.vlan
[constr_9423]	Coverage of all VLANs referenced by a CouplingPort by MacSecConnectivityAssociation and MacSecBypassedCommunication
[constr_9424]	10BASE-T1S MacSecConnectivityAssociation CouplingPortConnection reference consistency

▽

△

Number	Heading
[constr_9425]	Existence of CanSecSecureChannel.csci
[constr_9426]	Coverage of all CanFrameTriggerings and SoConIPduIdentifiers in the scope of a CanCommunicationConnector by MessageToCanSecSecureChannelMapping and MessageToBypassCanSecCommunicationMapping
[constr_9427]	Existence of MessageToCanSecSecureChannelMapping.secureChannel
[constr_9428]	Consistent MessageToCanSecSecureChannelMapping.secureChannel and AbstractMessageToSecureChannelMapping.canFrameTriggering mapping

Table E.94: Added Constraints in R26-11

E.19.5 Changed Constraints in R26-11

Number	Heading
[constr_1368]	Limitation of the target of references from DiagnosticConnection
[constr_3025]	Usage of NPdus in TpConnections
[constr_3116]	Overlap of ClientIdRanges in the context of the enclosing System
[constr_3117]	Allowed value of attribute clientId
[constr_3136]	Allowed payload of SecuredIPdus
[constr_3141]	Only IPdus shall be part of a ContainerIPdu
[constr_3142]	Mandatory headerIdLongHeader for longHeader
[constr_3143]	Mandatory headerIdShortHeader for shortHeader
[constr_3167]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m
[constr_3208]	executeDespiteDataUnavailability usage restriction for received ISignals
[constr_3325]	SecureCommunicationFreshnessProps, SecureCommunicationAuthenticationProps and CryptoServicePrimitive attribute value settings for standardized AUTOSAR security profiles
[constr_3402]	Mandatory offset if noHeader is used
[constr_3404]	Usage of ContainedIPduProps.updateIndicationBitPosition
[constr_3454]	Unique headerIdLongHeader for acceptConfigured
[constr_3455]	Unique headerIdShortHeader for acceptConfigured
[constr_3767]	CommunicationConnector.nmVariant setting to slavePassive
[constr_3768]	CommunicationConnector.nmVariant setting to slaveActive
[constr_3769]	CommunicationConnector.nmVariant setting to full

▽

△

Number	Heading
[constr_3770]	CommunicationConnector.nmVariant setting to passive
[constr_5074]	PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast that is aggregated by an EventHandler
[constr_5075]	Allowed references of SoConIPduIdentifiers by PduActivationRoutingGroup with eventGroupControlType set to activationMulticast or activationUnicastAndMulticast and allowed SoConIPduIdentifier references
[constr_5361]	MACsec configuration is only allowed on CouplingPorts aggregated by switch ports and EthernetCommunicationController
[constr_5362]	Relation between ContainerIPdu and contained PduTriggerings on sender side
[constr_5380]	Assignment of the same event Pdu to several EventHandlers is forbidden in case one of the EventHandlers has the multicastThreshold set to a value greater than 0 in the context of one VLAN

Table E.95: Changed Constraints in R26-11

E.19.6 Deleted Constraints in R26-11

Number	Heading
[constr_3021]	Mapping of SensorActuatorSwComponents to SensorActuator HwElements
[constr_3058]	References from SenderRecArrayElementMapping and from SenderRecRecordElementMapping to SystemSignals are not allowed within a SenderReceiverCompositeElementToSignalMapping
[constr_3144]	Mandatory IPdu.containedIPduProps for contained IPdus
[constr_3249]	Category of HwElement for SwcToEcuMapping
[constr_3327]	Effect of EndToEndTransformationDescription.upperHeaderBitsToShift value in PROFILE_22
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_5269]	Exclusion of ContainedIPduProps.containedPduTriggering reference
[constr_5270]	Exclusive usage of ContainerIPdu.containedPduTriggering and ContainerIPdu.containedIPduTriggeringProps
[constr_5315]	FlexrayArTpConnections within the same FlexrayArTpChannel not allowed to have the same address information
[constr_5375]	IPdu shall only be referenced once from a FlexrayArTpConnection in the role directTpSdu or reversedTpSdu on a FlexrayCluster
[constr_5465]	Existence of softwareComposition
[constr_5475]	Existence of SenderReceiverCompositeElementToSignalMapping.systemSignal
[constr_5476]	Existence of SenderReceiverCompositeElementToSignalMapping.typeMapping

▽

△

Number	Heading
[constr_9238]	Existence of FlexrayArTpChannel.ackType
[constr_9239]	Existence of FlexrayArTpChannel.extendedAddressing
[constr_9240]	Existence of FlexrayArTpChannel.maximumMessageLength
[constr_9241]	Existence of FlexrayArTpChannel.minimumSeparationTime
[constr_9242]	Existence of FlexrayArTpChannel.multicastSegmentation
[constr_9243]	Existence of FlexrayArTpChannel.tpConnection
[constr_9244]	Existence of FlexrayArTpConnection.directTpSdu
[constr_9245]	Existence of FlexrayArTpConnection.source
[constr_9246]	Existence of FlexrayArTpConnection.target
[constr_9268]	Existence of J1939TpConnection.dataPdu
[constr_9269]	Existence of J1939TpConnection.flowControlPdu

Table E.96: Deleted Constraints in R26-11

E.19.7 Added Advisories in R26-11

Number	Heading
[advisory_02005]	Usable portNumbers for AUTOSAR Communication
[advisory_03005]	ConsumedServiceInstance.remoteUnicastAddress should not reference an ApplicationEndpoint with portNumber = 0

Table E.97: Added Advisories in R26-11

E.19.8 Changed Advisories in R26-11

none

E.19.9 Deleted Advisories in R26-11

none

F Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

Class	***ARElement*** (abstract)			
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractMessageToSecureChannelMapping*, AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicabilityInfoSet, ApplicationPartition, *AutosarDataType*, *BaseType*, BlueprintMappingSet, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CpSoftwareCluster, CpSoftwareClusterBinaryManifestDescriptor, CpSoftwareClusterMappingSet, CpSoftwareClusterResourcePool, CryptoEllipticCurveProps, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, CryptoSignatureScheme, DataConstr, DataTransformationSet, DataTypeMappingSet, DdsCpConfig, *DiagnosticCommonElement*, DiagnosticConnection, DiagnosticContributionSet, DltArgumentPropsSet, DltContext, DltEcu, Documentation, E2EProfileCompatibilityProps, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EthIpProps, EthTcpIpIcmpProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FirewallRule, FlatMap, GeneralPurposeConnection, HwCategory, HwElement, HwType, *IEEE1722TpConnection*, IPSecConfigProps, IPv6ExtHeaderFilterSet, *IdsCommonElement*, IdsDesign, *Implementation*, ImpositionTimeDefinitionGroup, InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKayProps, MacSecParticipantSet, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, OsTaskProxy, PhysicalDimension, PhysicalDimensionMappingSet, *PortInterface*, PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, *SecureComProps*, *SecurityConnectivityAssociation*, SignalServiceTranslationPropsSet, SomeipSdClientEventGroupTimingConfig, SomeipSdClientServiceInstanceConfig, SomeipSdServerEventGroupTimingConfig, SomeipSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, SwComponentMappingConstraints, *SwComponentType*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, SwcBswMapping, System, SystemComSpecDefinitionSet, SystemSignal, SystemSignalGroup, TDCpSoftwareClusterMappingSet, TcpOptionFilterSet, *TimingExtension*, TlsConnectionGroup, TlvDataIdDefinitionSet, TransformationPropsSet, Unit, UnitGroup, *UploadablePackageElement*, ViewMapSet			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.1: ARElement

Class	**ARPackage**			
Note	AUTOSAR package, allowing to create top level packages to structure the contained ARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM.			
Base	*ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ARPackage.arPackage, AUTOSAR.arPackage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**ARPackage**			
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.shortName, element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	aggr	This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. **Tags:** atp.Splitkey=referenceBase, referenceBase.shortLabel xml.sequenceOffset=10

Table F.2: ARPackage

Class	**AUTOSAR**			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:** xml.globalElement=true			
Base	*ARObject*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, ar Package.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Stereotypes:** atpStructuredComment **Tags:** atp.Splitkey=fileInfoComment xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false

▽

△

Class	**AUTOSAR**			
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=20

Table F.3: AUTOSAR

Class	**AdminData**			
Note	AdminData represents the ability to express administrative information and custom extensions for an element. This administration information is to be treated as meta-data such as revision id or state of the file. There are basically the following kinds of meta-data • The language and/or used languages. • Revision information covering e.g. revision number, state, release date, changes. Note that this information can be given in general as well as related to a particular company. • Document meta-data specific for a company Beside that a custom extension of model-data is possible by • Special data			
Base	*ARObject*			
Aggregated by	AUTOSAR.adminData, *Describable*.adminData, *Identifiable*.adminData			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
docRevision (ordered)	DocRevision	*	aggr	This allows to denote information about the current revision of the object. Note that information about previous revisions can also be logged here. The entries shall be sorted descendant by date in order to reflect the history. Therefore the most recent entry representing the current version is denoted first. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=50 xml.typeElement=false xml.typeWrapperElement=false
language	LEnum	0..1	attr	This attribute specifies the master language of the document or the document fragment. The master language is the one in which the document is maintained and from which the other languages are derived from. In particular in case of inconsistencies, the information in the master language is priority. **Tags:** xml.sequenceOffset=20
sdg	Sdg	*	aggr	This property allows to keep special data which is not represented by the standard model. It can be utilized to keep e.g. tool specific data. **Tags:** atp.Splitkey=sdg, sdg.sdgCaption.shortName xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=60 xml.typeElement=false xml.typeWrapperElement=false

▽

△

Class	AdminData			
usedLanguages	MultiLanguagePlainText	0..1	aggr	This property specifies the languages which are provided in the document. Therefore it should only be specified in the top level admin data. For each language provided in the document there is one entry in MultilanguagePlain Text. The content of each entry can be used for illustration of the language. The used language itself depends on the language attribute in the entry. **Tags:** atp.Splitkey=usedLanguages xml.sequenceOffset=30

Table F.4: AdminData

Class	AnyInstanceRef			
Note	Describes a reference to any instance in an AUTOSAR model. This is the most generic form of an instance ref. Refer to the superclass notes for more details.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ApmcInstanceReferenceValue.value, ApmcUpstreamDocInstanceReferenceValue.value, ApmcUri InstanceReferenceValue.value, Collection.collectedInstance, Collection.sourceInstance, Documentation Context.feature, EcucInstanceReferenceValue.value, FlatInstanceDescriptor.ecuExtractReference, Flat InstanceDescriptor.upstreamReference, RptContainer.byPassPoint, RptHook.rptArHook, SecurityEvent ReportInstanceValue.object, ViewMap.firstElementInstance, ViewMap.secondElementInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	AtpClassifier	0..1	ref	This is the base from which navigation path begins. **Stereotypes:** atpDerived
contextElement (ordered)	AtpPrototype	*	ref	This is one step in the navigation path specified by the instance ref.
target	AtpFeature	0..1	ref	This is the target of the instance ref.

Table F.5: AnyInstanceRef

Class	ApplicationArrayDataType			
Note	An application data type which is an array, each element is of the same application data type. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, *ARObject*, *ApplicationCompositeDataType*, *ApplicationDataType*, *AtpBlueprint*, *Atp Classifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.
element	ApplicationArray Element	0..1	aggr	This association implements the concept of an array element. That is, in some cases it is necessary to be able to identify single array elements, e.g. as input values for an interpolation routine. **Tags:** atp.Splitkey=element, element.shortName

Table F.6: ApplicationArrayDataType

Class	**ApplicationArrayElement**			
Note	Describes the properties of the elements of an application array data type.			
Base	*ARObject*, *ApplicationCompositeElementDataPrototype*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationArrayDataType.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled.
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls how the information about the array size shall be interpreted.
indexDataType	ApplicationPrimitive DataType	0..1	ref	This reference can be taken to assign a `CompuMethod` of category `TEXTTABLE` to the array. The texttable entries associate a textual value to an index number such that the element with that index number is represented by a symbolic name. **Tags:** atp.Splitkey=indexDataType
maxNumberOf Elements	PositiveInteger	0..1	attr	The maximum number of elements that the array can contain. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table F.7: ApplicationArrayElement

Class	***ApplicationCompositeDataType*** (abstract)			
Note	Abstract base class for all application data types composed of other data types.			
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ApplicationArrayDataType, ApplicationRecordDataType			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table F.8: ApplicationCompositeDataType

Class	***ApplicationCompositeElementDataPrototype*** (abstract)			
Note	This class represents a data prototype which is aggregated within a composite application data type (record or array). It is introduced to provide a better distinction between target and context in instance Refs.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ApplicationArrayElement, ApplicationRecordElement			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	ApplicationDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table F.9: ApplicationCompositeElementDataPrototype

Class	***ApplicationDataType*** (abstract)			
Note	`ApplicationDataType` defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An `ApplicationDataType` represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianess, etc. It should be possible to model the application level aspects of a VFB system by using `ApplicationDataType`s only.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*ApplicationCompositeDataType*, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.10: ApplicationDataType

Class	**ApplicationError**			
Note	This is a user-defined error that is associated with an element of an AUTOSAR interface. It is specific for the particular functionality or service provided by the AUTOSAR software component.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ClientServerInterface.possibleError			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
errorCode	Integer	0..1	attr	The RTE generator is forced to assign this value to the corresponding error symbol. Note that for error codes certain ranges are predefined (see RTE specification).

Table F.11: ApplicationError

Class	**ApplicationPrimitiveDataType**			
Note	A primitive data type defines a set of allowed values. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, *ARObject*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.12: ApplicationPrimitiveDataType

Class	**ApplicationRecordDataType**			
Note	An application data type which can be decomposed into prototypes of other application data types. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, *ARObject*, *ApplicationCompositeDataType*, *ApplicationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	ApplicationRecordDataType			
element (ordered)	ApplicationRecord Element	*	aggr	Specifies an element of a record. The aggregation of `ApplicationRecordElement` is subject to variability with the purpose to support the conditional existence of elements inside a `ApplicationRecordData` Type. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table F.13: ApplicationRecordDataType

Class	ApplicationRecordElement			
Note	Describes the properties of one particular element of an application record data type.			
Base	*ARObject*, *ApplicationCompositeElementDataPrototype*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationRecordDataType.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing `ApplicationRecordElement` as optional. This means the that, at runtime, the `ApplicationRecordElement` may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the `ApplicationRecordElement` as not valid at the sending end of a communication and determine its validity at the receiving end.

Table F.14: ApplicationRecordElement

Class	ApplicationSwComponentType			
Note	The `ApplicationSwComponentType` is used to represent the application software. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table F.15: ApplicationSwComponentType

Class	ArgumentDataPrototype			
Note	An argument of an operation, carries direction and implementation information.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ClientServerOperation.argument			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
direction	ArgumentDirection Enum	0..1	attr	This attribute specifies the direction of the argument.

▽

△

Class	**ArgumentDataPrototype**			
serverArgument ImplPolicy	ServerArgumentImpl PolicyEnum	0..1	attr	This defines how the argument type of the servers `RunnableEntity` is implemented. If the attribute is not defined this has the same semantics as if the attribute is set to the value `useArgumentType` for primitive arguments and structures.

Table F.16: ArgumentDataPrototype

Enumeration	**ArgumentDirectionEnum**
Note	Use cases: • Arguments in ClientServerOperation can have different directions that need to be formally indicated because they have an impact on how the function signature looks like eventually. • Arguments in BswModuleEntry already determine a function signature, but the direction is used to specify the semantics, especially of pointer arguments.
Aggregated by	ArgumentDataPrototype.direction, DiagnosticSovdAccessArgument.direction, RunnableEntity Argument.direction, SwServiceArg.direction
Literal	***Description***
in	The argument value is passed to the callee. **Tags:** atp.EnumerationLiteralIndex=0
inout	The argument value is passed to the callee but also passed back from the callee to the caller. **Tags:** atp.EnumerationLiteralIndex=1
out	The argument value is passed from the callee to the caller. **Tags:** atp.EnumerationLiteralIndex=2

Table F.17: ArgumentDirectionEnum

Class	**ArrayValueSpecification**			
Note	Specifies the values for an array.			
Base	*ARObject*, *CompositeValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.argument, Constant Specification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compare Value, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compare Value, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
element (ordered)	ValueSpecification	*	aggr	The value for a single array element. All Value Specifications aggregated by ArrayValueSpecification shall have the same structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**ArrayValueSpecification**			
intendedPartial Initialization Count	PositiveInteger	0..1	attr	This attribute shall only have a meaning for dynamic arrays and shall be taken as a sanity check: the number filled in the attribute shall be identical to the number of ArrayValueSpecification.element. If the attribute does not exist it means that no partial initialization is intended.

Table F.18: ArrayValueSpecification

Class	**AssemblySwConnector**			
Note	`AssemblySwConnector`s are exclusively used to connect `SwComponentPrototype`s in the context of a `CompositionSwComponentType`.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port. **Tags:** atp.Splitkey=provider, provider.contextComponent, provider.targetPPort **InstanceRef implemented by:** PPortInComposition InstanceRef
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port. **Tags:** atp.Splitkey=requester, requester.context Component, requester.targetRPort **InstanceRef implemented by:** RPortInComposition InstanceRef

Table F.19: AssemblySwConnector

Class	***AtomicSwComponentType*** (abstract)			
Note	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Subclasses	ApplicationSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponent Type, NvBlockSwComponentType, SensorActuatorSwComponentType, ServiceProxySwComponent Type, ServiceSwComponentType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
internalBehavior	SwcInternalBehavior	0..1	aggr	The `SwcInternalBehavior`s owned by an `AtomicSwComponentType` can be located in a different physical file. Therefore the aggregation is <<atp Splitable>>. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalBehavior, internalBehavior.short Name, internalBehavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the `SymbolProps` for the `AtomicSwComponentType`. **Tags:** atp.Splitkey=symbolProps, symbolProps.short Name

Table F.20: AtomicSwComponentType

Class	***AtpFeature*** (abstract)			
Note	Features are properties via which a classifier classifies instances. Or: a classifier has features and every M0 instance of it will have those features.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AtpPrototype*, *AtpStructureElement*			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table F.21: AtpFeature

Class	***AtpInstanceRef*** (abstract)			
Note	An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node. An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).			
Base	*ARObject*			
Subclasses	AnyInstanceRef, ApplicationCompositeElementInPortInterfaceInstanceRef, ComponentInComposition InstanceRef, ComponentInSystemInstanceRef, *DataPrototypeInPortInterfaceInstanceRef*, Data PrototypeInSystemInstanceRef, InnerDataPrototypeGroupInCompositionInstanceRef, InnerPortGroupIn CompositionInstanceRef, InnerRunnableEntityGroupInCompositionInstanceRef, InstanceEventIn CompositionInstanceRef, ModeDeclarationGroupPrototypeInSystemInstanceRef, *ModeGroupInAtomic SwcInstanceRef*, ModeInBswInstanceRef, ModeInBswModuleDescriptionInstanceRef, ModeInSwc InstanceRef, OperationArgumentInComponentInstanceRef, *OperationInAtomicSwcInstanceRef*, OperationInSystemInstanceRef, PModeInSystemInstanceRef, ParameterDataPrototypeInSystem InstanceRef, ParameterInAtomicSWCTypeInstanceRef, PortGroupInSystemInstanceRef, *PortIn CompositionTypeInstanceRef*, RModeInAtomicSwcInstanceRef, RteEventInCompositionInstanceRef, RteEventInEcuInstanceRef, RteEventInSystemInstanceRef, RunnableEntityInCompositionInstanceRef, SwcServiceDependencyInSystemInstanceRef, *TriggerInAtomicSwcInstanceRef*, TriggerInSystem InstanceRef, VariableAccessInEcuInstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstanceRef, VariableInAtomicSWCTypeInstanceRef, *VariableInAtomic SwcInstanceRef*, VariableInComponentInstanceRef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
atpBase	AtpClassifier	0..1	ref	This is the base from which the navigaion path starts. **Stereotypes:** atpAbstract; atpDerived
atpContext Element (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. **Stereotypes:** atpAbstract
atpTarget	AtpFeature	0..1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table F.22: AtpInstanceRef

Class	***AutosarDataPrototype*** (abstract)			
Note	Base class for prototypical roles of an `AutosarDataType`.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ArgumentDataPrototype, ParameterDataPrototype, VariableDataPrototype			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	AutosarDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table F.23: AutosarDataPrototype

Class	***AutosarDataType*** (abstract)			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractImplementationDataType*, *ApplicationDataType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType. **Tags:** atp.Splitkey=swDataDefProps

Table F.24: AutosarDataType

Class	***BaseType*** (abstract)			
Note	This abstract meta-class represents the ability to specify a platform dependent base type.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	SwBaseType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseTypeDefinition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table F.25: BaseType

Class	**BaseTypeDirectDefinition**			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Aggregated by	*BaseType*.baseTypeDefinition			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
baseTypeEncoding	BaseTypeEncodingString	0..1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **Tags:** xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. **Tags:** xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. **Tags:** xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". **Tags:** xml.sequenceOffset=100

▽

△

Class	**BaseTypeDirectDefinition**			
native Declaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE. If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size. This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems. **Tags:** xml.sequenceOffset=120

Table F.26: BaseTypeDirectDefinition

Class	**BswImplementation**			
Note	Contains the implementation specific information in addition to the generic specification (BswModule Description and BswBehavior). It is possible to have several different BswImplementations referring to the same BswBehavior. **Tags:** atp.recommendedPackage=BswImplementations This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *Implementation*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arRelease Version	RevisionLabelString	0..1	attr	Version of the AUTOSAR Release on which this implementation is based. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.
behavior	BswInternalBehavior	0..1	ref	The behavior of this implementation. This relation is made as an association because • it follows the pattern of the SWCT • since ARElement cannot be split, but we want supply the implementation later, the BswImplementation is not aggregated in BswBehavior **Tags:** atp.Splitkey=behavior
preconfigured Configuration	EcucModule ConfigurationValues	*	ref	Reference to the set of preconfigured (i.e. fixed) configuration values for this BswImplementation. If the BswImplementation represents a cluster of several modules, more than one EcucModuleConfigurationValues element can be referred (at most one per module), otherwise at most one such element can be referred. **Tags:** atp.Splitkey=preconfiguredConfiguration xml.roleWrapperElement=true

▽

△

Class	**BswImplementation**			
recommended Configuration	EcucModule ConfigurationValues	*	ref	Reference to one or more sets of recommended configuration values for this module or module cluster. **Tags:** atp.Splitkey=recommendedConfiguration
vendorApiInfix	Identifier	0..1	attr	In driver modules which can be instantiated several times on a single ECU, SRS_BSW_00347 requires that the names of files, APIs, published parameters and memory allocation keywords are extended by the vendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific API name is generated as follows: <Module Name>_<vendorId>_ <vendorApiInfix>_<API name from SWS>. E.g. assuming that the vendorId of the implementer is 123 and the implementer chose a vendorApiInfix of "v11r456" an API name Can_Write defined in the SWS will translate to Can_123_v11r456_Write. This attribute is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1. See also SWS_BSW_00102.
vendorSpecific ModuleDef	EcucModuleDef	*	ref	Reference to • the vendor specific EcucModuleDef used in this Bsw Implementation if it represents a single module • several EcucModuleDefs used in this Bsw Implementation if it represents a cluster of modules • one or no EcucModuleDefs used in this Bsw Implementation if it represents a library **Tags:** atp.Splitkey=vendorSpecificModuleDef xml.roleWrapperElement=true

Table F.27: BswImplementation

Class	**BswInternalBehavior**			
Note	Specifies the behavior of a BSW module or a BSW cluster w.r.t. the code entities visible by the BSW Scheduler. It is possible to have several different BswInternalBehaviors referring to the same BswModule Description.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *InternalBehavior*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.internalBehavior			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	Defines an AUTOSAR typed memory-block that needs to be available for each instance of the Basic Software Module. The aggregation of arTypedPerInstanceMemory is subject to variability with the purpose to support variability in the Basic Software Module's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory, arTypedPer InstanceMemory.shortName, arTypedPerInstance Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**BswInternalBehavior**			
bswPerInstance MemoryPolicy	BswPerInstance MemoryPolicy	*	aggr	Policy for a arTypedPerInstanceMemory The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bswPerInstanceMemoryPolicy, bswPer InstanceMemoryPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
clientPolicy	BswClientPolicy	*	aggr	Policy for a requiredClientServerEntry. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=clientPolicy, clientPolicy.variationPoint.short Label vh.latestBindingTime=preCompileTime
distinguished Partition	BswDistinguished Partition	*	aggr	Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=distinguishedPartition, distinguished Partition.shortName, distinguishedPartition.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
entity	BswModuleEntity	*	aggr	A code entity for which the behavior is described **Stereotypes:** atpVariation **Tags:** atp.Splitkey=entity, entity.shortName, entity.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=5
event	BswEvent	*	aggr	An event required by this module behavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=event, event.shortName, event.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
exclusiveArea Policy	BswExclusiveArea Policy	*	aggr	Policy for an ExclusiveArea in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a basic software module for its implementation. **Tags:** atp.Splitkey=includedDataTypeSet
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Tags:** atp.Splitkey=includedModeDeclarationGroupSet

▽

△

Class	**BswInternalBehavior**			
internal TriggeringPoint	BswInternalTriggering Point	*	aggr	An internal triggering point. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint, internalTriggering Point.shortName, internalTriggeringPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=2 This Attribute is only used by the AUTOSAR Classic Platform.
internal TriggeringPoint Policy	BswInternalTriggering PointPolicy	*	aggr	Policy for an internalTriggeringPoint in this BswInternal Behavior.. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPointPolicy, internal TriggeringPointPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeReceiver Policy	BswModeReceiver Policy	*	aggr	Implementation policy for the reception of mode switches. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeReceiverPolicy, modeReceiver Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
modeSender Policy	BswModeSenderPolicy	*	aggr	Implementation policy for providing a mode group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeSenderPolicy, modeSender Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
parameterPolicy	BswParameterPolicy	*	aggr	Policy for a perInstanceParameter in this BswInternal Behavior. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=parameterPolicy, parameterPolicy.variation Point.shortLabel vh.latestBindingTime=preCompileTime
perInstance Parameter	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) needed by this BswInternal Behavior. The role name perInstanceParameter is chosen in analogy to the similar role in the context of SwcInternal Behavior. In contrast to constantMemory, this object is not allocated locally by the module's code, but by the BSW Scheduler and it is accessed from the BSW module via the BSW Scheduler API. The main use case is the support of software emulation of calibration data. The aggregation is subject to variability with the purpose to support implementation variants. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=perInstanceParameter, perInstance ▽

▽

△

Class	**BswInternalBehavior**			
				△ Parameter.shortName, perInstanceParameter.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
receptionPolicy	BswDataReception Policy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=receptionPolicy, receptionPolicy.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
releasedTrigger Policy	BswReleasedTrigger Policy	*	aggr	Policy for a releasedTrigger. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=releasedTriggerPolicy, releasedTrigger Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
schedulerName Prefix	BswSchedulerName Prefix	*	aggr	Optional definition of one or more prefixes to be used for the BswScheduler. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=schedulerNamePrefix, schedulerName Prefix.shortName, schedulerNamePrefix.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50
sendPolicy	BswDataSendPolicy	*	aggr	Policy for a providedData. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sendPolicy, sendPolicy.variationPoint.short Label vh.latestBindingTime=preCompileTime
service Dependency	BswService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item. The aggregation is subject to variability with the purpose to support the conditional existence of ServiceNeeds. The aggregation is splitable in order to support that ServiceNeeds might be provided in later development steps. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceDependency, service Dependency.ident.shortName, service Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
triggerDirect Implementation	BswTriggerDirect Implementation	*	aggr	Specifies a trigger to be directly implemented via OS calls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=triggerDirectImplementation, triggerDirect Implementation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=15

▽

△

Class	**BswInternalBehavior**			
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation. **Tags:** atp.Splitkey=variationPointProxy, variationPoint Proxy.shortName

Table F.28: BswInternalBehavior

Class	**CalibrationParameterValue**			
Note	Specifies instance specific calibration parameter values used to initialize the memory objects implementing calibration parameters in the generated RTE code. RTE generator will use the implInitValue to override the initial values specified for the DataPrototypes of a component type. The applInitValue is used to exchange init values with the component vendor not publishing the transformation algorithm between ApplicationDataTypes and ImplementationDataTypes or defining an instance specific initialization of components which are only defined with ApplicationDataTypes. Note: If both representations of init values are available these need to represent the same content. Note further that in this case an explicit mapping of ValueSpecification is not implemented because calibration parameters are delivered back after the calibration phase.			
Base	*ARObject*			
Aggregated by	CalibrationParameterValueSet.calibrationParameterValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ApplicationDataType **Tags:** atp.Splitkey=applInitValue
implInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ImplementationDataType **Tags:** atp.Splitkey=implInitValue
initialized Parameter	FlatInstanceDescriptor	0..1	ref	This represents the parameter that is initialized by the CalibrationParameterValue. **Tags:** atp.Splitkey=initializedParameter This Attribute is only used by the AUTOSAR Classic Platform.

Table F.29: CalibrationParameterValue

Class	**CanXlFrameTriggeringProps**			
Note	This element indicates the frame being CAN XL and contains further CAN XL specific attributes.			
Base	*ARObject*			
Aggregated by	CanFrameTriggering.canXlFrameTriggeringProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
acceptance Field	PositiveInteger	0..1	attr	Acceptance field of a CAN XL message.
priorityId	PositiveInteger	0..1	attr	Priority ID of a CAN XL message.
sduType	PositiveInteger	0..1	attr	SDU type of a CAN XL message.
vcid	PositiveInteger	0..1	attr	Virtual CAN network ID of a CAN XL message.

Table F.30: CanXlFrameTriggeringProps

Class	ClientServerInterface			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation	ClientServerOperation	*	aggr	`ClientServerOperation`(s) of this `ClientServerInterface`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=operation, operation.shortName, operation.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface. **Tags:** atp.Splitkey=possibleError, possibleError.shortName

Table F.31: ClientServerInterface

Class	ClientServerOperation			
Note	An operation declared within the scope of a client/server interface.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.command, *AtpClassifier*.atpFeature, ClientServerInterface.operation, DiagnosticDataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface.write, DiagnosticExtendedDataRecordInterface.provide, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutineInterface.stop, ServiceInterface.method			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this `ClientServerOperation`. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
diagArgIntegrity	Boolean	0..1	attr	This attribute shall only be used in the implementation of diagnostic routines to support the case where input and output arguments are allocated in a shared buffer and might unintentionally overwrite input arguments by tentative write operations to output arguments. This situation can happen during sliced execution or while output parameters are arrays (call by reference). The value true means that the `ClientServerOperation` is aware of the usage of a shared buffer and takes precautions to avoid unintentional overwrite of input arguments. If the attribute does not exist or is set to false the `ClientServerOperation` does not have to consider the usage of a shared buffer. This Attribute is only used by the AUTOSAR Classic Platform.
possibleError	ApplicationError	*	ref	Possible errors that may by raised by the referring operation. **Tags:** atp.Splitkey=possibleError This Attribute is only used by the AUTOSAR Classic Platform.

Table F.32: ClientServerOperation

Class	Collection			
Note	This meta-class specifies a collection of elements. A collection can be utilized to express additional aspects for a set of elements. Note that Collection is an ARElement. Therefore it is applicable e.g. for EvaluatedVariant, even if this is not obvious. Usually the category of a Collection is "SET". On the other hand, a Collection can also express an arbitrary relationship between elements. This is denoted by the category "RELATION" (see also [TPS_GST_00347]). In this case the collection represents an association from "sourceElement" to "targetElement" in the role "role". **Tags:** atp.recommendedPackage=Collections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autoCollect	AutoCollectEnum	0..1	attr	This attribute reflects how far the referenced objects are part of the collection. **Tags:** xml.sequenceOffset=20
collected Instance	AtpFeature	*	iref	This instance ref supports the use case that a particular instance is part of the collection. **Tags:** atp.Splitkey=collectedInstance, collectedInstance.context Element, collectedInstance.target xml.sequenceOffset=60 **InstanceRef implemented by:** AnyInstanceRef
collection Semantics	NameToken	0..1	attr	Provides the ability to express the semantics of a Collection depending on the intended use case. The collectionSemantics is specified as a NameToken which must be agreed by all stakeholders. **Tags:** xml.sequenceOffset=25
element	Identifiable	*	ref	This is an element in the collection. Note that Collection itself is collectable. Therefore collections can be nested. In case of category="RELATION" this represents the target end of the relation. **Tags:** atp.Splitkey=element xml.sequenceOffset=40
elementRole	Identifier	0..1	attr	This attribute allows to denote a particular role of the collection. Note that the applicable semantics shall be mutually agreed between the two parties. In particular it denotes the role of element in the context of sourceElement. **Tags:** xml.sequenceOffset=30
sourceElement	Identifiable	*	ref	Only if Category = “RELATION”. This represents the source of a relation. **Tags:** atp.Splitkey=sourceElement xml.sequenceOffset=50
sourceInstance	AtpFeature	*	iref	Only if Category = “RELATION”. This represents the source instance of a relation. **Tags:** atp.Splitkey=sourceInstance, sourceInstance.context Element, sourceInstance.target xml.sequenceOffset=70 **InstanceRef implemented by:** AnyInstanceRef

Table F.33: Collection

Class	ComplexDeviceDriverSwComponentType			
Note	The ComplexDeviceDriverSwComponentType is a special AtomicSwComponentType that has direct access to hardware on an ECU and which is therefore linked to a specific ECU or specific hardware. The ComplexDeviceDriverSwComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hardware Element	HwDescriptionEntity	*	ref	Reference from the ComplexDeviceDriverSwComponent Type to the description of the used HwElements. **Tags:** atp.Splitkey=hardwareElement

Table F.34: ComplexDeviceDriverSwComponentType

Class	CompuConstTextContent			
Note	This meta-class represents the textual content of a scale.			
Base	*ARObject*, *CompuConstContent*			
Aggregated by	CompuConst.compuConstContentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
vt	VerbatimString	0..1	attr	This represents a textual constant in the computation method.

Table F.35: CompuConstTextContent

Class	CompuMethod			
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant. **Tags:** atp.recommendedPackage=CompuMethods			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuInternal ToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** atp.Splitkey=compuInternalToPhys xml.sequenceOffset=80
compuPhysTo Internal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:** atp.Splitkey=compuPhysToInternal xml.sequenceOffset=90
displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. **Tags:** xml.sequenceOffset=20

△

Class	**CompuMethod**			
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. **Tags:** atp.Splitkey=unit xml.sequenceOffset=30

Table F.36: CompuMethod

Class	**CompuScale**			
Note	This meta-class represents the ability to specify one segment of a segmented computation method.			
Base	*ARObject*			
Aggregated by	CompuScales.compuScale			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
a2lDisplayText	String	0..1	attr	The value of this attribute shall be taken for generating one display text (specifically the OutVal) within the equivalent of the enclosing `CompuMethod` in A2L.
compuInverse Value	CompuConst	0..1	aggr	This is the inverse value of the constraint. This supports the case that the scale is not reversible per se. **Tags:** atp.Splitkey=compuInverseValue xml.sequenceOffset=60
compuScale Contents	CompuScaleContents	0..1	aggr	This represents the computation details of the scale. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=70 xml.typeElement=false xml.typeWrapperElement=false
desc	MultiLanguageOverview Paragraph	0..1	aggr	<desc> represents a general but brief description of the object in question. **Tags:** atp.Splitkey=desc xml.sequenceOffset=30
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
mask	PositiveUnlimitedInteger	0..1	attr	In difference to all the other computational methods every COMPU-SCALE will be applied including the bit MASK. Therefore it is allowed for this type of COMPU-METHOD, that COMPU-SCALES overlap. To calculate the string reverse to a value, the string has to be split and the according value for each substring has to be summed up. The sum is finally transmitted. The processing has to be done in order of the COMPU-SCALE elements. **Tags:** xml.sequenceOffset=35
shortLabel	Identifier	0..1	attr	This element specifies a short name for the particular scale. The name can for example be used to derive a programming language identifier. **Tags:** xml.sequenceOffset=20

▽

△

Class	CompuScale			
symbol	CIdentifier	0..1	attr	The symbol, if provided, is used by code generators to get a C identifier for the CompuScale. The name will be used as is for the code generation, therefore it needs to be unique within the generation context. **Tags:** xml.sequenceOffset=25
upperLimit	Limit	0..1	attr	This specifies the upper limit of a of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

Table F.37: CompuScale

Class	CompuScaleConstantContents			
Note	This meta-class represents the fact that a particular scale of the computation method is constant.			
Base	*ARObject*, *CompuScaleContents*			
Aggregated by	CompuScale.compuScaleContents			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuConst	CompuConst	0..1	aggr	This represents the fact that the scale is a constant. The use case is mainly a non interpolated scale. It is a simplification of the fact that a constant scale can also be expressed as rational function of order 0. **Tags:** xml.sequenceOffset=90

Table F.38: CompuScaleConstantContents

Class	CompuScales			
Note	This meta-class represents the ability to stepwise express a computation method.			
Base	*ARObject*, *CompuContent*			
Aggregated by	Compu.compuContent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuScale (ordered)	CompuScale	*	aggr	This represents one scale within the compu method. Note that it contains a Variationpoint in order to support blueprints of enumerations. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table F.39: CompuScales

Class	«atpMixedString» **ConditionByFormula**			
Note	This class represents a condition which is computed based on system constants according to the specified expression. The expected result is considered as boolean value. The result of the expression is interpreted as a condition. • "0" represents "false"; • a value other than zero is considered "true"			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Aggregated by	VariationPoint.swSyscond, VariationPointProxy.conditionAccess			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bindingTime	BindingTimeEnum	0..1	attr	This attribute specifies the point in time when condition may be evaluated at earliest. At this point in time all referenced system constants shall have a value. **Tags:** xml.attribute=true

Table F.40: ConditionByFormula

Enumeration	**CouplingPortRoleEnum**
Note	Defines the role a CouplingPort takes in the context of a CouplingElement.
Aggregated by	CouplingPort.couplingPortRole
Literal	***Description***
hostPort	The hostPort is connected to an ECU (host ecu). The host ECU controls the connected Coupling Element (e.g. Ethernet switch). **Tags:** atp.EnumerationLiteralIndex=0
standardPort	A CoupingPort can be a standardPort that is used to connect the CouplingElement with Coupling Ports outside the ECU. **Tags:** atp.EnumerationLiteralIndex=2
upLinkPort	A CouplingPort can be connected to another CouplingPort of a CouplingElement located on the same ECU (CouplingElement.ecuInstance) using the CouplingPortConnection. This is used to model a cascaded switch. **Tags:** atp.EnumerationLiteralIndex=1

Table F.41: CouplingPortRoleEnum

Class	**DataConstr**			
Note	This meta-class represents the ability to specify constraints on data. **Tags:** atp.recommendedPackage=DataConstrs			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. **Tags:** atp.Splitkey=dataConstrRule xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table F.42: DataConstr

Class	**DataPrototype** (abstract)			
Note	Base class for prototypical roles of any data type.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*ApplicationCompositeElementDataPrototype*, *AutosarDataPrototype*			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level. **Tags:** atp.Splitkey=swDataDefProps

Table F.43: DataPrototype

Class	**DataPrototypeInSystemInstanceRef**			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	DiagnosticEdrSenderPortMapping.dataPrototype, DiagnosticEnvDataElementCondition.dataPrototype, DiagnosticServiceDataMapping.mappedDataElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	System	0..1	ref	This represents the base of the InstanceRef **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:** xml.sequenceOffset=30
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=50
contextPort	PortPrototype	0..1	ref	This represents the PortPrototype that is contained in the InstanceRef. **Tags:** xml.sequenceOffset=40
contextRoot Composition	RootSwComposition Prototype	0..1	ref	**Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	**Tags:** xml.sequenceOffset=45
targetData Prototype	DataPrototype	0..1	ref	This represents the target of the InstanceRef **Tags:** xml.sequenceOffset=60

Table F.44: DataPrototypeInSystemInstanceRef

Class	**DataPrototypeMapping**
Note	Defines the mapping of two particular `VariableDataPrototype`s, `ParameterDataPrototype`s or `ArgumentDataPrototype`s with non-equal `shortName`s, non-equal structure (specific condition is described by [constr_1187]), and/or non-equal semantic (resolution or range) in context of two different `SenderReceiverInterface`, `NvDataInterface` or `ParameterInterface` or Operations. If the semantic is unequal, the following rules apply: The `textTableMapping` is only applicable if the referred `DataPrototype`s are typed by `AutosarDataType` referring to `CompuMethod`s of `category` `TEXTTABLE`, `SCALE_LINEAR_AND_TEXTTABLE` or `BITFIELD_TEXTTABLE`. In the case that the `DataPrototype`s are typed by `AutosarDataType` either referring to `CompuMethod`s of `category` `LINEAR`, `IDENTICAL` or referring to no `CompuMethod` (which is similar as `IDENTICAL`) the linear conversion factor is calculated out of the `factorSiToUnit` and `offsetSiToUnit` attributes of the referred `Unit`s and the `CompuRationalCoeffs` of a `compuInternalToPhys` of the referred `CompuMethod`s.

▽

△

Class	**DataPrototypeMapping**			
Base	*ARObject*			
Aggregated by	ClientServerOperationMapping.argumentMapping, VariableAndParameterInterfaceMapping.dataMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstData Prototype	AutosarDataPrototype	0..1	ref	First to be mapped DataPrototype in context of a Sender ReceiverInterface, NvDataInterface, ParameterInterface or Operation. **Tags:** atp.Splitkey=firstDataPrototype
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference defines the need to execute the Data Transformation <Mip>_<transformerId> functions of the transformation chain when communicating from the Data PrototypeMapping.firstDataPrototype to the Data PrototypeMapping.secondDataPrototype. This reference also specifies the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain (i.e. from the DataPrototype Mapping.secondDataPrototype to the DataPrototype Mapping.firstDataPrototype) if the referenced Data Transformation is symmetric, i.e. attribute Data Transformation.dataTransformationKind is set to symmetric. **Tags:** atp.Splitkey=firstToSecondDataTransformation
secondData Prototype	AutosarDataPrototype	0..1	ref	Second to be mapped DataPrototype in context of a SenderReceiverInterface, NvDataInterface, Parameter Interface or Operation. **Tags:** atp.Splitkey=secondDataPrototype
secondToFirst Data Transformation	DataTransformation	0..1	ref	This defines the need to execute the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain when communicating from the DataPrototypeMapping.secondDataPrototype to the Data PrototypeMapping.firstDataPrototype. **Tags:** atp.Splitkey=secondToFirstDataTransformation
subElement Mapping	SubElementMapping	*	aggr	This represents the owned SubelementMapping. **Tags:** atp.Splitkey=subElementMapping
textTable Mapping	TextTableMapping	*	aggr	Applied TextTableMapping(s) **Tags:** atp.Splitkey=textTableMapping

Table F.45: DataPrototypeMapping

Enumeration	**DataTransformationErrorHandlingEnum**
Note	This enumeration defines different ways how a RunnableEntity shall handle transformer errors.
Aggregated by	PortAPIOption.errorHandling
Literal	***Description***
noTransformerError Handling	A runnable does not handle transformer errors. **Tags:** atp.EnumerationLiteralIndex=0
transformerError Handling	The runnable implements the handling of transformer errors. **Tags:** atp.EnumerationLiteralIndex=1

Table F.46: DataTransformationErrorHandlingEnum

Enumeration	**DataTransformationKindEnum**
Note	This enumeration contributes to the definition of the scope of the DataTransformation.
Aggregated by	DataTransformation.dataTransformationKind
Literal	***Description***
asymmetricFrom ByteArray	The DataTransformation shall only be applied to the receiving end only, i.e. transform from byte array to data type. **Tags:** atp.EnumerationLiteralIndex=0
asymmetricToByte Array	The DataTransformation shall be applied to the sending end only, i.e. from data type to byte array. **Tags:** atp.EnumerationLiteralIndex=1
symmetric	The DataTransformation shall be applied at both the sending and the receiving end of the communication. **Tags:** atp.EnumerationLiteralIndex=2

Table F.47: DataTransformationKindEnum

Enumeration	**DataTransformationStatusForwardingEnum**
Note	This enumeration defines different ways how a RunnableEntity shall be able to forward status code into the transformer chain.
Aggregated by	PortAPIOption.transformerStatusForwarding
Literal	***Description***
noTransformer StatusForwarding	The RunnableEntity is not able to forward a transformer status code. **Tags:** atp.EnumerationLiteralIndex=0
transformerStatus Forwarding	The RunnableEntity is able to forward a transformer status code. **Tags:** atp.EnumerationLiteralIndex=1

Table F.48: DataTransformationStatusForwardingEnum

Class	**DataTypeMap**			
Note	This class represents the relationship between `ApplicationDataType` and its implementing `AbstractImplementationDataType`.			
Base	*ARObject*			
Aggregated by	DataTypeMappingSet.dataTypeMap			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
applicationData Type	ApplicationDataType	0..1	ref	This is the corresponding `ApplicationDataType` **Tags:** atp.Splitkey=applicationDataType
implementation DataType	AbstractImplementation DataType	0..1	ref	This is the corresponding `AbstractImplementationDataType`. **Tags:** atp.Splitkey=implementationDataType

Table F.49: DataTypeMap

Class	**DataTypeMappingSet**
Note	This class represents a list of mappings between `ApplicationDataType`s and `ImplementationDataType`s. In addition, it can contain mappings between `ImplementationDataType`s and `ModeDeclarationGroup`s. **Tags:** atp.recommendedPackage=DataTypeMappingSets
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**DataTypeMappingSet**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an `ApplicationDataType` and its `AbstractImplementationDataType`. **Tags:** atp.Splitkey=dataTypeMap
modeRequest TypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an `ModeDeclarationGroup` and its `AbstractImplementationDataType`. **Tags:** atp.Splitkey=modeRequestTypeMap

Table F.50: DataTypeMappingSet

Class	**DelegationSwConnector**			
Note	A delegation connector delegates one inner `PortPrototype` (a port of a component that is used inside the composition) to a outer `PortPrototype` of compatible type that belongs directly to the composition (a port that is owned by the composition).			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
innerPort	PortPrototype	0..1	iref	The port that belongs to the ComponentPrototype in the composition **Tags:** atp.Splitkey=innerPort, innerPort.context Component, innerPort.targetPPort, innerPort.targetRPort **InstanceRef implemented by:** PortInCompositionType InstanceRef
outerPort	PortPrototype	0..1	ref	The port that is located on the outside of the Composition Type **Tags:** atp.Splitkey=outerPort

Table F.51: DelegationSwConnector

Class	***Describable*** (abstract)			
Note	This meta-class represents the ability to add a descriptive documentation to non identifiable elements.			
Base	*ARObject*			
Subclasses	CyclicTiming, EventControlledTiming, FunctionGroupErrorMapping, HwElementConnector, HwPin Connector, HwPinGroupConnector, IPduTiming, Ipv4DhcpServerConfiguration, Ipv6DhcpServer Configuration, PncMapping, *TransformationComSpecProps*, *TransformationDescription*, *Transformation ISignalProps*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data for the describable object. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-20
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Describable. It affects the expected existence of attributes and the applicability of constraints. **Tags:** xml.sequenceOffset=-50

▽

△

Class	***Describable*** (abstract)			
desc	MultiLanguageOverview Paragraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". **Tags:** atp.Splitkey=desc xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-30

Table F.52: Describable

Class	**DltApplication**			
Note	This meta-class represents the application from which the log and trace message originates.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DltEcu.application			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
application Description	String	0..1	attr	This attribute can be used to describe the applicationId that is used in the log and trace message in more detail.
applicationId	String	0..1	attr	This attribute identifies the SW-C/BSW module in the log and trace message.
context	DltContext	*	ref	Definition of ContextIds for the Application. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=context, context.dltContext, context.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table F.53: DltApplication

Class	**DltContext**			
Note	This meta-class represents the Context that groups Log and Trace messages that are generated by an application. **Tags:** atp.recommendedPackage=DltContexts			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
context Description	String	0..1	attr	This attribute can be used to describe the contextId that is used in the log and trace message in more detail.
contextId	String	0..1	attr	This attribute is used to group log and trace messages produced by an application to distinguish functionality.

▽

△

Class	DltContext			
dltMessage	DltMessage	*	ref	Group of Log and Trace messages assigned to the Dlt Context **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dltMessage, dltMessage.dltMessage, dltMessage.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table F.54: DltContext

Class	DltEcu			
Note	This element represents an Ecu or Machine that produces logging and tracing information. **Tags:** atp.recommendedPackage=DltEcus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application	DltApplication	*	aggr	Application on DltEcu that provides log or trace data. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=application, application.shortName, application.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
ecuId	String	0..1	attr	This attribute defines the name of the ECU for use within the Dlt protocol.

Table F.55: DltEcu

Class	DltMessage			
Note	This element defines a DltMessage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LogAndTraceMessageCollectionSet.dltMessage			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dltArgument (ordered)	DltArgument	*	aggr	Ordered collection of DltArguments in the DltMessage.
messageId	PositiveInteger	0..1	attr	This attribute defines the unique Id for the DltMessage.
messageLine Number	PositiveInteger	0..1	attr	This attribute describes the position in the source file in which this log message was called.
messageSource File	String	0..1	attr	This attribute describes the source file in which this log message was called.
messageType Info	String	0..1	attr	This attribute describes the message Type
privacyLevel	PrivacyLevel	0..1	aggr	The Privacy Level helps to identify the Log and Trace content towards the degree of privacy to it. **Tags:** atp.Splitkey=privacyLevel

Table F.56: DltMessage

Class	DoIpActivationLineNeeds			
Note	A DoIP entity needs to be informed when an external tester is attached or activated. The DoIpActivation ServiceNeeds specifies the trigger for such an event. Examples would be a Pdu via a regular communication bus, a PWM signal, or an I/O. For details please refer to the ISO 13400.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table F.57: DoIpActivationLineNeeds

Class	DoIpGidNeeds			
Note	The DoIpGidNeeds indicates that the software-component owning this ServiceNeeds is providing the GID number either after a GID Synchronisation or by other means like e.g. flashed EEPROM parameter. This need can be used independent from DoIpGidSynchronizationNeeds and is necessary if the GID can not be provided out of the DoIP configuration options.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table F.58: DoIpGidNeeds

Class	DoIpGidSynchronizationNeeds			
Note	The DoIpGidSynchronizationNeeds indicates that the software-component owning this ServiceNeeds is triggered by the DoIP entity to start a synchronization of the GID (Group Identification) on the DoIP service 0x0001, 0x0002, 0x0003 or before announcement via service 0x0004 according to ISO 13400-2:2012 if necessary. Note that this need is only relevant for DoIP synchronization masters.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table F.59: DoIpGidSynchronizationNeeds

Class	DoIpPowerModeStatusNeeds			
Note	The DoIpPowerModeStatusNeeds indicates that the software-component owning this ServiceNeeds is providing the PowerModeStatus for the DoIP service 0x4003 according to ISO 13400-2:2012.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table F.60: DoIpPowerModeStatusNeeds

Class	***DoIpServiceNeeds*** (abstract)			
Note	This represents an abstract base class for ServiceNeeds related to DoIP.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Subclasses	DoIpActivationLineNeeds, DoIpGidNeeds, DoIpGidSynchronizationNeeds, DoIpPowerModeStatus Needs, DoIpRoutingActivationAuthenticationNeeds, DoIpRoutingActivationConfirmationNeeds, Further ActionByteNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.61: DoIpServiceNeeds

Class	**EcuAbstractionSwComponentType**			
Note	The ECUAbstraction is a special AtomicSwComponentType that resides between a software-component that wants to access ECU periphery and the Microcontroller Abstraction. The EcuAbstractionSw ComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hardware Element	HwDescriptionEntity	*	ref	Reference from the EcuAbstractionComponentType to the description of the used HwElements. **Tags:** atp.Splitkey=hardwareElement

Table F.62: EcuAbstractionSwComponentType

Class	***EcucContainerDef*** (abstract)			
Note	Base class used to gather common attributes of configuration container definitions. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AtpDefinition*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	EcucChoiceContainerDef, EcucParamConfContainerDef			
Aggregated by	EcucDestinationUriPolicy.container, EcucModuleDef.container, EcucParamConfContainerDef.sub Container			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
destinationUri	EcucDestinationUriDef	*	ref	Several destinationUris can be defined for an Ecuc ContainerDef. With such destinationUris an Ecuc ContainerDef is applicable for several EcucUriReference Defs. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=destinationUri
linkTime Support	Boolean	0..1	attr	Indicates whether the container multiplicity can be changed at LinkTime. Please note that Link-Time is only supported in the Vendor Specific Module Definition (VSMD).

▽

△

Class	***EcucContainerDef*** (abstract)			
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies which MultiplicityConfigurationClass this container is available for which ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_ DEFINITION. If the category attribute of the EcucModule Def is set to VENDOR_SPECIFIC_MODULE_ DEFINITION and if the upperMultiplicity is greater than the lowerMultiplicity then this aggregation is mandatory. **Tags:** atp.Splitkey=multiplicityConfigClass atp.Status=obsolete xml.namePlural=MULTIPLICITY-CONFIG-CLASSES
origin	String	0..1	attr	This attribute specifies whether this configuration container is an AUTOSAR standardized container or whether it is vendor-specific.
postBuild Loadable Multiplicity	Boolean	0..1	attr	Indicates whether the container multiplicity can be changed at post-build loadable time by updating the configuration binary. TRUE means yes, FALSE means no.
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a container may have different number of instances in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
requiresIndex	Boolean	0..1	attr	Used to define whether the value element for this definition shall be provided with an index.

Table F.63: EcucContainerDef

Class	**EcucContainerValue**			
Note	Represents a Container definition in the ECU Configuration Description. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *EcucIndexableValue*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucContainerValue.subContainer, EcucModuleConfigurationValues.container			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
definition	EcucContainerDef	0..1	ref	Reference to the definition of this Container in the ECU Configuration Parameter Definition. **Tags:** atp.Splitkey=definition xml.sequenceOffset=-10
parameterValue	EcucParameterValue	*	aggr	Aggregates all ECU Configuration Values within this Container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=parameterValue, parameterValue.variation Point.shortLabel vh.latestBindingTime=postBuild
referenceValue	EcucAbstractReference Value	*	aggr	Aggregates all References with this container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=referenceValue, referenceValue.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	EcucContainerValue			
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=subContainer, subContainer.shortName, subContainer.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table F.64: EcucContainerValue

Class	EcucValueCollection			
Note	This represents the anchor point of the ECU configuration description. **Tags:** atp.recommendedPackage=EcucValueCollections This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecucValue	EcucModule ConfigurationValues	*	ref	References to the configuration of individual software modules that are present on this ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecucValue, ecucValue.ecucModule ConfigurationValues, ecucValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ecuExtract	System	0..1	ref	Represents the extract of the System Configuration that is relevant for the ECU configured with that ECU Configuration Description. **Tags:** atp.Splitkey=ecuExtract

Table F.65: EcucValueCollection

Class	EndToEndTransformationComSpecProps			
Note	The class EndToEndTransformationIComSpecProps specifies port specific configuration properties for EndToEnd transformer attributes.			
Base	*ARObject*, *Describable*, *TransformationComSpecProps*			
Aggregated by	ClientComSpec.transformationComSpecProps, *ReceiverComSpec*.transformationComSpecProps, ServerComSpec.transformationComSpecProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clearFromValid ToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
disableEndTo EndCheck	Boolean	0..1	attr	Disables/Enables the E2E check. The E2Eheader is removed from the payload independent from the setting of this attribute.
disableEndTo EndState Machine	Boolean	0..1	attr	Disables the E2EStateMachine (only E2E check functionality is performed)
e2eProfile Compatibility Props	E2EProfileCompatibility Props	0..1	ref	Reference to additional settings for the E2E state machine. **Tags:** atp.Splitkey=e2eProfileCompatibilityProps

▽

△

Class	**EndToEndTransformationComSpecProps**			
maxDelta Counter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and Max DeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorState Init	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INIT. The minimum value is 0.
maxErrorState Invalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INVALID. The minimum value is 0.
maxErrorState Valid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_VALID. The minimum value is 0.
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers. This Attribute is only used by the AUTOSAR Classic Platform.
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT. The minimum value is 1.
minOkState Invalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID. The minimum value is 1.
minOkState Valid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID. The minimum value is 1.
syncCounterInit	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers. This Attribute is only used by the AUTOSAR Classic Platform.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.
windowSize Invalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSize Valid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table F.66: EndToEndTransformationComSpecProps

Class	EvaluatedVariantSet			
Note	This meta class represents the ability to express if a set of ARElements is able to support one or more particular variants. In other words, for a given set of evaluatedElements this meta class represents a table of evaluated variants, where each PredefinedVariant represents one column. In this column each descendant sw SystemconstantValue resp. postbuildVariantCriterionValue represents one entry. In a graphical representation each swSystemconstantValueSet / postBuildVariantCriterionValueSet could be used as an intermediate headline in the table column. If the approvalStatus is "APPROVED" it expresses that the collection of CollectableElements is known be valid for the given evaluatedVariants. Note that the EvaluatedVariantSet is a CollectableElement. This allows to establish a hierarchy of EvaluatedVariantSets. **Tags:** atp.recommendedPackage=EvaluatedVariantSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
approvalStatus	NameToken	0..1	attr	Defines the approval status of a predefined variant. Two values are predefined: "APPROVED" and "REJECTED": • Approved variants are known to work. • Rejected variants are known NOT to work. Further values can be approved on a per-company basis; within AUTOSAR only "APPROVED" and "REJECTED" should be recognized.
evaluated Element	CollectableElement	*	ref	This represents a particular element which is evaluated in context of the EvaluatedVariants. The approvalStatus applies to this element (and all of its descendants). In other words, the referenced elements are those that were considered when the predefined variant was evaluated. **Tags:** atp.Splitkey=evaluatedElement
evaluated Variant	PredefinedVariant	*	ref	This metaclass represents one particular variant which was evaluated. LowerMultiplicity is set to 0 to support a stepwise approach. **Tags:** atp.Splitkey=evaluatedVariant

Table F.67: EvaluatedVariantSet

Class	***ExecutableEntity*** (abstract)			
Note	Abstraction of executable code.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*BswModuleEntity*, RunnableEntity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
activation Reason	ExecutableEntity ActivationReason	*	aggr	If the `ExecutableEntity` provides at least one `activationReason` element the RTE resp. BSW Scheduler shall provide means to read the activation vector of this executable entity execution. If no `activationReason` element is provided the feature of being able to determine the activating RTEEvent is disabled for this `ExecutableEntity`. **Tags:** atp.Splitkey=activationReason, activation Reason.shortName



△

Class	***ExecutableEntity*** (abstract)			
canEnter	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=canEnter, canEnter.exclusiveArea, canEnter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity. **Tags:** atp.Splitkey=exclusiveAreaNestingOrder
minimumStartInterval	TimeValue	0..1	attr	Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.
reentrancyLevel	ReentrancyLevelEnum	0..1	attr	The reentrancy level of this ExecutableEntity. See the documentation of the enumeration type ReentrancyLevelEnum for details. Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.
runsInside	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=runsInside, runsInside.exclusiveArea, runsInside.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this code entity. Via an association to the same SwAddrMethod, it can be specified that several code entities (even of different modules or components) shall be located in the same memory without already specifying the memory section itself. **Tags:** atp.Splitkey=swAddrMethod

Table F.68: ExecutableEntity

Class	***ExecutionTime*** (abstract)			
Note	Base class for several means how to describe the ExecutionTime of software. The required context information is provided through this class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AnalyzedExecutionTime, MeasuredExecutionTime, RoughEstimateOfExecutionTime, SimulatedExecutionTime			
Aggregated by	ResourceConsumption.executionTime			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
exclusiveArea	ExclusiveArea	0..1	ref	Reference to the ExclusiveArea this execution time is provided for. **Tags:** atp.Splitkey=exclusiveArea
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this execution time is described. **Tags:** atp.Splitkey=executableEntity

▽

△

Class	***ExecutionTime*** (abstract)			
hardware Configuration	HardwareConfiguration	0..1	aggr	Provides information on the HardwareConfiguration used to specify this ExecutionTime. **Tags:** atp.Splitkey=hardwareConfiguration
hwElement	HwElement	0..1	ref	The hardware element (e.g. type of ECU) for which the execution time is specified. **Tags:** atp.Splitkey=hwElement
includedLibrary	DependencyOnArtifact	*	ref	If this dependency is specified, the execution time of the library code is included in the execution time data for the runnable. **Tags:** atp.Splitkey=includedLibrary
memorySection Location	MemorySectionLocation	*	aggr	Provides information on the MemorySectionLocation which is involved in the ExecutionTime description. **Tags:** atp.Splitkey=memorySectionLocation
softwareContext	SoftwareContext	0..1	aggr	Provides information on the detailed SoftwareContext used to provide the ExecutionTime description. **Tags:** atp.Splitkey=softwareContext

Table F.69: ExecutionTime

Class	***FibexElement*** (abstract)			
Note	ASAM FIBEX elements specifying Communication and Topology.			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*BusMirrorChannelMapping*, *CommunicationCluster*, ConsumedProvidedServiceInstanceGroup, CouplingElement, EcuInstance, EthernetWakeupSleepOnDatalineConfigSet, *Frame*, Gateway, Global TimeDomain, ISignal, ISignalGroup, ISignalIPduGroup, NmConfig, *Pdu*, PdurIPduGroup, Secure CommunicationPropsSet, ServiceInstanceCollectionSet, SocketConnectionIpduIdentifierSet, *TpConfig*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.70: FibexElement

Class	***HeapUsage*** (abstract)			
Note	Describes the heap memory usage of a SW-Component.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	MeasuredHeapUsage, RoughEstimateHeapUsage, WorstCaseHeapUsage			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hardware Configuration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this heap usage is describing. **Tags:** atp.Splitkey=hardwareConfiguration This Attribute is only used by the AUTOSAR Classic Platform.
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this heap usage usage is given. **Tags:** atp.Splitkey=hwElement This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	***HeapUsage*** (abstract)			
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this heap usage is provided for. **Tags:** atp.Splitkey=softwareContext This Attribute is only used by the AUTOSAR Classic Platform.

Table F.71: HeapUsage

Class	**HwElement**			
Note	This represents the ability to describe Hardware Elements on an instance level. The particular types of hardware are distinguished by the category. This category determines the applicable attributes. The possible categories and attributes are defined in HwCategory. **Tags:** atp.recommendedPackage=HwElements			
Base	*ARElement*, *ARObject*, *CollectableElement*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hwElement Connection	HwElementConnector	*	aggr	This represents one particular connection between two hardware elements. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwElementConnection, hwElement Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=110
hwPinGroup	HwPinGroup	*	aggr	This aggregation is used to describe the connection facilities of a hardware element. Note that hardware element has no pins but only pingroups. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=hwPinGroup, hwPinGroup.shortName, hw PinGroup.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=90
nestedElement	HwElement	*	ref	This association is used to establish hierarchies of hw elements. Note that one particular HwElement can be target of this association only once. I.e. multiple instantiation of the same HwElement is not supported (at any hierarchy level). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nestedElement, nestedElement.hwElement, nestedElement.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=70

Table F.72: HwElement

Class	**HwPinGroup**
Note	This meta-class represents the ability to describe groups of pins which are used to connect hardware elements. This group acts as a bundle of pins. Thereby they allow to describe high level connections. Pin groups can even be nested.
Base	*ARObject*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	HwPinGroup			
Aggregated by	HwElement.hwPinGroup, HwPinGroupContent.hwPinGroup			
Attribute	Type	Mult.	Kind	Note
hwPinGroup Content	HwPinGroupContent	0..1	aggr	This aggregation describes the contained pins/pin groups. **Tags:** atp.Splitkey=hwPinGroupContent

Table F.73: HwPinGroup

Class	**Identifiable** (abstract)
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*
Subclasses	ARPackage, *AbstractDoIpLogicAddressProps*, *AbstractEvent*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstanceFilter*, *AbstractServiceInstance*, AppOsTask ProxyToEcuTaskProxyMapping, ApplicationEndpoint, ApplicationError, ApplicationPartitionToEcuPartition Mapping, AppliedStandard, AsynchronousServerCallResultPoint, *AtpBlueprint*, *AtpClassifier*, *Atp Feature*, AutosarOperationArgumentInstance, AutosarVariableInstance, *BinaryManifestAddressable Object*, BinaryManifestItemDefinition, *BinaryManifestResource*, BinaryManifestResourceDefinition, Block State, BswInternalTriggeringPoint, BswModuleDependency, *BuildActionEntity*, BuildActionEnvironment, CanSecSecureChannel, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClientIdDefinition, Client ServerOperation, Code, *CollectableElement*, ComManagementMapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEvent Group, *CouplingElementAbstractDetails*, CouplingPort, *CouplingPortAbstractShaper*, *CouplingPort StructuralElement*, *CpSoftwareClusterResource*, CpSoftwareClusterResourceToApplicationPartition Mapping, CpSoftwareClusterToApplicationPartitionMapping, CpSoftwareClusterToEcuInstanceMapping, CpSoftwareClusterToResourceMapping, *CryptoServiceMapping*, CyclicHandlingComDataToOsTask ProxyMapping, DataPrototypeGroup, DataPrototypeTransformationPropsIdent, DataTransformation, *Dds AbstractServiceInstanceElementCp*, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, Diagnostic ExtendedDataRecordElement, DiagnosticFunctionInhibitSource, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, DltArgumentProps, DltLogChannel, Dlt Message, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, ECUMapping, *EOCExecutableEntity RefAbstract*, EcuPartition, EcuPartitionToCoreMapping, EcucContainerValue, *EcucDefinitionElement*, EcucDestinationUriDef, EcucEnumerationLiteralDef, EcucQuery, EcucValidationCondition, Ethernet WakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeature Relation, FMFeatureRestriction, FMFeatureSelection, FlatInstanceDescriptor, FlexrayTpConnection Control, FlexrayTpNode, FlexrayTpPduPool, *FrameTriggering*, GlobalTimeGateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, *IEEE1722 TpAcfBus*, *IEEE1722TpAcfBusPart*, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignal Triggering, *IdentCaption*, ImpositionTime, InternalTriggeringPoint, J1939Node, J1939SharedAddress Cluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacAddress VlanMembership, MacMulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, Mode Declaration, ModeDeclarationMapping, ModeSwitchPoint, ModeSwitchSenderComSpecProps, Network Endpoint, *NmCluster*, NmEcu, *NmNode*, NvBlockDescriptor, *PackageableElement*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, *Physical Channel*, PortElementToCommunicationResourceMapping, PortGroup, *PortInterfaceMapping*, Queued ReceiverComSpecProps, ResourceConsumption, RootSwCompositionPrototype, RptComponent, Rpt Container, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptService Point, RteEventInCompositionSeparation, RteEventInCompositionToOsTaskProxyMapping, RteEventIn SystemSeparation, RteEventInSystemToOsTaskProxyMapping, RunnableEntityGroup, *SdgAttribute*, Sdg Class, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunication FreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, *ServerCallPoint*, ServerComSpecProps, *ServiceNeeds*, SignalServiceTranslationElementProps, SignalServiceTranslation EventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, *StackUsage*, Static SocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplicationPartitionMapping, SwcToEcuMapping, SwcToImplMapping, SwitchAsynchronousTraffic ShaperGroupEntry, SwitchAtsInstanceEntry, SwitchFlowMeteringEntry, SwitchStreamFilterActionDest ▽

▽

△

Class	***Identifiable*** (abstract)			
	△ PortModification, SwitchStreamFilterEntry, SwitchStreamFilterRule, SwitchStreamGateEntry, SwitchStreamIdentification, SystemMapping, SystemSignalGroupToCommunicationResourceMapping, SystemSignalToCommunicationResourceMapping, TDCpSoftwareClusterMapping, TDCpSoftwareClusterResourceMapping, TcpOptionFilterList, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, TimingModeInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, TraceableTable, TraceableText, *TracedFailure*, TransformationISignalPropsIdent, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, WaitPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-40
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. **Tags:** xml.sequenceOffset=-50
desc	MultiLanguageOverviewParagraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". **Tags:** atp.Splitkey=desc xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-30
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is ▽

▽

△

Class	Identifiable (abstract)			
				△ "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp. **Tags:** xml.attribute=true

Table F.74: Identifiable

Class	Implementation (abstract)			
Note	Description of an implementation a single software component or module. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	BswImplementation, SwcImplementation			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
buildAction Manifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=buildActionManifest, buildAction Manifest.buildActionManifest, buildAction Manifest.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	*	aggr	Specifies the provided implementation code. **Tags:** atp.Splitkey=codeDescriptor, codeDescriptor.short Name
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released **Tags:** atp.Splitkey=compiler, compiler.shortName
generated Artifact	DependencyOnArtifact	*	aggr	Relates to an artifact that will be generated during the integration of this Implementation by an associated generator tool. Note that this is an optional information since it might not always be in the scope of a single module or component to provide this information. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=generatedArtifact, generatedArtifact.short Name, generatedArtifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
hwElement	HwElement	*	ref	The hardware elements (e.g. the processor) required for this implementation. **Tags:** atp.Splitkey=hwElement
linker	Linker	*	aggr	Specifies the linker for which this implementation has been released. **Tags:** atp.Splitkey=linker, linker.shortName
mcSupport	McSupportData	0..1	aggr	The measurement & calibration support data belonging to this implementation. The measurement & calibration support data belonging to this implementation. **Tags:** atp.Splitkey=mcSupport
programming Language	Programminglanguage Enum	0..1	attr	Programming language the implementation was created in.

▽

△

Class	***Implementation*** (abstract)			
requiredArtifact	DependencyOnArtifact	*	aggr	Specifies that this Implementation depends on the existence of another artifact (e.g. a library). This aggregation of DependencyOnArtifact is subject to variability with the purpose to support variability in the implementations. Different algorithms in the implementation might cause different dependencies, e.g. the number of used libraries. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredArtifact, requiredArtifact.shortName, requiredArtifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
required GeneratorTool	DependencyOnArtifact	*	aggr	Relates this Implementation to a generator tool in order to generate additional artifacts during integration. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=requiredGeneratorTool, requiredGeneratorTool.shortName, requiredGeneratorTool.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
resource Consumption	ResourceConsumption	0..1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class. **Tags:** atp.Splitkey=resourceConsumption, resourceConsumption.shortName
swcBsw Mapping	SwcBswMapping	0..1	ref	This allows a mapping between an SWC and a BSW behavior to be attached to an implementation description (for AUTOSAR Service, ECU Abstraction and Complex Driver Components). It is up to the methodology to define whether this reference has to be set for the Swc- or Bsw Implementtion or for both. **Tags:** atp.Splitkey=swcBswMapping
swVersion	RevisionLabelString	0..1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.
usedCode Generator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	0..1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

Table F.75: Implementation

Class	**ImplementationDataType**			
Note	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code. **Tags:** atp.recommendedPackage=ImplementationDataTypes			
Base	*ARElement*, *ARObject*, *AbstractImplementationDataType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Autosar DataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.

▽

△

Class	**ImplementationDataType**			
isStructWith Optional Element	Boolean	0..1	attr	This attribute is only valid if the attribute category is set to STRUCTURE. If set to true, this attribute indicates that the ImplementationDataType has been created with the intention to define at least one element of the structure as optional.
subElement (ordered)	ImplementationData TypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of `ImplementationDataTypeElement` is subject to variability with the purpose to support the conditional existence of elements inside a `ImplementationDataType` representing a structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the Implementation DataType. **Tags:** atp.Splitkey=symbolProps, symbolProps.short Name
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

Table F.76: ImplementationDataType

Class	**ImplementationDataTypeElement**			
Note	Declares a data object which is locally aggregated. Such an element can only be used within the scope where it is aggregated. This element either consists of further `subElement`s or it is further defined via its `swDataDefProps`. There are several use cases within the system of `ImplementationDataType`s for such a local declaration: • It can represent the elements of an array, defining the element type and array size • It can represent an element of a struct, defining its type • It can be the local declaration of a debug element.			
Base	*ARObject*, *AbstractImplementationDataTypeElement*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ImplementationDataType.subElement, ImplementationDataTypeElement.sub Element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
arrayImplPolicy	ArrayImplPolicyEnum	0..1	attr	This attribute controls the implementation of the payload of an array. It shall only be used if the enclosing `ImplementationDataType` constitutes an array.
arraySize	PositiveInteger	0..1	attr	The existence of this attributes (if bigger than 0) defines the size of an array and declares that this `ImplementationDataTypeElement` represents the type of each single array element. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled in case of a variable size array.
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the meaning of the value of the array size.

▽

△

Class	**ImplementationDataTypeElement**			
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing `ImplementationDataTypeElement` as optional. This means that, at runtime, the `ImplementationDataTypeElement` may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the `CppImplementationDataTypeElement` as not valid at the sending end of a communication and determine its validity at the receiving end.
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs"). The aggregation of `ImplementationDataTypeElement` is subject to variability with the purpose to support the conditional existence of elements inside a `ImplementationDataType` representing a structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement. **Tags:** atp.Splitkey=swDataDefProps

Table F.77: ImplementationDataTypeElement

Class	***ImplementationProps*** (abstract)			
Note	Defines a symbol to be used as (depending on the concrete case) either a complete replacement or a prefix when generating code artifacts.			
Base	*ARObject*, *Referrable*			
Subclasses	BswSchedulerNamePrefix, ExecutableEntityActivationReason, SectionNamePrefix, SymbolProps, SymbolicNameProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
symbol	CIdentifier	0..1	attr	The symbol to be used as (depending on the concrete case) either a complete replacement or a prefix.

Table F.78: ImplementationProps

Class	***InternalBehavior*** (abstract)			
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswInternalBehavior, SwcInternalBehavior			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	***InternalBehavior*** (abstract)			
constant Memory	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior. The shortName of ParameterDataPrototype has to be equal to the ''C' identifier of the described constant. The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType. The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=constantMemory, constantMemory.short Name, constantMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior **Tags:** atp.Splitkey=dataTypeMapping
exclusiveArea	ExclusiveArea	*	aggr	This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveArea, exclusiveArea.shortName, exclusiveArea.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaNestingOrder, exclusiveArea NestingOrder.shortName, exclusiveAreaNesting Order.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	***InternalBehavior*** (abstract)			
staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurerment variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE. The shortName of the VariableDataPrototype has to be equal with the ''C' identifier of the described variable. The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=staticMemory, staticMemory.shortName, staticMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table F.79: InternalBehavior

Class	**MemorySection**			
Note	Provides a description of an abstract memory section used in the Implementation for code or data. It shall be declared by the Implementation Description of the module or component, which actually allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the generated Implementation Description of the RTE shall contain the corresponding MemorySections. The attribute "symbol" (if symbol is missing: "shortName") defines the module or component specific section name used in the code. For details see the document "Specification of Memory Mapping". Typically the section name is build according the pattern: <SwAddrMethod shortName>[_<further specialization nominator>][_<alignment>] where • **[<SwAddrMethod shortName>]** is the shortName of the referenced SwAddrMethod • **[_<further specialization nominator>]** is an optional infix to indicate the specialization in the case that several MemorySections for different purpose of the same Implementation Description referring to the same or equally named SwAddrMethods. • **[_<alignment>]** is the alignment attributes value and is only applicable in the case that the memory AllocationKeywordPolicy value of the referenced SwAddrMethod is set to addrMethodShortNameAnd Alignment MemorySection used to Implement the code of RunnableEntitys and BswSchedulableEntitys shall have a symbol (if missing: shortName) identical to the referred SwAddrMethod to conform to the generated RTE header files. In addition to the section name described above, a prefix is used in the corresponding macro code in order to define a name space. This prefix is by default given by the shortName of the BswModule Description resp. the SwComponentType. It can be superseded by the prefix attribute.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.memorySection			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
alignment	AlignmentType	0..1	attr	The attribute describes the typical alignment of objects within this memory section.

▽

△

Class	**MemorySection**			
executableEntity	ExecutableEntity	*	ref	Reference to the ExecutableEntitites located in this section. This allows to locate different Executable Entitities in different sections even if the associated Sw Addrmethod is the same. This is applicable to code sections only. **Tags:** atp.Splitkey=executableEntity
prefix	SectionNamePrefix	0..1	ref	The prefix used to set the memory section's namespace in the code. The existence of a prefix element supersedes rules for a default prefix (such as the Bsw ModuleDescription's shortName). This allows the user to define several name spaces for memory sections within the scope of one module, cluster or SWC. **Tags:** atp.Splitkey=prefix
size	PositiveInteger	0..1	attr	The size in bytes of the section.
swAddrmethod	SwAddrMethod	0..1	ref	This association indicates that this module specific (abstract) memory section is part of an overall SwAddr Method, referred by the upstream declarations (e.g. calibration parameters, data element prototypes, code entities) which share a common addressing strategy. This can be evaluated for the ECU configuration of the build support. This association shall always be declared by the Implementation description of the module or component, which allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the software components only declare the grouping of its data prototypes to SwAddrMethods, and the generated Implementation Description of the RTE actually sets up this association. **Tags:** atp.Splitkey=swAddrmethod
symbol	Identifier	0..1	attr	Defines the section name as explained in the main description. By using this attribute for code generation (instead of the shortName) it is possible to define several different MemorySections having the same name - e.g. symbol = CODE - but using different sectionName Prefixes.

Table F.80: MemorySection

Class	**MetaDataItem**			
Note	This meta-class represents a single meta-data item.			
Base	*ARObject*			
Aggregated by	MetaDataItemSet.metaDataItem			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
length	PositiveInteger	0..1	attr	This attribute determines the length of the `MetaDataItem` at run-time. The length is given in bytes.
metaDataItem Type	TextValueSpecification	0..1	aggr	This aggregation contributes the specification of the concrete meta-data item type.

Table F.81: MetaDataItem

Class	MetaDataItemSet			
Note	This meta-class represents the ability to define a set of meta-data items to be used in SenderReceiverInterfaces.			
Base	*ARObject*			
Aggregated by	SenderReceiverInterface.metaDataItemSet			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	VariableDataPrototype	*	ref	This reference identifies the dataElement for which the ordered list of meta-data items is defined. **Tags:** atp.Splitkey=dataElement
metaDataItem (ordered)	MetaDataItem	*	aggr	This aggregation represents the ordered definition of meta-data items.

Table F.82: MetaDataItemSet

Class	ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationGroup.modeDeclaration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	PositiveInteger	0..1	attr	The RTE shall take the value of this attribute for generating the source code representation of this Mode Declaration.

Table F.83: ModeDeclaration

Class	ModeDeclarationGroup			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.recommendedPackage=ModeDeclarationGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialMode	ModeDeclaration	0..1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred. **Tags:** atp.Splitkey=initialMode
mode Declaration	ModeDeclaration	*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeDeclaration, modeDeclaration.shortName, modeDeclaration.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime
modeManager ErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user). **Tags:** atp.Splitkey=modeManagerErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.

▽

△

Class	**ModeDeclarationGroup**			
modeTransition	ModeTransition	*	aggr	This represents the avaliable ModeTransitions of the ModeDeclarationGroup **Tags:** atp.Splitkey=modeTransition, modeTransition.shortName This Attribute is only used by the AUTOSAR Classic Platform.
modeUserErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager). **Tags:** atp.Splitkey=modeUserErrorBehavior This Attribute is only used by the AUTOSAR Classic Platform.
onTransitionValue	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses. This Attribute is only used by the AUTOSAR Classic Platform.

Table F.84: ModeDeclarationGroup

Class	**ModeDeclarationGroupPrototype**			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.providedModeGroup, BswModuleDescription.requiredModeGroup, FirewallStateSwitchInterface.firewallStateMachine, FunctionGroupSet.functionGroup, ModeSwitchInterface.modeGroup, Process.processStateMachine, StateManagementStateNotification.stateMachine			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time. This Attribute is only used by the AUTOSAR Classic Platform.
type	ModeDeclarationGroup	0..1	tref	The "collection of ModeDeclarations" (= ModeDeclarationGroup) supported by a component **Stereotypes:** isOfType **Tags:** atp.Splitkey=type

Table F.85: ModeDeclarationGroupPrototype

Class	**ModeSwitchInterface**			
Note	A mode switch interface declares a `ModeDeclarationGroupPrototype` to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
modeGroup	ModeDeclarationGroupPrototype	0..1	aggr	The `ModeDeclarationGroupPrototype` of this mode interface. **Tags:** atp.Splitkey=modeGroup, modeGroup.shortName

Table F.86: ModeSwitchInterface

Enumeration	**NmVariantEnum**
Note	Supported NmCoordinator roles.
Aggregated by	*CommunicationConnector*.nmVariant, *NmNode*.nmVariant
Literal	***Description***
full	AUTOSAR NM is available **Tags:** atp.EnumerationLiteralIndex=3
light	No AUTOSAR NM is available, but functionality to shut down a channel **Tags:** atp.EnumerationLiteralIndex=1
none	No NM available **Tags:** atp.EnumerationLiteralIndex=0
passive	AUTOSAR NM running in passive mode available **Tags:** atp.EnumerationLiteralIndex=2
slaveActive	No NM is available. This is used for e.g. LIN slaves **Tags:** atp.EnumerationLiteralIndex=4
slavePassive	No NM is available. This used for e.g. Ethernet communication channels with OA TC10 compliant hardware **Tags:** atp.EnumerationLiteralIndex=5

Table F.87: NmVariantEnum

Class	**NonqueuedReceiverComSpec**			
Note	Communication attributes specific to non-queued receiving.			
Base	*ARObject*, *RPortComSpec*, *ReceiverComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aliveTimeout	TimeValue	0..1	attr	Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description. If the aliveTimeout attribute is 0 no timeout monitoring shall be performed. This Attribute is only used by the AUTOSAR Classic Platform.
enableUpdate	Boolean	0..1	attr	This attribute controls whether application code is entitled to check whether the value of the corresponding Variable DataPrototype has been updated. This Attribute is only used by the AUTOSAR Classic Platform.
filter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement. **Tags:** atp.Splitkey=filter
handleData Status	Boolean	0..1	attr	If this attribute is set to true, then the Rte_IStatus API shall exist. If the attribute does not exist or is set to false, then the Rte_IStatus API may still exist in response to the existence of further conditions. This Attribute is only used by the AUTOSAR Classic Platform.



△

Class	**NonqueuedReceiverComSpec**			
handleNever Received	Boolean	0..1	attr	This attribute specifies whether for the corresponding VariableDataPrototype the "never received" flag is available. If yes, the RTE is supposed to assume that initially the VariableDataPrototype has not been received before. After the first reception of the corresponding VariableDataPrototype the flag is cleared. • If the value of this attribute is set to "true" the flag is required. • If set to "false", the RTE shall not support the "never received" functionality for the corresponding Variable DataPrototype. This Attribute is only used by the AUTOSAR Classic Platform.
handleTimeout Type	HandleTimeoutEnum	0..1	attr	This attribute controls the behavior with respect to the handling of timeouts. This Attribute is only used by the AUTOSAR Classic Platform.
initValue	ValueSpecification	0..1	aggr	Initial value to be used in case the sending component is not yet initialized. If the sender also specifies an initial value, then the receiver's value will be used. **Tags:** atp.Splitkey=initValue This Attribute is only used by the AUTOSAR Classic Platform.
returnNoNew DataEnabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return No New Data Error if no new data is received from COM. This Attribute is only used by the AUTOSAR Classic Platform.
timeout Substitution Value	ValueSpecification	0..1	aggr	This attribute represents the substitution value applicable in the case of a timeout. **Tags:** atp.Splitkey=timeoutSubstitutionValue This Attribute is only used by the AUTOSAR Classic Platform.
transportError CountEnabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return the number of transport errors (i.e. COM, SecOC errors) that happened since the last call of the respective API. This Attribute is only used by the AUTOSAR Classic Platform.
valueErrorCount Enabled	Boolean	0..1	attr	This attribute defines whether the RTE API functions related to the RPortPrototype shall return the number of value errors (i.e. out of range, invalid value) that happened since the last call of the respective API. This Attribute is only used by the AUTOSAR Classic Platform.

Table F.88: NonqueuedReceiverComSpec

Class	**NumericalValueSpecification**
Note	A numerical ValueSpecification which is intended to be assigned to a Primitive data element. Note that the numerical value is a variant, it can be computed by a formula.
Base	*ARObject*, *ValueSpecification*

▽

△

Class	**NumericalValueSpecification**			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	Numerical	0..1	attr	This is the value itself. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table F.89: NumericalValueSpecification

Class	**NvBlockSwComponentType**			
Note	The NvBlockSwComponentType defines non volatile data which data can be shared between Sw ComponentPrototypes. The non volatile data of the NvBlockSwComponentType are accessible via provided and required ports. **Tags:** atp.recommendedPackage=SwComponentTypes This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
bulkNvData Descriptor	BulkNvDataDescriptor	*	aggr	This aggregation formally defines the bulk Nv Blocks that are provided to the application software by the enclosing NvBlockSwComponentType. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=bulkNvDataDescriptor, bulkNvData Descriptor.shortName, bulkNvDataDescriptor.variation Point.shortLabel vh.latestBindingTime=preCompileTime
nvBlock Descriptor	NvBlockDescriptor	*	aggr	Specification of the properties of exactly one NVRAM Block. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=nvBlockDescriptor, nvBlockDescriptor.short Name, nvBlockDescriptor.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table F.90: NvBlockSwComponentType

Class	**NvDataInterface**
Note	A non volatile data interface declares a number of `VariableDataPrototype`s to be exchanged between non volatile block components and atomic software components. **Tags:** atp.recommendedPackage=PortInterfaces

▽

△

Class	NvDataInterface			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nvData	VariableDataPrototype	*	aggr	The VariableDataPrototype of this nv data interface. **Tags:** atp.Splitkey=nvData, nvData.shortName

Table F.91: NvDataInterface

Class	PPortPrototype			
Note	Component port providing a certain port interface.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedInterface

Table F.92: PPortPrototype

Class	PRPortPrototype			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provided Required Interface	PortInterface	0..1	tref	This represents the `PortInterface` used to type the `PRPortPrototype`. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedRequiredInterface

Table F.93: PRPortPrototype

Class	***PackageableElement*** (abstract)			
Note	This meta-class specifies the ability to be a member of an AUTOSAR package.			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*ARElement*, EnumerationMappingTable, *FibexElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table F.94: PackageableElement

Class	ParameterDataPrototype			
Note	A `ParameterDataPrototype` represents a formalized generic piece of information that is typically immutable by the application software layer, but mutable by measurement and calibration tools. `ParameterDataPrototype` is used in various contexts and the specific context gives the otherwise generic `ParameterDataPrototype` a dedicated semantics.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswInternalBehavior.perInstanceParameter, *InternalBehavior*.constantMemory, NvBlockDescriptor.romBlock, ParameterInterface.parameter, SwcInternalBehavior.perInstanceParameter, SwcInternalBehavior.sharedParameter			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype **Tags:** atp.Splitkey=initValue

Table F.95: ParameterDataPrototype

Class	ParameterInterface			
Note	A parameter interface declares a number of parameter and characteristic values to be exchanged between parameter components and software components. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
parameter	ParameterDataPrototype	*	aggr	The ParameterDataPrototype of this ParameterInterface. **Tags:** atp.Splitkey=parameter, parameter.shortName

Table F.96: ParameterInterface

Class	ParameterSwComponentType			
Note	The ParameterSwComponentType defines parameters and characteristic values accessible via provided Ports. The provided values are the same for all connected SwComponentPrototypes **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
constantMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular ParameterSwComponentType **Tags:** atp.Splitkey=constantMapping
dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular ParameterSwComponentType **Tags:** atp.Splitkey=dataTypeMapping



△

Class	ParameterSwComponentType			
instantiation DataDefProps	InstantiationDataDefProps	*	aggr	The purpose of this is that within the context of a given SwComponentType some data def properties of individual instantiations can be modified. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of PortPrototypes **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationDataDefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table F.97: ParameterSwComponentType

Class	PassThroughSwConnector			
Note	This kind of `SwConnector` can be used inside a `CompositionSwComponentType` to connect two delegation `PortPrototype`s.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
providedOuterPort	AbstractProvidedPortPrototype	0..1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector. **Tags:** atp.Splitkey=providedOuterPort
requiredOuterPort	AbstractRequiredPortPrototype	0..1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector. **Tags:** atp.Splitkey=requiredOuterPort

Table F.98: PassThroughSwConnector

Class	PhysConstrs			
Note	This meta-class represents the ability to express physical constraints. Therefore it has (in opposite to InternalConstrs) a reference to a Unit.			
Base	*ARObject*			
Aggregated by	DataConstrRule.physConstrs			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
maxDiff	Numerical	0..1	attr	Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis. **Tags:** xml.sequenceOffset=60
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in curves and maps. **Tags:** xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This specifies the monotony constraints on the data object. Note that this applies only to curves and maps. **Tags:** xml.sequenceOffset=70

▽

△

Class	**PhysConstrs**			
unit	Unit	0..1	ref	This is the unit to which the physical constraints relate to. In particular, it is the physical unit of the specified limits. **Tags:** atp.Splitkey=unit xml.sequenceOffset=80
upperLimit	Limit	0..1	attr	This specifies the upper limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table F.99: PhysConstrs

Class	**PncMappingIdent**			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable PncMapping.			
Base	*ARObject*, *Referrable*			
Aggregated by	PncMapping.ident			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.100: PncMappingIdent

Class	**PortAPIOption**			
Note	Options how to generate the signatures of calls for an AtomicSwComponentType in order to communicate over a PortPrototype (for calls into a RunnableEntity as well as for calls from a Runnable Entity to the PortPrototype).			
Base	*ARObject*			
Aggregated by	SwcInternalBehavior.portAPIOption			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
enableTake Address	Boolean	0..1	attr	If set to true, the software-component is able to use the API reference for deriving a pointer to an object.
errorHandling	DataTransformation ErrorHandlingEnum	0..1	attr	This specifies whether a RunnableEntity accessing a Port Prototype that is referenced by this PortAPIOption shall specifically handle transformer errors or not.
indirectAPI	Boolean	0..1	attr	If set to true this attribute specifies an "indirect API" to be generated for the associated port which means that the software-component is able to access the actions on a port via a pointer to an object representing a port. This allows e.g. iterating over ports in a loop. This option has no effect for PPortPrototypes of client/server interfaces.
port	PortPrototype	0..1	ref	The option is valid for generated functions related to communication over this port **Stereotypes:** atpIdentityContributor
portArgValue (ordered)	PortDefinedArgument Value	*	aggr	An argument value defined by this port.
supported Feature	SwcSupportedFeature	*	aggr	This collection specifies which features are supported by the RunnableEntitys which access a PortPrototype that it referenced by this PortAPIOption. **Tags:** atp.Splitkey=supportedFeature

▽

△

Class	PortAPIOption			
transformer Status Forwarding	DataTransformation StatusForwardingEnum	0..1	attr	This attribute specifies whether a RunnableEntity accessing a PortPrototype that is referenced by this PortAPIOption shall be able to forward a status code to the transformer chain.

Table F.101: PortAPIOption

Class	PortGroup			
Note	Group of ports which share a common functionality. Example: need specific network resources. This information shall be available on the VFB level in order to delegate it properly via compositions. When propagated into the ECU extract, this information is used as input for the configuration of Services like the Communication Manager. A PortGroup is defined locally in a component (which can be a composition) and refers to the "outer" ports belonging to the group as well as to the "inner" groups which propagate this group into the components which are part of a composition. A PortGroup within an atomic SWC cannot be linked to inner groups.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.portGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
innerGroup	PortGroup	*	iref	Links a PortGroup in a composition to another PortGroup, that is defined in a component which is part of this CompositionSwComponentType. **Tags:** atp.Splitkey=innerGroup, innerGroup.context, innerGroup.target **InstanceRef implemented by:** InnerPortGroupInCompositionInstanceRef
outerPort	PortPrototype	*	ref	Outer PortPrototype of this AtomicSwComponentType which belongs to the group. A port can belong to several groups or to no group at all. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=outerPort, outerPort.portPrototype, outerPort.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table F.102: PortGroup

Class	***PortInterface*** (abstract)			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ClientServerInterface, *DataInterface*, ModeSwitchInterface, TriggerInterface			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	***PortInterface*** (abstract)			
isService	Boolean	0..1	attr	This flag is set if the `PortInterface` is to be used for communication between an • `ApplicationSwComponentType` or • `ServiceProxySwComponentType` or • `SensorActuatorSwComponentType` or • `ComplexDeviceDriverSwComponentType` • `ServiceSwComponentType` • `EcuAbstractionSwComponentType` and a `ServiceSwComponentType` (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime This Attribute is only used by the AUTOSAR Classic Platform.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service. This Attribute is only used by the AUTOSAR Classic Platform.

Table F.103: PortInterface

Class	***PortInterfaceMapping*** (abstract)			
Note	Specifies one `PortInterfaceMapping` to support the connection of Ports typed by two different `PortInterface`s with PortInterface elements having unequal names and/or unequal semantic (resolution or range).			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ClientServerInterfaceMapping, ModeInterfaceMapping, TriggerInterfaceMapping, VariableAndParameterInterfaceMapping			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.104: PortInterfaceMapping

Class	***PortPrototype*** (abstract)			
Note	Base class for the ports of an AUTOSAR software component. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this `PortPrototype` with respect to client/server communication. **Tags:** atp.Splitkey=clientServerAnnotation
delegatedPortAnnotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port. **Tags:** atp.Splitkey=delegatedPortAnnotation

▽

△

Class	***PortPrototype*** (abstract)			
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port. **Tags:** atp.Splitkey=ioHwAbstractionServerAnnotation
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port. **Tags:** atp.Splitkey=modePortAnnotation
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port. **Tags:** atp.Splitkey=nvDataPortAnnotation
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port. **Tags:** atp.Splitkey=parameterPortAnnotation
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication. **Tags:** atp.Splitkey=senderReceiverAnnotation
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port. **Tags:** atp.Splitkey=triggerPortAnnotation

Table F.105: PortPrototype

Class	**PostBuildVariantCriterion**			
Note	This class specifies one particular PostBuildVariantSelector. **Tags:** atp.recommendedPackage=PostBuildVariantCriterions			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
compuMethod	CompuMethod	0..1	ref	The compuMethod specifies the possible values for the variant criterion serving as an enumerator. **Tags:** atp.Splitkey=compuMethod

Table F.106: PostBuildVariantCriterion

Class	**RPortPrototype**			
Note	Component port requiring a certain port interface.			
Base	*ARObject*, *AbstractRequiredPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mayBe Unconnected	Boolean	0..1	attr	If set to true, this attribute indicates that the enclosing `RPortPrototype` may be left unconnected and that this aspect has explicitly been considered in the software-component's design. This Attribute is only used by the AUTOSAR Classic Platform.
required Interface	PortInterface	0..1	tref	The interface that this port requires. **Stereotypes:** isOfType **Tags:** atp.Splitkey=requiredInterface

Table F.107: RPortPrototype

Class	**ReceiverComSpec** (abstract)			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	*ARObject*, *RPortComSpec*			
Subclasses	NonqueuedReceiverComSpec, QueuedReceiverComSpec			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compositeNetworkRepresentation	CompositeNetworkRepresentation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a ReceiverComSpec. The purpose of this aggregation is to be able to specify the network representation of leaf elements of Application CompositeDataTypes. **Tags:** atp.Splitkey=compositeNetworkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
dataElement	AutosarDataPrototype	0..1	ref	Data element these attributes belong to. **Stereotypes:** atpIdentityContributor
handleOutOfRange	HandleOutOfRangeEnum	0..1	attr	This attribute controls how values that are out of the specified range are handled according to the values of HandleOutOfRangeEnum. This Attribute is only used by the AUTOSAR Classic Platform.
handleOutOfRangeStatus	HandleOutOfRangeStatusEnum	0..1	attr	Control the way how return values are created in case of an out-of-range situation. This Attribute is only used by the AUTOSAR Classic Platform.
networkRepresentation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus. **Tags:** atp.Splitkey=networkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
receptionProps	ReceptionComSpecProps	0..1	aggr	"This aggregation represents the definition transmission props in the context of the enclosing ReceiverComSpec. **Tags:** atp.Splitkey=receptionProps
replaceWith	VariableAccess	0..1	aggr	This aggregation is used to identify the AutosarDataPrototype to be taken for sourcing an external replacement in the out-of-range and invalidValue handling. **Tags:** atp.Splitkey=replaceWith, replaceWith.shortName This Attribute is only used by the AUTOSAR Classic Platform.
transformationComSpecProps	TransformationComSpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation. **Tags:** atp.Splitkey=transformationComSpecProps

Table F.108: ReceiverComSpec

Class	**ReceptionComSpecProps**			
Note	This meta-class defines a set of reception attributes which the application software is assumed to implement.			
Base	*ARObject*			
Aggregated by	*ReceiverComSpec*.receptionProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**ReceptionComSpecProps**			
comHandler TaskMapping Enabled	Boolean	0..1	attr	This attribute defines whether the ComHandler Task Mapping is activated or deactivated. This Attribute is only used by the AUTOSAR Classic Platform.
dataUpdate Period	TimeValue	0..1	attr	This attribute defines the period in which the application shall check for updated data. This attribute is used for the configuration of the E2E protection, but may also indicate a general data reception period.
invalidValue BitfieldErrors Enabled	Boolean	0..1	attr	This attribute defines whether the invalid value error bitfield shall be used which contains the information of invalid value errors of the direct child elements of the enclosing structure and not of elements of sub-structures. This Attribute is only used by the AUTOSAR Classic Platform.
outOfRange BitfieldErrors Enabled	Boolean	0..1	attr	This attribute defines whether the out-of-range error bitfield shall be used which contains the information of out-of-range errors of the direct child elements of the enclosing structure and not of elements of sub-structures. This Attribute is only used by the AUTOSAR Classic Platform.
timeout	TimeValue	0..1	attr	This attribute defines the time interval after which the application shall assume that the to be received data reception has timed out, i.e. the respective data has not been received for that amount of time.

Table F.109: ReceptionComSpecProps

Class	***Referrable*** (abstract)			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	*ARObject*			
Subclasses	*AtpDefinition*, BswDistinguishedPartition, *BswModuleCallPoint*, BswModuleClientServerEntry, BswVariableAccess, CouplingPortTrafficClassAssignment, *DiagnosticEnvModeElement*, EthernetPriorityRegeneration, ExclusiveAreaNestingOrder, *HwDescriptionEntity*, *ImplementationProps*, LinSlaveConfigIdent, ModeTransition, *MultilanguageReferrable*, PncMappingIdent, *SingleLanguageReferrable*, SoConIPduIdentifier, TpConnectionIdent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. **Stereotypes:** atpIdentityContributor **Tags:** xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:** atp.Splitkey=shortNameFragment xml.sequenceOffset=-90

Table F.110: Referrable

Class	**RoleBasedPortAssignment**			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPort Prototype) of an AtomicSwComponentType. With this assignment, the role of the service port can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct connector.			
Base	*ARObject*			
Aggregated by	NvBlockDescriptor.clientServerPort, SwcServiceDependency.assignedPort			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
portPrototype	PortPrototype	0..1	ref	Service PortPrototype used in the assigned role. This PortPrototype shall either belong to the same AtomicSw ComponentType as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockSw ComponentType as the NvBlockDescriptor. **Tags:** atp.Splitkey=portPrototype
role	Identifier	0..1	attr	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a Port Interface as standardized in the Software Specification of the related AUTOSAR Service.

Table F.111: RoleBasedPortAssignment

Class	**RunnableEntity**			
Note	A `RunnableEntity` represents the smallest code-fragment that is provided by an `AtomicSwComponentType` and are executed under control of the RTE. `RunnableEntity`s are for instance set up to respond to data reception or operation invocation on a server.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *ExecutableEntity*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.runnable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
argument (ordered)	RunnableEntity Argument	*	aggr	This represents the formal definition of an argument to a RunnableEntity.
asynchronous ServerCall ResultPoint	AsynchronousServer CallResultPoint	*	aggr	The server call result point admits a runnable to fetch the result of an asynchronous server call. The aggregation of AsynchronousServerCallResultPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes and the variant existence of server call result points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=asynchronousServerCallResultPoint, asynchronousServerCallResultPoint.shortName, asynchronousServerCallResultPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
canBeInvoked Concurrently	Boolean	0..1	attr	If the value of this attribute is set to "true" the enclosing `RunnableEntity` can be invoked concurrently (even for one instance of the corresponding `AtomicSwComponentType`). This implies that it is the responsibility of the implementation of the `RunnableEntity` to take care of this form of concurrency.



△

Class	**RunnableEntity**			
dataRead Access	VariableAccess	*	aggr	RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataReadAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataReadAccess in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReadAccess, dataReadAccess.shortName, dataReadAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointBy Argument	VariableAccess	*	aggr	RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of an argument in the function signature. The aggregation of dataReceivePointByArgument is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data receive points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReceivePointByArgument, dataReceivePointByArgument.shortName, dataReceivePointByArgument.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointByValue	VariableAccess	*	aggr	RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of the return value. The aggregation of dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataReceivePointByValue, dataReceivePointByValue.shortName, dataReceivePointByValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataSendPoint	VariableAccess	*	aggr	RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataSendPoint is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data send points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataSendPoint, dataSendPoint.shortName, dataSendPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
dataWrite Access	VariableAccess	*	aggr	RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=dataWriteAccess, dataWriteAccess.shortName, dataWriteAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
external TriggeringPoint	ExternalTriggeringPoint	*	aggr	The aggregation of ExternalTriggeringPoint is subject to variability with the purpose to support the conditional existence of trigger ports or the variant existence of external triggering points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=externalTriggeringPoint, externalTriggeringPoint.ident.shortName, externalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internal TriggeringPoint	InternalTriggeringPoint	*	aggr	The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint, internalTriggeringPoint.shortName, internalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeAccess Point	ModeAccessPoint	*	aggr	The runnable has a mode access point. The aggregation of ModeAccessPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode access points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeAccessPoint, modeAccessPoint.ident.shortName, modeAccessPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeSwitch Point	ModeSwitchPoint	*	aggr	The runnable has a mode switch point. The aggregation of ModeSwitchPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode switch points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=modeSwitchPoint, modeSwitchPoint.shortName, modeSwitchPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
parameter Access	ParameterAccess	*	aggr	The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a Parameter DataPrototype which may either be local or within a Port Prototype. The aggregation of ParameterAccess is subject to variability with the purpose to support the conditional existence of parameter ports and component local parameters as well as the variant existence of Parameter Access (points) in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=parameterAccess, parameterAccess.shortName, parameterAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
readLocal Variable	VariableAccess	*	aggr	The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. The aggregation of readLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of read LocalVariable (points) in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=readLocalVariable, readLocalVariable.shortName, readLocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
serverCallPoint	ServerCallPoint	*	aggr	The RunnableEntity has a ServerCallPoint. The aggregation of ServerCallPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes or the variant existence of server call points in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serverCallPoint, serverCallPoint.shortName, serverCallPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
symbol	CIdentifier	0..1	attr	The symbol describing this `RunnableEntity`'s entry point. This is considered the API of the `RunnableEntity` and is required during the RTE contract phase.
waitPoint	WaitPoint	*	aggr	The `WaitPoint` associated with the `RunnableEntity`. **Tags:** atp.Splitkey=waitPoint, waitPoint.shortName

▽

△

Class	RunnableEntity			
writtenLocal Variable	VariableAccess	*	aggr	The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=writtenLocalVariable, writtenLocal Variable.shortName, writtenLocalVariable.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table F.112: RunnableEntity

Class	*SenderComSpec* (abstract)			
Note	Communication attributes for a sender port (`PPortPrototype` typed by `SenderReceiverInterface`).			
Base	*ARObject*, *PPortComSpec*			
Subclasses	NonqueuedSenderComSpec, QueuedSenderComSpec			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a SenderComSpec. **Tags:** atp.Splitkey=compositeNetworkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
dataElement	AutosarDataPrototype	0..1	ref	Data element these quality of service attributes apply to. **Stereotypes:** atpIdentityContributor
handleOutOf Range	HandleOutOfRange Enum	0..1	attr	This attribute controls how out-of-range values shall be dealt with. This Attribute is only used by the AUTOSAR Classic Platform.
network Representation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus. **Tags:** atp.Splitkey=networkRepresentation This Attribute is only used by the AUTOSAR Classic Platform.
transmission Acknowledge	Transmission Acknowledgement Request	0..1	aggr	Requested transmission acknowledgement for data element. **Tags:** atp.Splitkey=transmissionAcknowledge This Attribute is only used by the AUTOSAR Classic Platform.
transmission Props	TransmissionComSpec Props	0..1	aggr	This aggregation represents the definition transmission props in the context of the enclosing SenderComSpec. **Tags:** atp.Splitkey=transmissionProps

▽

△

Class	***SenderComSpec*** (abstract)			
usesEndToEnd Protection	Boolean	0..1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table F.113: SenderComSpec

Class	**SenderReceiverInterface**			
Note	A sender/receiver interface declares a number of data elements to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataElement	VariableDataPrototype	*	aggr	The data elements of this `SenderReceiverInterface`. **Tags:** atp.Splitkey=dataElement, dataElement.short Name
invalidation Policy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement **Tags:** atp.Splitkey=invalidationPolicy
metaDataItem Set	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with `dataElement`s of the enclosing `SenderReceiverInterface` **Tags:** atp.Splitkey=metaDataItemSet

Table F.114: SenderReceiverInterface

Class	**SensorActuatorSwComponentType**			
Note	The SensorActuatorSwComponentType introduces the possibility to link from the software representation of a sensor/actuator to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
sensorActuator	HwDescriptionEntity	0..1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware. **Tags:** atp.Splitkey=sensorActuator

Table F.115: SensorActuatorSwComponentType

Class	***ServerCallPoint*** (abstract)
Note	If a RunnableEntity owns a `ServerCallPoint` it is entitled to invoke a particular `ClientServerOperation` of a specific `RPortPrototype` of the corresponding `AtomicSwComponentType`. This Class is only used by the AUTOSAR Classic Platform.
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Subclasses	AsynchronousServerCallPoint, SynchronousServerCallPoint

▽

△

Class	**ServerCallPoint** (abstract)			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation	ClientServerOperation	0..1	iref	The operation that is called by this runnable. **Tags:** atp.Splitkey=operation, operation.contextRPort, operation.targetRequiredOperation **InstanceRef implemented by:** ROperationInAtomicSwcInstanceRef
timeout	TimeValue	0..1	attr	Time in seconds before the server call times out and returns with an error message. It depends on the call type (synchronous or asynchronous) how this is reported.

Table F.116: ServerCallPoint

Class	**ServiceNeeds** (abstract)			
Note	This expresses the abstract needs that a Software Component or Basic Software Module has on the configuration of an AUTOSAR Service to which it will be connected. "Abstract needs" means that the model abstracts from the Configuration Parameters of the underlying Basic Software.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswMgrNeeds, ChargeManagerNeeds, ComMgrUserNeeds, CryptoKeyManagementNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, *DiagnosticCapabilityElement*, DltUserNeeds, *DoIpServiceNeeds*, EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, GeneralPurposeTimerServiceNeeds, GlobalSupervisionNeeds, IdsMgrCustomTimestampNeeds, IdsMgrNeeds, IndicatorStatusNeeds, J1939DcmDm19Support, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequestServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpointNeeds, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, VendorSpecificServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table F.117: ServiceNeeds

Class	**ServiceProxySwComponentType**
Note	This class provides the ability to express a software-component which provides access to an internal service for remote ECUs. It acts as a proxy for the service providing access to the service. An important use case is the request of vehicle mode switches: Such requests can be communicated via sender-receiver interfaces across ECU boundaries, but the mode manager being responsible to perform the mode switches is an AUTOSAR Service which is located in the Basic Software and is not visible in the VFB view. To handle this situation, a ServiceProxySwComponentType will act as proxy for the mode manager. It will have R-Ports to be connected with the mode requestors on VFB level and Service-Ports to be connected with the local mode manager at ECU integration time. Apart from the semantics, a ServiceProxySwComponentType has these specific properties: • A prototype of it can be mapped to more than one ECUs in the system description. • Exactly one additional instance of it will be created in the ECU-Extract per ECU to which the prototype has been mapped. • For remote communication, it can have only R-Ports with sender-receiver interfaces and 1:n semantics. • There shall be no connectors between two prototypes of any ServiceProxySwComponentType. **Tags:** atp.recommendedPackage=SwComponentTypes
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*
Aggregated by	ARPackage.element

▽

△

Class	**ServiceProxySwComponentType**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.118: ServiceProxySwComponentType

Class	**ServiceSwComponentType**			
Note	ServiceSwComponentType is used for configuring services for a given ECU. Instances of this class are only to be created in ECU Configuration phase for the specific purpose of the service configuration. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *Collectable Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.119: ServiceSwComponentType

Enumeration	**ServiceVersionAcceptanceKindEnum**
Note	Defined the possible acceptance kinds for required service instances.
Aggregated by	ConsumedServiceInstance.versionDrivenFindBehavior, RequiredSomeipServiceInstance.version DrivenFindBehavior
Literal	***Description***
exactOrAnyMinor Version	Search for ANY or specific minor version service instance and select either ALL returned service instances (in case of ANY) or exactly the specific minor version service instances defined in required MinorVersion. **Tags:** atp.EnumerationLiteralIndex=0
minimumMinor Version	Search for ANY minor version service instance and select only those service instances which have an equal or greater minor version than given in requiredMinorVersion. **Tags:** atp.EnumerationLiteralIndex=1

Table F.120: ServiceVersionAcceptanceKindEnum

Class	***SignalPathConstraint*** (abstract)			
Note	Additional guidelines for the System Generator, which specific way a signal between two Software Components should take in the network without defining in which frame and with which timing it is transmitted.			
Base	*ARObject*			
Subclasses	CommonSignalPath, ForbiddenSignalPath, PermissibleSignalPath, SeparateSignalPath			
Aggregated by	SystemMapping.signalPathConstraint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the signal path constraint. **Tags:** atp.Splitkey=introduction

Table F.121: SignalPathConstraint

Class	**SomeipSdClientServiceInstanceConfig**			
Note	Client specific settings that are relevant for the configuration of SOME/IP Service-Discovery. **Tags:** atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initialFind Behavior	InitialSdDelayConfig	0..1	aggr	Controls initial find behavior of clients. **Tags:** atp.Splitkey=initialFindBehavior
priority	PositiveInteger	0..1	attr	This attribute defines the VLAN frame priority for Service Discovery messages that result from RequiredSomeip ServiceInstances that are referncing this SomeipSdClient ServiceInstanceConfig (Find, SubscribeEventGroup, Stop SubscribeEventgroup). Values from 0 (best effort) to 7 (highest) are allowed.
serviceFind TimeToLive	PositiveInteger	0..1	attr	This attribute represents the ability to define the time in seconds the service find is valid. Note! The TTL value for FindService entries is not used and shall be ignored by the server service. This configuration is only kept for backward compatibility. Default value if not specified shall be 0xFFFFFF. This Attribute is only used by the AUTOSAR Classic Platform.

Table F.122: SomeipSdClientServiceInstanceConfig

Class	**SomeipServiceVersion**			
Note	This meta-class represents the ability to describe a version of a SOME/IP Service.			
Base	*ARObject*			
Aggregated by	ConsumedServiceInstance.blocklistedVersion, RequiredSomeipServiceInstance.blocklistedVersion, SomeipServiceInterfaceDeployment.serviceInterfaceVersion			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. **Tags:** xml.sequenceOffset=10
minorVersion	PositiveInteger	0..1	attr	Minor Version of the ServiceInterface. **Tags:** xml.sequenceOffset=20

Table F.123: SomeipServiceVersion

Class	***StackUsage*** (abstract)			
Note	Describes the stack memory usage of a software.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	MeasuredStackUsage, RoughEstimateStackUsage, WorstCaseStackUsage			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this stack usage is described. **Tags:** atp.Splitkey=executableEntity
hardware Configuration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this stack usage is describing. **Tags:** atp.Splitkey=hardwareConfiguration

▽

△

Class	***StackUsage*** (abstract)			
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this stack usage is given. **Tags:** atp.Splitkey=hwElement
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this stack usage is provided for. **Tags:** atp.Splitkey=softwareContext

Table F.124: StackUsage

Primitive	**String**
Note	This represents a String in which white-space shall be normalized before processing. For example: in order to compare two Strings: • leading and trailing white-space needs to be removed • consecutive white-space (blank, cr, lf, tab) needs to be replaced by one blank. **Tags:** xml.xsd.customType=STRING xml.xsd.type=string

Table F.125: String

Class	**SubElementMapping**			
Note	This meta-class allows for the definition of mappings of elements of a composite data type.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.subElementMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstElement	SubElementRef	0..1	aggr	This represents the first element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=firstElement, firstElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
secondElement	SubElementRef	0..1	aggr	This represents the second element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=secondElement, secondElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
textTable Mapping	TextTableMapping	*	aggr	This allows for the text-table translation of individual elements of a composite data type. **Tags:** atp.Splitkey=textTableMapping

Table F.126: SubElementMapping

Class	***SubElementRef*** (abstract)
Note	This meta-class provides the ability to reference elements of composite data type.
Base	*ARObject*
Subclasses	ApplicationCompositeDataTypeSubElementRef, ImplementationDataTypeSubElementRef

▽

△

Class	***SubElementRef*** (abstract)			
Aggregated by	SubElementMapping.firstElement, SubElementMapping.secondElement			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.127: SubElementRef

Class	**SwBaseType**			
Note	This meta-class represents a base type used within ECU software. **Tags:** atp.recommendedPackage=BaseTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *BaseType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.128: SwBaseType

Class	***SwComponentType*** (abstract)			
Note	Base class for AUTOSAR software components.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AtomicSwComponentType*, CompositionSwComponentType, ParameterSwComponentType			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of `ConsistencyNeeds` owned by the enclosing `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=consistencyNeeds, consistencyNeeds.shortName, consistencyNeeds.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
port	PortPrototype	*	aggr	The `PortPrototype`s through which this `SwComponentType` can communicate. The aggregation of `PortPrototype` is subject to variability with the purpose to support the conditional existence of `PortPrototype`s. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=port, port.shortName, port.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portGroup, portGroup.shortName, portGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	***SwComponentType*** (abstract)			
swcMapping Constraint	SwComponentMapping Constraints	*	ref	Reference to constraints that are valid for this Sw ComponentType. **Tags:** atp.Splitkey=swcMappingConstraint This Attribute is only used by the AUTOSAR Classic Platform.
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the `SwComponentType`. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=swComponentDocumentation, sw ComponentDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which `UnitGroup`s are relevant in the context of referencing `SwComponentType`. **Tags:** atp.Splitkey=unitGroup This Attribute is only used by the AUTOSAR Classic Platform.

Table F.129: SwComponentType

Class	***SwConnector*** (abstract)			
Note	The base class for connectors between ports. Connectors have to be identifiable to allow references from the system constraint template.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AssemblySwConnector, DelegationSwConnector, PassThroughSwConnector			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
mapping	PortInterfaceMapping	0..1	ref	Reference to a `PortInterfaceMapping` specifying the mapping of unequal named `PortInterface` elements of the two different `PortInterface`s typing the two `PortPrototype`s which are referenced by the `SwConnector`. **Tags:** atp.Splitkey=mapping

Table F.130: SwConnector

Class	«atpVariation» **SwDataDefProps**
Note	This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated. Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations. SwDataDefProps covers various aspects: • Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet • Implementation aspects, mainly expressed by swImplPolicy, swAddrMethod, baseType, implementationDataType and additionalNativeTypeQualifier • Access policy for the MCD system, mainly expressed by swCalibrationAccess • Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalid Value • Code generation policy provided by swRecordLayout **Tags:** vh.latestBindingTime=codeGenerationTime
Base	*ARObject*
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, Cpp ImplementationDataTypeElement.swDataDefProps, *DataPrototype*.swDataDefProps, DataPrototype TransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, Diagnostic EnvDataElementCondition.swDataDefProps, DiagnosticExtendedDataRecordElement.swDataDefProps, DiagnosticSovdPrimitiveContentElement.swDataDefProps, DltArgumentProps.networkRep, FlatInstance Descriptor.swDataDefProps, ImplementationDataTypeElement.swDataDefProps, InstantiationDataDef Props.swDataDefProps, ISignal.networkRepresentationProps, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.swDataDefProps, *ReceiverComSpec*.network Representation, SecurityEventContextDataElement.networkRepresentation, *SenderComSpec*.network Representation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTarget Props.swDataDefProps, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, System Signal.physicalProps

Attribute	*Type*	*Mult.*	*Kind*	*Note*
additionalNative TypeQualifier	NativeDeclarationString	0..1	attr	This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string. **Tags:** xml.sequenceOffset=235
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** atp.Splitkey=annotation xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. **Tags:** atp.Splitkey=baseType xml.sequenceOffset=50

▽

△

Class	«atpVariation» **SwDataDefProps**			
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:** atp.Splitkey=compuMethod xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:** atp.Splitkey=dataConstr xml.sequenceOffset=190
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. **Tags:** xml.sequenceOffset=210
display Presentation	DisplayPresentation Enum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.
implementation DataType	AbstractImplementation DataType	0..1	ref	This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially • redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype • the target type of a pointer (see SwPointerTarget Props), if it does not refer to a base type directly • the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly • the data type of an SwServiceArg, if it does not refer to a base type directly **Tags:** atp.Splitkey=implementationDataType xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** atp.Splitkey=invalidValue xml.sequenceOffset=255
stepSize	Float	0..1	attr	This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself. **Tags:** atp.Splitkey=swAddrMethod xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended typical alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced Sw AddrMethod. **Tags:** xml.sequenceOffset=33

▽

△

Class	«atpVariation» **SwDataDefProps**			
swBit Representation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. **Tags:** atp.Splitkey=swBitRepresentation xml.sequenceOffset=60
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:** xml.sequenceOffset=70
swCalprmAxis Set	SwCalprmAxisSet	0..1	aggr	This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters. **Tags:** atp.Splitkey=swCalprmAxisSet xml.sequenceOffset=90
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** atp.Splitkey=swComparisonVariable xml.sequenceOffset=170 xml.typeElement=false
swData Dependency	SwDataDependency	0..1	aggr	Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system). **Tags:** atp.Splitkey=swDataDependency xml.sequenceOffset=200
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. **Tags:** atp.Splitkey=swHostVariable xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. **Tags:** xml.sequenceOffset=230
swIntended Resolution	Numerical	0..1	attr	The purpose of this element is to describe the requested quantization of data objects early on in the design process. The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula). In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution. The resolution is specified in the physical domain according to the property "unit". **Tags:** xml.sequenceOffset=240
swInterpolation Method	Identifier	0..1	attr	This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked. **Tags:** xml.sequenceOffset=250

▽

△

Class	«atpVariation» **SwDataDefProps**			
swIsVirtual	Boolean	0..1	attr	This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency . **Tags:** xml.sequenceOffset=260
swPointerTarget Props	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. **Tags:** atp.Splitkey=swPointerTargetProps xml.sequenceOffset=280
swRecord Layout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:** atp.Splitkey=swRecordLayout xml.sequenceOffset=290
swRefresh Timing	MultidimensionalTime	0..1	aggr	This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system. So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing. **Tags:** atp.Splitkey=swRefreshTiming xml.sequenceOffset=300
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** atp.Splitkey=swTextProps xml.sequenceOffset=120
swValueBlock Size	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
swValueBlock SizeMult (ordered)	Numerical	*	attr	This attribute is used to specify the dimensions of a value block (VAL_BLK) for the case that that value block has more than one dimension. The dimensions given in this attribute are ordered such that the first entry represents the first dimension, the second entry represents the second dimension, and so on. For one-dimensional value blocks the attribute swValue BlockSize shall be used and this attribute shall not exist. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
unit	Unit	0..1	ref	Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible. **Tags:** atp.Splitkey=unit xml.sequenceOffset=350

▽

△

Class	«atpVariation» **SwDataDefProps**			
valueAxisData Type	ApplicationPrimitive DataType	0..1	ref	The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType. **Tags:** atp.Splitkey=valueAxisDataType xml.sequenceOffset=355

Table F.131: SwDataDefProps

Enumeration	**SwImplPolicyEnum**
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Aggregated by	BswInternalTriggeringPoint.swImplPolicy, InternalTriggeringPoint.swImplPolicy, SwDataDefProps.swImplPolicy, Trigger.swImplPolicy
Literal	*Description*
const	forced implementation such that the running software within the ECU shall not modify it. For example implemented with the "const" modifier in C. This can be applied for parameters (not for those in NVRAM) as well as argument data prototypes. **Tags:** atp.EnumerationLiteralIndex=0
fixed	This data element is fixed. In particular this indicates, that it might also be implemented e.g. as in place data, (#DEFINE). **Tags:** atp.EnumerationLiteralIndex=1
measurementPoint	The data element is created for measurement purposes only. The data element is never read directly within the ECU software. In contrast to a "standard" data element in an unconnected provide port is, this unconnection is guaranteed for measurementPoint data elements. **Tags:** atp.EnumerationLiteralIndex=2
queued	The content of the data element is queued and the data element has 'event' semantics, i.e. data elements are stored in a queue and all data elements are processed in 'first in first out' order. The queuing is intended to be implemented by RTE Generator. This value is not applicable for parameters. **Tags:** atp.EnumerationLiteralIndex=3
standard	This is applicable for all kinds of data elements. For variable data prototypes the 'last is best' semantics applies. For parameter there is no specific implementation directive. **Tags:** atp.EnumerationLiteralIndex=4

Table F.132: SwImplPolicyEnum

Class	**SwRecordLayout**			
Note	Defines how the data objects (variables, calibration parameters etc.) are to be stored in the ECU memory. As an example, this definition specifies the sequence of axis points in the ECU memory. Iterations through axis values are stored within the sub-elements swRecordLayoutGroup. **Tags:** atp.recommendedPackage=SwRecordLayouts			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**SwRecordLayout**			
swRecord LayoutGroup	SwRecordLayoutGroup	0..1	aggr	This is the top level record layout group. **Tags:** atp.Splitkey=swRecordLayoutGroup xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table F.133: SwRecordLayout

Class	**SwRecordLayoutGroup**			
Note	Specifies how a record layout is set up. Using SwRecordLayoutGroup it recursively models iterations through axis values. The subelement swRecordLayoutGroupContentType may reference other Sw RecordLayouts, SwRecordLayoutVs and SwRecordLayoutGroups for the modeled record layout.			
Base	*ARObject*			
Aggregated by	SwRecordLayout.swRecordLayoutGroup, SwRecordLayoutGroupContent.swRecordLayoutGroup			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
category	AsamRecordLayout Semantics	0..1	attr	This attribute denotes the semantics in particular in terms of the corresponding A2L-Keyword. This is to support the mapping of the more general record layouts in AUTOSAR/ MSR to the specific A2l keywords. It is possible to express the specific semantics of A2l recordlayout keywords in swRecordlayoutGroup but not always vice versa. Therefore the mapping is provided in this optional attribute. **Tags:** xml.sequenceOffset=5
desc	MultiLanguageOverview Paragraph	0..1	aggr	This aggregation allows a brief description about the particular record layout group which can help to identify the entry. In-depth documentation should be added to the introduction of the surrounding record layout. **Tags:** atp.Splitkey=desc xml.sequenceOffset=20
shortLabel	Identifier	0..1	attr	This attribute specifies a name which can be used e.g. when ECU code is generated from the record layout group. **Tags:** xml.sequenceOffset=3
swGenericAxis ParamType	SwGenericAxisParam Type	0..1	ref	This association allows to specify record layout groups to iterate over generic axis parameters. For example, if the generic axis parameter is an array, the record layout group will iterate over this array. Obviously, the axis referred to by swRecordLayoutGroup Axis shall be a generic axis in which the referenced Sw GenericAxisType is aggregated. **Tags:** atp.Splitkey=swGenericAxisParamType xml.sequenceOffset=50

▽

△

Class	**SwRecordLayoutGroup**			
swRecord Layout Component	Identifier	0..1	attr	This attribute is used to denote the component to which the group in question applies. Thus, the record layout supports structured objects. This secures independence from the sequence of components, because they can be referred to via name. **Tags:** xml.sequenceOffset=90
swRecord LayoutGroup Axis	AxisIndexType	0..1	attr	This attribute specifies the iteration axis number for a Sw RecordLayoutGroup. The current record layout group then refers exactly to the axis with this number. This means that the values are taken by iterating along the thus referenced axis. **Tags:** xml.sequenceOffset=30
swRecord LayoutGroup ContentType	SwRecordLayoutGroup Content	0..1	aggr	This is the contents of the recordLayout which is produced for every step of iteration. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=100 xml.typeElement=false xml.typeWrapperElement=false
swRecord LayoutGroup From	RecordLayoutIterator Point	0..1	attr	This attribute specifies the iterator index for the point in the axis from which a record layout group is commenced. Negative values are also possible, i.e. the value -4 counts from the fourth value from the end. If this property is missing, the iteration starts with '1'. **Tags:** xml.sequenceOffset=60
swRecord LayoutGroup Index	NameToken	0..1	attr	This attribute attributes a symbolic name to the iterator of the superimposed record layout group. This can be referenced as a loop index in contained SwRecordLayout V elements. **Tags:** xml.sequenceOffset=40
swRecord LayoutGroup Step	Integer	0..1	attr	This attribute specifies the step width for the iterator index that is used for the current record layout group. Note that negative values are also possible, in case of the starting point is higher than the endpoint. If the property is missing, the step width is "1". **Tags:** xml.sequenceOffset=80
swRecord LayoutGroupTo	RecordLayoutIterator Point	0..1	attr	This attribute specifies the end point for the iteration. Negative values are also possible, i.e. the value -4 counts up to the fourth value from the end. If this property is not there, the iteration ends at "-1" which is the last element. Note that depending on the arraySizeSemantics of Sw TextProps the iteration ends at the value specified in sw MaxTextSize. **Tags:** xml.sequenceOffset=70

Table F.134: SwRecordLayoutGroup

Class	SwSystemconst			
Note	This element defines a system constant which serves an input to select a particular variation point. In particular a system constant serves as an operand of the binding function (swSyscond) in a Variation point. Note that the binding process can only happen if a value was assigned to to the referenced system constants. **Tags:** atp.recommendedPackage=SwSystemconsts			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDefProps	SwDataDefProps	0..1	aggr	This denotes the data definition properties of the system constant. This supports to express the limits and optionally a conversion within the internal to physical values by a compu method. **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=40

Table F.135: SwSystemconst

Class	SwSystemconstValue			
Note	This meta-class assigns a particular value to a system constant.			
Base	*ARObject*			
Aggregated by	SwSystemconstantValueSet.swSystemconstantValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
annotation	Annotation	*	aggr	This provides the ability to add information why the value is set like it is. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=30
swSystemconst	SwSystemconst	0..1	ref	This is the system constant to which the value applies. **Tags:** atp.Splitkey=swSystemconst xml.sequenceOffset=10
value	Numerical	0..1	attr	This is the particular value of a system constant. It is specified as Numerical. Further restrictions may apply by the definition of the system constant. The value attribute defines the internal value of the Sw Systemconst as it is processed in the Formula Language. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table F.136: SwSystemconstValue

Class	SwSystemconstantValueSet
Note	This meta-class represents the ability to specify a set of system constant values. **Tags:** atp.recommendedPackage=SwSystemconstantValueSets
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Aggregated by	ARPackage.element



△

Class	SwSystemconstantValueSet			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
sw Systemconstant Value	SwSystemconstValue	*	aggr	This is one particular value of a system constant. **Tags:** atp.Splitkey=swSystemconstantValue

Table F.137: SwSystemconstantValueSet

Class	SwcImplementation			
Note	This meta-class represents a specialization of the general Implementation meta-class with respect to the usage in application software. **Tags:** atp.recommendedPackage=SwcImplementations This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *Implementation*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
behavior	SwcInternalBehavior	0..1	ref	The internal behavior implemented by this Implementation. **Tags:** atp.Splitkey=behavior
perInstance MemorySize	PerInstanceMemory Size	*	aggr	Allows a definition of the size of the per-instance memory for this implementation. The aggregation of PerInstance MemorySize is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects, in this case PerInstanceMemory. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=perInstanceMemorySize, perInstance MemorySize.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
required RTEVendor	String	0..1	attr	Identify a specific RTE vendor. This information is potentially important at the time of integrating (in particular: linking) the application code with the RTE. The semantics is that (if the association exists) the corresponding code has been created to fit to the vendor-mode RTE provided by this specific vendor. Attempting to integrate the code with another RTE generated in vendor mode is in general not possible.

Table F.138: SwcImplementation

Class	SwcInternalBehavior			
Note	The `SwcInternalBehavior` of an `AtomicSwComponentType` describes the relevant aspects of the software-component with respect to the RTE, i.e. the `RunnableEntity`s and the `RTEEvent`s they respond to.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *InternalBehavior*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	*AtomicSwComponentType*.internalBehavior, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**SwcInternalBehavior**			
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	Defines an AUTOSAR typed memory-block that needs to be available for each instance of the SW-component. This is typically only useful if `supportsMultipleInstantiation` is set to "true" or if the component defines NVRAM access via permanent blocks. The aggregation of `arTypedPerInstanceMemory` is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory, arTypedPer InstanceMemory.shortName, arTypedPerInstance Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
event	RTEEvent	*	aggr	This is a `RTEEvent` specified for the particular `SwcInternalBehavior`. The aggregation of `RTEEvent` is subject to variability with the purpose to support the conditional existence of `RTEEvent`s. Note: the number of `RTEEvent`s might vary due to the conditional existence of `PortPrototype`s using `DataReceivedEvent`s or due to different scheduling needs of algorithms. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=event, event.shortName, event.variation Point.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea Policy	SwcExclusiveArea Policy	*	aggr	Options how to generate the ExclusiveArea related APIs. When no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
explicitInter Runnable Variable	VariableDataPrototype	*	aggr	Implement state message semantics for establishing communication among runnables of the same component. The aggregation of explicitInterRunnable Variable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitInterRunnableVariable, explicitInter RunnableVariable.shortName, explicitInterRunnable Variable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
implicitInter Runnable Variable	VariableDataPrototype	*	aggr	Implement state message semantics for establishing communication among runnables of the same component. The aggregation of implicitInterRunnable Variable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=implicitInterRunnableVariable, implicitInter RunnableVariable.shortName, implicitInterRunnable Variable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The `includedDataTypeSet` is used by a software component for its implementation. **Tags:** atp.Splitkey=includedDataTypeSet
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Tags:** atp.Splitkey=includedModeDeclarationGroupSet
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	The purpose of this is that within the context of a given SwComponentType some data def properties of individual instantiations can be modified. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of Port Prototypes and component local memories like "per InstanceParameter" or "arTypedPerInstanceMemory". **Stereotypes:** atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
perInstance Memory	PerInstanceMemory	*	aggr	Defines a per-instance memory object needed by this software component. The aggregation of PerInstance Memory is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=perInstanceMemory, perInstance Memory.shortName, perInstanceMemory.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
perInstance Parameter	ParameterData Prototype	*	aggr	Defines parameter(s) or characteristic value(s) that needs to be available for each instance of the software-component. This is typically only useful if supportsMultipleInstantiation is set to "true". The aggregation of perInstanceParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=perInstanceParameter, perInstance Parameter.shortName, perInstanceParameter.variation Point.shortLabel vh.latestBindingTime=preCompileTime
portAPIOption	PortAPIOption	*	aggr	Options for generating the signature of port-related calls from a runnable to the RTE and vice versa. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=portAPIOption, portAPIOption.port, port APIOption.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
runnable	RunnableEntity	*	aggr	This is a `RunnableEntity` specified for the particular `SwcInternalBehavior`. The aggregation of `RunnableEntity` is subject to variability with the purpose to support the conditional existence of `RunnableEntity`s. Note: the number of `RunnableEntity`s might vary due to the conditional existence of `PortPrototype`s using `DataReceivedEvent`s or due to different scheduling needs of algorithms. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=runnable, runnable.shortName, runnable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
service Dependency	SwcService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item. The aggregation of SwcServiceDependency is subject to variability with the purpose to support the conditional existence of ports as well as the conditional existence of ServiceNeeds. The SwcServiceDependency owned by an SwcInternal Behavior can be located in a different physical file in order to support that SwcServiceDependency might be provided in later development steps or even by different expert domain (e.g OBD expert for Obd related Service Needs) tools. Therefore the aggregation is <<atp Splitable>>. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceDependency, service Dependency.shortName, serviceDependency.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
shared Parameter	ParameterData Prototype	*	aggr	Defines parameter(s) or characteristic value(s) shared between SwComponentPrototypes of the same Sw ComponentType The aggregation of sharedParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=sharedParameter, sharedParameter.short Name, sharedParameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
supports Multiple Instantiation	Boolean	0..1	attr	Indicate whether the corresponding software-component can be multiply instantiated on one ECU. In this case the attribute will result in an appropriate component API on programming language level (with or without instance handle).
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation. **Tags:** atp.Splitkey=variationPointProxy, variationPoint Proxy.shortName

Table F.139: SwcInternalBehavior

Class	**SymbolProps**			
Note	This meta-class represents the ability to attach with the symbol attribute a symbolic name that is conform to C language requirements to another meta-class, e.g. `AtomicSwComponentType`, that is a potential subject to a name clash on the level of RTE source code.			
Base	*ARObject*, *ImplementationProps*, *Referrable*			
Aggregated by	Allocator.namespace, ApApplicationErrorDomain.namespace, *AtomicSwComponentType*.symbolProps, *CppImplementationDataType*.namespace, ImplementationDataType.symbolProps, *Port Interface*.namespace, SecurityEventDefinition.eventSymbolName, TraceSwitchConfig.namespace			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.140: SymbolProps

Class	**SynchronousServerCallPoint**			
Note	This means that the RunnableEntity is supposed to perform a blocking wait for a response from the server. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServerCallPoint*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order. **Tags:** atp.Splitkey=calledFromWithinExclusiveArea

Table F.141: SynchronousServerCallPoint

Enumeration	**TcpRoleEnum**
Note	This enumeration defines whether a TCP node has the tcp server role or the client role.
Aggregated by	StaticSocketConnection.tcpRole
Literal	***Description***
connect	Connects the client to a remote TCP host. **Tags:** atp.EnumerationLiteralIndex=0
listen	Socket is put into the server mode (listen for connections). **Tags:** atp.EnumerationLiteralIndex=1

Table F.142: TcpRoleEnum

Class	**TextValueSpecification**			
Note	The purpose of TextValueSpecification is to define the labels that correspond to enumeration values.			
Base	*ARObject*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, MetaDataItem.metaDataItemType, Nonqueued ReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, Nonqueued SenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInit Value, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.init Value, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, Persistency KeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, Record ValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalid Value, VariableDataPrototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
value	VerbatimString	0..1	attr	This is the value itself. Note that vt uses the | operator to separate the values for the different bitfield masks in case that the semantics of the related DataPrototype is described by means of a BITFIELD_TEXTTABLE in the associated CompuMethod.

Table F.143: TextValueSpecification

Class	***TransformationComSpecProps*** (abstract)			
Note	`TransformationComSpecProps` holds all the attributes for transformers that are port specific.			
Base	*ARObject*, *Describable*			
Subclasses	EndToEndTransformationComSpecProps, UserDefinedTransformationComSpecProps			
Aggregated by	ClientComSpec.transformationComSpecProps, *ReceiverComSpec*.transformationComSpecProps, ServerComSpec.transformationComSpecProps			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table F.144: TransformationComSpecProps

Class	**TransmissionComSpecProps**
Note	This meta-class defines a set of transmission attributes which the application software is assumed to implement.
Base	*ARObject*
Aggregated by	*SenderComSpec*.transmissionProps

▽

△

Class	**TransmissionComSpecProps**			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
dataUpdate Period	TimeValue	0..1	attr	This attribute defines the period in which the application is assumed to transmit the respective data.
minimumSend Interval	TimeValue	0..1	attr	This attribute defines the minimum interval between two consecutive transmissions of the respective data the application is assumed to ensure.
onChangeData Prototype	DataPrototype Reference	*	aggr	This reference defines which DataPrototypes trigger the onChange transmission of the data. **Tags:** atp.Splitkey=onChangeDataPrototype
transmission Mode	TransmissionMode DefinitionEnum	0..1	attr	The attribute defines the mode in which the application is assumed to transmit the respective data.

Table F.145: TransmissionComSpecProps

Class	**Trigger**			
Note	A trigger which is provided (i.e. released) or required (i.e. used to activate something) in the given context.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.releasedTrigger, BswModuleDescription.requiredTrigger, ServiceInterface.trigger, TriggerInterface.trigger			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers. This Attribute is only used by the AUTOSAR Classic Platform.
triggerPeriod	MultidimensionalTime	0..1	aggr	Optional definition of a period in case of a periodically (time or angle) driven external trigger. **Tags:** atp.Splitkey=triggerPeriod This Attribute is only used by the AUTOSAR Classic Platform.

Table F.146: Trigger

Class	**TriggerInterface**			
Note	A trigger interface declares a number of triggers that can be sent by an trigger source. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
trigger	Trigger	*	aggr	The Trigger of this trigger interface. **Tags:** atp.Splitkey=trigger, trigger.shortName

Table F.147: TriggerInterface

Class	***ValueSpecification*** (abstract)			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	*ARObject*			
Subclasses	*AbstractRuleBasedValueSpecification*, ApplicationValueSpecification, *CompositeValueSpecification*, ConstantReference, NotAvailableValueSpecification, NumericalValueSpecification, ReferenceValue Specification, TextValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValue Specification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoService Key.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElement Condition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.init Value, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, Nv ProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table F.148: ValueSpecification

Class	**VariableAccess**			
Note	The presence of a `VariableAccess` implies that a `RunnableEntity` needs access to a `VariableDataPrototype`. The kind of access is specified by the role in which the class is used.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *ReceiverComSpec*.replaceWith, RunnableEntity.dataReadAccess, Runnable Entity.dataReceivePointByArgument, RunnableEntity.dataReceivePointByValue, RunnableEntity.data SendPoint, RunnableEntity.dataWriteAccess, RunnableEntity.readLocalVariable, RunnableEntity.written LocalVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
accessed Variable	AutosarVariableRef	0..1	aggr	This denotes the accessed variable. **Tags:** atp.Splitkey=accessedVariable
scope	VariableAccessScope Enum	0..1	attr	This attribute allows for constraining the scope of the corresponding communication. For example, it possible to express whether the communication is intended to cross the boundary of an ECU or whether it is intended not to cross the boundary of a single partition.

Table F.149: VariableAccess

Class	**VariableAndParameterInterfaceMapping**			
Note	Defines the mapping of `VariableDataPrototype`s or `ParameterDataPrototype`s in context of two different `SenderReceiverInterface`s, `NvDataInterface`s or `ParameterInterface`s.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

▽

△

Class	**VariableAndParameterInterfaceMapping**			
dataMapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular VariableData Prototypes or ParameterDataPrototypes with unequal names and/or unequal semantic (resolution or range) in context of two different SenderReceiverInterfaces, Nv DataInterfaces or ParameterInterfaces **Tags:** atp.Splitkey=dataMapping

Table F.150: VariableAndParameterInterfaceMapping

Class	**VariableDataPrototype**			
Note	A `VariableDataPrototype` represents a formalized generic piece of information that is typically mutable by the application software layer. `VariableDataPrototype` is used in various contexts and the specific context gives the otherwise generic `VariableDataPrototype` a dedicated semantics.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	ApplicationInterface.indication, *AtpClassifier*.atpFeature, BswInternalBehavior.arTypedPerInstance Memory, BswModuleDescription.providedData, BswModuleDescription.requiredData, BulkNvData Descriptor.bulkNvBlock, DiagnosticSovdAccessArgument.contentObject, *InternalBehavior*.staticMemory, NvBlockDescriptor.ramBlock, NvDataInterface.nvData, SenderReceiverInterface.dataElement, Service Interface.event, SwcInternalBehavior.arTypedPerInstanceMemory, SwcInternalBehavior.explicitInter RunnableVariable, SwcInternalBehavior.implicitInterRunnableVariable			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype **Tags:** atp.Splitkey=initValue

Table F.151: VariableDataPrototype

Class	**VariationPoint**			
Note	This meta-class represents the ability to express a "structural variation point". The container of the variation point is part of the selected variant if swSyscond evaluates to true and each postBuildVariant Criterion is fulfilled.			
Base	*ARObject*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
blueprint Condition	DocumentationBlock	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint. Note that variationPoints are not allowed within a blueprintCondition. **Tags:** xml.sequenceOffset=28
desc	MultiLanguageOverview Paragraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:** xml.sequenceOffset=20
formalBlueprint Generator	BlueprintGenerator	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint by using ARMQL. Note that variationPoints are not allowed within a formal BlueprintGenerator. **Tags:** atp.Status=draft xml.sequenceOffset=30

▽

△

Class	**VariationPoint**			
postBuildVariant Condition	PostBuildVariant Condition	*	aggr	This is the set of post build variant conditions which all shall be fulfilled in order to (postbuild) bind the variation point. **Tags:** xml.sequenceOffset=40
sdg	Sdg	0..1	aggr	An optional special data group is attached to every variation point. These data can be used by external software systems to attach application specific data. For example, a variant management system might add an identifier, an URL or a specific classifier. **Tags:** xml.sequenceOffset=50
shortLabel	Identifier	0..1	attr	This provides a name to the particular variation point to support the RTE generator. It is necessary for supporting splitable aggregations and if binding time is later than codeGenerationTime, as well as some RTE conditions. It needs to be unique with in the enclosing Identifiables with the same ShortName. **Stereotypes:** atpIdentityContributor **Tags:** xml.sequenceOffset=10
swSyscond	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. Note that the multiplicity is 0..1 in order to support pure postBuild variants. **Tags:** xml.sequenceOffset=30

Table F.152: VariationPoint

Class	**ViewMap**			
Note	The ViewMap allows to relate any number of elements on the "first" side to any number of elements on the "second" side. Since the ViewMap does not address a specific mapping use-case the roles "first" and "second" shall imply this generality. This mapping allows to trace transformations of artifacts within the AUTOSAR environment. The references to the mapped elements can be plain references and/or InstanceRefs.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ViewMapSet.viewMap			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
firstElement	Referrable	*	ref	Reference to identifiable elements on the first "side". **Tags:** atp.Splitkey=firstElement xml.sequenceOffset=20
firstElement Instance	AtpFeature	*	iref	InstanceRefs to elements on the first "side". **Tags:** atp.Splitkey=firstElementInstance, firstElement Instance.contextElement, firstElementInstance.target xml.sequenceOffset=50 **InstanceRef implemented by:** AnyInstanceRef
role	Identifier	0..1	attr	This attribute is used to describe specific mapping scenarios, e.g. the mappings: • AR_AbstractSystemDescription_SystemDescription • AR_SystemDescription_SystemExtract **Tags:** xml.sequenceOffset=10

▽

△

Class	**ViewMap**			
secondElement	Referrable	*	ref	Reference to identifiable elements on the second "side". **Tags:** atp.Splitkey=secondElement xml.sequenceOffset=30
secondElement Instance	AtpFeature	*	iref	InstanceRefs to elements on the second "side". **Tags:** atp.Splitkey=secondElementInstance, secondElement Instance.contextElement, secondElementInstance.target xml.sequenceOffset=60 **InstanceRef implemented by:** AnyInstanceRef

Table F.153: ViewMap

Class	**ViewMapSet**			
Note	Collection of ViewMaps that are used to establish relationships between different AUTOSAR artifacts. **Tags:** atp.recommendedPackage=ViewMapSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
viewMap	ViewMap	*	aggr	ViewMaps that are collected by the ViewMapSet. **Tags:** atp.Splitkey=viewMap, viewMap.shortName

Table F.154: ViewMapSet

G Non-Splitable Elements in this Template

This chapter contains a table of all model elements which are **not splitable** in the scope of this document.

Note for R26-11:

The life cycle of the stereotype ≪atpSplitable≫ will end with the AUTOSAR Release R26-11. Any remaining references to the stereotype ≪atpSplitable≫ in this document will be removed in release R27-05.

Name of non splitable element
BinaryManifestResource.item
BinaryManifestResourceDefinition.itemDefinition
ComponentInSystemInstanceRef.base
ComponentInSystemInstanceRef.contextComponent
ComponentInSystemInstanceRef.contextComposition
ComponentInSystemInstanceRef.targetComponent
CouplingElementSwitchDetails.streamFilter
CouplingElementSwitchDetails.switchStreamIdentification
CouplingPortScheduler.predecessor
DataPrototypeInClientServerInterfaceInstanceRef.base
DataPrototypeInClientServerInterfaceInstanceRef.contextDataPrototypeInCs
DataPrototypeInClientServerInterfaceInstanceRef.rootDataPrototypeInCs
DataPrototypeInClientServerInterfaceInstanceRef.targetDataPrototypeInCs
DataPrototypeInPortInterfaceInstanceRef.abstractBase
DataPrototypeInPortInterfaceInstanceRef.contextDataPrototype
DataPrototypeInPortInterfaceInstanceRef.rootDataPrototype
DataPrototypeInPortInterfaceInstanceRef.targetDataPrototype
DataPrototypeInSenderReceiverInterfaceInstanceRef.base
DataPrototypeInSenderReceiverInterfaceInstanceRef.contextDataPrototypeInSr
DataPrototypeInSenderReceiverInterfaceInstanceRef.rootDataPrototypeInSr
DataPrototypeInSenderReceiverInterfaceInstanceRef.targetDataPrototypeInSr
DataPrototypeTransformationProps.ident
DataTransformation.transformerChain
ImplementationDataTypeElementInPortInterfaceRef.contextImplementationDataElement
LinSlaveConfig.ident
LinSporadicFrame.substitutedFrame
MacMulticastConfiguration.macMulticastGroup
ModeDeclarationGroupPrototypeInSystemInstanceRef.base
ModeDeclarationGroupPrototypeInSystemInstanceRef.contextComponent
ModeDeclarationGroupPrototypeInSystemInstanceRef.contextComposition
ModeDeclarationGroupPrototypeInSystemInstanceRef.contextPort
ModeDeclarationGroupPrototypeInSystemInstanceRef.targetModeDeclarationGroupPrototype
OperationInSystemInstanceRef.base
OperationInSystemInstanceRef.contextComponent

▽

△

Name of non splitable element
OperationInSystemInstanceRef.contextComposition
OperationInSystemInstanceRef.contextPort
OperationInSystemInstanceRef.targetOperation
ParameterDataPrototypeInSystemInstanceRef.base
ParameterDataPrototypeInSystemInstanceRef.contextComponent
ParameterDataPrototypeInSystemInstanceRef.contextComposition
ParameterDataPrototypeInSystemInstanceRef.contextPort
ParameterDataPrototypeInSystemInstanceRef.targetParameterDataPrototype
PermissibleSignalPath.physicalChannel
PncMapping.ident
PortGroupInSystemInstanceRef.base
PortGroupInSystemInstanceRef.contextComponent
PortGroupInSystemInstanceRef.contextComposition
PortGroupInSystemInstanceRef.target
RteEventInCompositionInstanceRef.base
RteEventInCompositionInstanceRef.contextSwComponent
RteEventInCompositionInstanceRef.targetRteEvent
RteEventInSystemInstanceRef.base
RteEventInSystemInstanceRef.contextRootComposition
RteEventInSystemInstanceRef.contextSwComponent
RteEventInSystemInstanceRef.targetRteEvent
TpConnection.ident
TransformationISignalProps.ident
TriggerInSystemInstanceRef.base
TriggerInSystemInstanceRef.contextComponent
TriggerInSystemInstanceRef.contextComposition
TriggerInSystemInstanceRef.contextPort
TriggerInSystemInstanceRef.targetTrigger
VariableDataPrototypeInSystemInstanceRef.base
VariableDataPrototypeInSystemInstanceRef.contextComponent
VariableDataPrototypeInSystemInstanceRef.contextComposition
VariableDataPrototypeInSystemInstanceRef.contextPort
VariableDataPrototypeInSystemInstanceRef.targetDataPrototype

Table G.1: Usage of non splitable elements

H Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped ≪atpVariation≫ in the scope of this document.

Each entry in following Table consists of the identification of the model element itself and the applicable value of the tagged value vh.latestBindingTime.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to [1].

Variation Point	*Latest Binding Time*
AbstractCanCluster	postBuild
AbstractCanCommunicationController	postBuild
AbstractMacSecCommunication.couplingPort	postBuild
AbstractMacSecCommunication.vlan	postBuild
AbstractMessageToSecureChannelMapping.canFrameTriggering	postBuild
AbstractMessageToSecureChannelMapping.soConIPduIdentifier	postBuild
AbstractServiceInstance.capabilityRecord	postBuild
AbstractServiceInstance.methodActivationRoutingGroup	postBuild
AliasNameSet.aliasName	preCompileTime
BusMirrorChannel.channel	systemDesignTime
BusMirrorChannelMapping.ecuInstance	systemDesignTime
BusMirrorChannelMapping.globalTimeDomain	systemDesignTime
BusMirrorChannelMapping.targetPduTriggering	postBuild
CanCluster	postBuild
CanCommunicationController	postBuild
CanSecConnectivityAssociation.canCommunicationConnector	postBuild
CanSecConnectivityAssociation.secureChannel	postBuild
CanSecSecureChannel.txCommunicationConnector	postBuild
CanTpConfig.tpAddress	postBuild
CanTpConfig.tpChannel	postBuild
CanTpConfig.tpConnection	postBuild
CanTpConfig.tpEcu	postBuild
CanTpConfig.tpNode	postBuild
ClientIdDefinitionSet.clientIdDefinition	postBuild
ClientIdRange.lowerLimit	preCompileTime
ClientIdRange.upperLimit	preCompileTime
CommunicationCluster	postBuild
CommunicationCluster.physicalChannel	systemDesignTime
CommunicationConnector.ecuCommPortInstance	postBuild
CommunicationConnector.explicitWakeupChannel	postBuild
CommunicationConnector.explicitWakeupPnc	postBuild
CommunicationController	postBuild
ConsumedEventGroup.eventMulticastAddress	postBuild
ConsumedEventGroup.sdClientTimerConfig	postBuild

▽

△

Variation Point	*Latest Binding Time*
ConsumedProvidedServiceInstanceGroup.consumedServiceInstance	postBuild
ConsumedProvidedServiceInstanceGroup.providedServiceInstance	postBuild
ConsumedServiceInstance.allowedServiceProvider	postBuild
ConsumedServiceInstance.consumedEventGroup	postBuild
ConsumedServiceInstance.eventMulticastSubscriptionAddress	postBuild
ConsumedServiceInstance.localUnicastAddress	postBuild
ConsumedServiceInstance.remoteUnicastAddress	postBuild
ConsumedServiceInstance.sdClientTimerConfig	postBuild
CouplingElement.couplingElementDetails	postBuild
CouplingElement.couplingPort	postBuild
CouplingPort.macAddressVlanAssignment	postBuild
CouplingPortConnection.nodePort	postBuild
CpSoftwareCluster.swComponentAssignment	postBuild
CpSoftwareCluster.swComposition	systemDesignTime
CpSoftwareClusterMappingSet.portElementToComResourceMapping	postBuild
CpSoftwareClusterMappingSet.resourceToApplicationPartitionMapping	systemDesignTime
CpSoftwareClusterMappingSet.softwareClusterToResourceMapping	preCompileTime
CpSoftwareClusterMappingSet.swcToApplicationPartitionMapping	postBuild
CpSoftwareClusterToEcuInstanceMapping.swCluster	systemDesignTime
DataTransformationSet.dataTransformation	codeGenerationTime
DataTransformationSet.transformationTechnology	codeGenerationTime
DdsCpConsumedServiceInstance.consumedEvent	postBuild
DdsCpConsumedServiceInstance.consumedField	postBuild
DdsCpConsumedServiceInstance.consumedOperation	postBuild
DdsCpConsumedServiceInstance.localUnicastAddress	systemDesignTime
DdsCpConsumedServiceInstance.staticRemoteMulticastAddress	systemDesignTime
DdsCpConsumedServiceInstance.staticRemoteUnicastAddress	systemDesignTime
DdsCpProvidedServiceInstance.localUnicastAddress	systemDesignTime
DdsCpProvidedServiceInstance.providedEvent	postBuild
DdsCpProvidedServiceInstance.providedField	postBuild
DdsCpProvidedServiceInstance.providedOperation	postBuild
DdsCpProvidedServiceInstance.staticRemoteMulticastAddress	systemDesignTime
DdsCpProvidedServiceInstance.staticRemoteUnicastAddress	systemDesignTime
DdsTransformationISignalProps	postBuild
DltConfig.globalTimeDomain	systemDesignTime
EcuInstance.associatedConsumedProvidedServiceInstanceGroup	postBuild
EcuInstance.commController	postBuild
EcuInstance.connector	postBuild
EndToEndTransformationISignalProps	postBuild
EthernetCluster	postBuild
EthernetCluster.couplingPortConnection	postBuild
EthernetCommunicationController	postBuild
EventHandler.eventMulticastAddress	postBuild

▽

△

Variation Point	*Latest Binding Time*
EventHandler.sdServerEgTimingConfig	postBuild
FlatMap.instance	postBuild
FlexrayCluster	postBuild
FlexrayCommunicationController	postBuild
FlexrayTpConfig.pduPool	postBuild
FlexrayTpConfig.tpAddress	postBuild
FlexrayTpConfig.tpConnection	postBuild
FlexrayTpConfig.tpConnectionControl	postBuild
FlexrayTpConfig.tpEcu	postBuild
FlexrayTpConfig.tpNode	postBuild
Frame.pduToFrameMapping	postBuild
FrameTriggering.pduTriggering	postBuild
Gateway.frameMapping	postBuild
Gateway.iPduMapping	postBuild
Gateway.signalMapping	postBuild
GlobalTimeDomain.clonedFromTimeDomain	postBuild
GlobalTimeDomain.domainId	postBuild
GlobalTimeDomain.gateway	postBuild
GlobalTimeDomain.globalTimeDomainProperty	postBuild
GlobalTimeDomain.globalTimeMaster	postBuild
GlobalTimeDomain.globalTimeSubDomain	postBuild
GlobalTimeDomain.icvSecureComProps	postBuild
GlobalTimeDomain.pduTriggering	postBuild
GlobalTimeDomain.slave	postBuild
IEEE1722TpAcfBus.acfPart	postBuild
IEEE1722TpAcfConnection.acfTransportedBus	postBuild
IEEE1722TpConfig.tpConnection	postBuild
IEEE1722TpConnection.globalTimeDomain	systemDesignTime
ISignal.dataTransformation	codeGenerationTime
ISignalGroup.comBasedSignalGroupTransformation	codeGenerationTime
ISignalIPdu.iPduTimingSpecification	postBuild
ISignalIPdu.iSignalToPduMapping	postBuild
ISignalIPduGroup.iSignalIPdu	postBuild
ISignalIPduGroup.nmPdu	postBuild
J1939Cluster	postBuild
J1939Node.pduTriggering	postBuild
J1939ProtectedIPdu.payload	postBuild
J1939TpConfig.tpAddress	postBuild
J1939TpConfig.tpConnection	postBuild
J1939TpConfig.tpNode	postBuild
LinCluster	postBuild
LinCommunicationController	postBuild
LinMaster	postBuild

▽

△

Variation Point	*Latest Binding Time*
LinPhysicalChannel.scheduleTable	postBuild
LinSlave	postBuild
LinTpConfig.tpAddress	postBuild
LinTpConfig.tpConnection	postBuild
LinTpConfig.tpNode	postBuild
McGroupDataRefSet	preCompileTime
MessageToBypassCanSecCommunicationMapping.canCommunicationConnector	postBuild
MessageToCanSecSecureChannelMapping.secureChannel	postBuild
MultiplexedIPdu.dynamicPart	postBuild
MultiplexedIPdu.staticPart	postBuild
NmCluster.nmNode	postBuild
NmConfig.nmCluster	postBuild
NmConfig.nmClusterCoupling	postBuild
NmConfig.nmIfEcu	preCompileTime
PdurIPduGroup.iPdu	postBuild
PduTriggering.iSignalTriggering	postBuild
PduTriggering.secOcCryptoServiceMapping	postBuild
PhysicalChannel.commConnector	postBuild
PhysicalChannel.frameTriggering	postBuild
PhysicalChannel.iSignalTriggering	postBuild
PhysicalChannel.pduTriggering	postBuild
PncMapping.pncConsumedProvidedServiceInstanceGroup	postBuild
ProvidedServiceInstance.allowedServiceConsumer	postBuild
ProvidedServiceInstance.eventHandler	postBuild
ProvidedServiceInstance.localUnicastAddress	postBuild
ProvidedServiceInstance.remoteMulticastSubscriptionAddress	postBuild
ProvidedServiceInstance.remoteUnicastAddress	postBuild
ProvidedServiceInstance.sdServerTimerConfig	postBuild
ServiceInstanceCollectionSet.serviceInstance	postBuild
SoAdConfig.socketAddress	postBuild
SocketAddress.staticSocketConnection	postBuild
SomeipTpConnection.tpConcurrentProcessingSdu	postBuild
SOMEIPTransformationISignalProps	postBuild
StaticSocketConnection.iPduIdentifier	postBuild
StaticSocketConnection.remoteAddress	postBuild
System.fibexElement	postBuild
System.j1939SharedAddressCluster	postBuild
System.mapping	postBuild
System.rootSoftwareComposition	systemDesignTime
System.swCluster	systemDesignTime
System.systemDocumentation	systemDesignTime
SystemMapping.applicationPartitionToEcuPartitionMapping	postBuild
SystemMapping.comManagementMapping	systemDesignTime

▽

△

Variation Point	*Latest Binding Time*
SystemMapping.cryptoServiceMapping	postBuild
SystemMapping.dataMapping	postBuild
SystemMapping.ddsISignalToTopicMapping	postBuild
SystemMapping.ecuPartitionToCoreMapping	systemDesignTime
SystemMapping.ecuResourceMapping	systemDesignTime
SystemMapping.j1939ControllerApplicationToJ1939NodeMapping	preCompileTime
SystemMapping.mappingConstraint	systemDesignTime
SystemMapping.pncMapping	systemDesignTime
SystemMapping.portElementToComResourceMapping	postBuild
SystemMapping.resourceEstimation	systemDesignTime
SystemMapping.resourceToApplicationPartitionMapping	systemDesignTime
SystemMapping.signalPathConstraint	systemDesignTime
SystemMapping.softwareClusterToApplicationPartitionMapping	systemDesignTime
SystemMapping.softwareClusterToResourceMapping	preCompileTime
SystemMapping.swClusterMapping	systemDesignTime
SystemMapping.swcToApplicationPartitionMapping	postBuild
SystemMapping.swImplMapping	preCompileTime
SystemMapping.swMapping	preCompileTime
SystemMapping.systemSignalGroupToComResourceMapping	systemDesignTime
SystemMapping.systemSignalToComResourceMapping	systemDesignTime
TransformationISignalProps	postBuild
TransformationTechnology.transformationDescription	postBuild
UserDefinedCluster	postBuild
UserDefinedCommunicationController	postBuild
UserDefinedTransformationISignalProps	postBuild

Table H.1: Usage of variation points