Document Title	Specification of I-PDU Multiplexer
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	182

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Document Change History

Date	Release	Changed by	Change Description
2018-10-31	4.4.0	AUTOSAR Release Management	• Introduce priority for Tx ContainedI-Pdus with LastIsBest collection semantics • Header File Cleanup • Limitations on Container PDU with MDT • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2017-12-08	4.3.1	AUTOSAR Release Management	• introduced static multiple-PDU-to-Container handling as draft • revised error definitions • minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• updated TX-confirmation handling • added support for MetaData • Com-Stack API harmonization • 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	• Added Multiple PDU to Container Mapping • Extension of IpduMSelectorField-Length

Document Change History

Date	Release	Changed by	Change Description
2014-03-31	4.1.3	AUTOSAR Release Management	• Editorial changes and minor corrections • No major functional change
2013-10-31	4.1.2	AUTOSAR Release Management	• Revised configuration structure of dynamic and static segments to enforce layout constraints already by the configuration structure • Few bug fixes and clarifications • Editorial changes • Removed chapter(s) on change documentation
2013-03-15	4.1.1	AUTOSAR Administration	• Reworked according to the new SWS_BSWGeneral, harmonization of post-build configuration • Allowing reception of nothing but the static part
2011-12-22	4.0.3	AUTOSAR Administration	• Minor bug fixes and editorial changes • Added configurable JIT-update

Document Change History

Date	Release	Changed by	Change Description
2010-09-30	3.1.5	AUTOSAR Administration	• Updated: tables for mandatory and optional interfaces, SWS_IpduM_00020, SWS_IpduM_00027, SWS_IpduM_00028, SWS_IpduM_00032, SWS_IpduM_00060, SWS_IpduM_00068, SWS_IpduM_00083, SWS_IpduM_00104, ECUC_IpduM_00112, IPDUM117_Conf, SWS_IpduM_00143 and IPDUM162 • Removed: IPDUM013, IPDUM030, IPDUM050_Conf, IPDUM051_Conf, IPDUM063, IPDUM064, IPDUM065, IPDUM072, IPDUM099 and IPDUM154 • Added: pre-compile configuration variant (Chapter 10), ECUC_IpduM_00162, ECUC_IpduM_00163, ECUC_IpduM_00164 and SWS_IpduM_00165
2010-02-02	3.1.4	AUTOSAR Administration	• Harmonization of FIBEX multiplexing and AUTOSAR multiplexing • Many small corrections based on conformance tests and validation activities • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Fixed generated figures and captions
2007-12-21	3.0.1	AUTOSAR Administration	• SWS improvements by AUTOSAR Technical Office • Defined maximum I-PDU size for FlexRay to 254 bytes • Document meta information extended • Small layout adaptations made

Document Change History			
Date	**Release**	**Changed by**	**Change Description**
2007-01-24	2.1.15	AUTOSAR Administration	• Integrated into BSW Scheduler header file structure • Sequence diagrams clarified • Superfluous text removed • Maximum IPDU size clarified • Signature for IpduM_Transmit made consistent with rest of stack. • “Advice for users” revised • Revision Information” added • Legal disclaimer revised
2006-05-16	2.0	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 and functional overview 9
2 Acronyms and abbreviations 10
3 Related documentation 11
3.1 Input documents 11
3.2 Related standards and norms 11
3.3 Related specification 11
4 Constraints and assumptions 12
4.1 Limitations 12
4.2 Applicability to car domains 12
4.3 Applicability to safety related environments 12
5 Dependencies to other modules 13
5.1 AUTOSAR OS 13
5.2 RTE (BSW Scheduler) 13
5.3 PDU Router 14
5.4 COM 14
5.5 File structure 15
5.5.1 Code file structure 15
6 Requirements traceability 16
7 Functional specification 21
7.1 General 21
7.2 I-PDU Multiplexing 21
7.2.1 Definitions and Layout 21
7.2.2 General 23
7.2.3 Initialization 24
7.2.4 Transmission 24
7.2.5 Reception 28
7.2.6 Metadata handling 28
7.3 Multiple-PDU-to-Container handling 28
7.3.1 Dynamic Container Layout 29
7.3.2 Static Container Layout 30
7.3.3 Transmission 30
7.3.4 Transmission of Dynamic Containers 33
7.3.5 Transmission of Static Containers 35
7.3.6 Reception 36
7.3.7 Reception of Dynamic Containers 37
7.3.8 Reception of Static Containers 38
7.3.9 Errorhandling 38
7.3.10 Metadata handling 39
7.4 Error classification 39
7.4.1 Development Errors 39
7.4.2 Runtime Errors 40
7.4.3 Transient Faults 40
7.4.4 Production Errors 40

7.4.5 Extended Production Errors 40
7.5 Error detection and notification 41
8 API specification 42
8.1 Imported types 42
8.2 Type definitions 42
8.2.1 IpduM_ConfigType 42
8.3 Function definitions 42
8.3.1 IpduM_Init 42
8.3.2 IpduM_GetVersionInfo 43
8.3.3 IpduM_Transmit 43
8.4 Call-back notifications 44
8.4.1 IpduM_RxIndication 44
8.4.2 IpduM_TxConfirmation 44
8.4.3 IpduM_TriggerTransmit 45
8.5 Scheduled functions 46
8.6 Expected Interfaces 47
8.6.1 Mandatory Interfaces 47
8.6.2 Optional Interfaces 47
8.6.3 Configurable interfaces 48
9 Sequence diagrams 49
9.1 Transmission of a multiplexed I-PDU and Transmit confirmation 49
9.2 Transmission of a multiplexed I-PDU without Trigger 51
9.3 Reception of the multiplexed I-PDU 51
9.4 Trigger Transmit 53
10 Configuration specification 54
10.1 How to read this chapter 54
10.2 Containers and configuration parameters 54
10.2.1 Configuration overview 54
10.2.2 IpduM 55
10.2.3 IpduMConfig 55
10.2.4 IpduMGeneral 56
10.2.5 IpduMTxPathway 60
10.2.6 IpduMTxRequest 60
10.2.7 IpduMTxDynamicPart 63
10.2.8 IpduMTxDynamicSegment 64
10.2.9 IpduMTxStaticPart 65
10.2.10 IpduMTxStaticSegment 67
10.2.11 IpduMRxPathway 68
10.2.12 IpduMRxIndication 68
10.2.13 IpduMRxDynamicPart 70
10.2.14 IpduMRxDynamicSegment 71
10.2.15 IpduMRxStaticPart 72
10.2.16 IpduMRxStaticSegment 72
10.2.17 IpduMSelectorField 73
10.2.18 IpduMContainerRxPdu 75
10.2.19 IpduMContainedRxPdu 78
10.2.20 IpduMContainerTxPdu 82
10.2.21 IpduMContainedTxPdu 85

10.3 Published Information 89
10.3.1 IpduMPublishedInformation 89
10.4 Configuration Rules 90
10.4.1 Selector Field 90
10.4.2 Placement of static and dynamic parts 90
10.4.3 Multiple PDU to Container Mapping 90
10.4.4 Priority 91
11 Not applicable requirements 92

1 Introduction and functional overview

This specification describes the functionality, APIs and the configuration of the AUTOSAR Basic Software module I-PDU Multiplexer IpduM.

PDU multiplexing means using the same PCI (Protocol Control Information) of a PDU (Protocol Data Unit) with more than one unique layout of its SDU (Service Data Unit). A selector field is a piece of the SDU of the multiplexed PDU. It is used to distinguish the contents of the multiplexed PDUs from each other.

Multiplexing of PDUs is currently known from CAN, but is not restricted to this communication system.

On sender-side, the I-PDU Multiplexer module is responsible to combine appropriate I-PDUs from COM to new, multiplexed I-PDUs and send them back to the PDU Router. On receiver-side, it is responsible to interpret the content of multiplexed I-PDUs and provide COM with its appropriate separated I-PDUs taking into account the value of the selector field.

2 Acronyms and abbreviations

Abbreviation / Acronym:	***Description:***
COM I-PDU	I-PDU assembled in the COM module out of COM Signals
contained I-PDU	I-PDU assembled into or extracted from a Container PDU
Container PDU	PDU containing I-PDUs and headers
dynamic part	see [4]
instance of an I-PDU	IpduM I-PDU with one specific layout and content
Instances of a Container	Instances of the same Container PDU
IpduM	I-PDU Multiplexer
IpduM I-PDU	I-PDU assembled in the IpduM module out of two COM I-PDUs
multiplexed I-PDU	see IpduM I-PDU
segment	The static or dynamic part may consist of more than one piece. These pieces are called segments. See also Chapter 7.2.1 and **Figure 2**.
selector field	see [4]
signal	see [5]
signal group	see [5]
static part	see [4]

3 Related documentation

3.1 Input documents

[1] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

[2] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral.pdf

[3] Specification of RTE
AUTOSAR_SWS_RTE.pdf

[4] Requirements on I-PDU Multiplexer
AUTOSAR_SRS_IPDUMultiplexer.pdf

[5] Specification of Communication
AUTOSAR_SWS_COM.pdf

[6] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

3.2 Related standards and norms

None

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [6], which is also valid for IPDU Multiplexer.

Thus, the specification SWS BSW General shall be considered as additional and required specification for IPDU Multiplexer.

4 Constraints and assumptions

4.1 Limitations

For transmission of multiplexed I-PDUs, minimum delay time observation cannot be taken into account. For more details, see [5] and 7.2.4.1.

For transmission of container PDUs with static layout, minimum delay time cannot be ensured if two or more contained PDUs have each MDT configuration.

4.2 Applicability to car domains

No restrictions.

4.3 Applicability to safety related environments

This document has been created in absence of a safety case and a safety plan. Thus, the direct results of this document can only be used within safety relevant systems after repeating certain process steps as required in the IEC 61508.

5 Dependencies to other modules

This chapter lists all the features from other modules that are used by the AUTOSAR IpduM and functionalities that are provided by AUTOSAR IpduM to other modules. Because the IpduM module deals with PDUs that are either sourced or sunk by other modules, care must be taken that shared configuration items are consistent between the modules.

The IpduM is arranged next to the PDU Router in the layered architecture of AUTOSAR; see [1] and **Figure 1**.

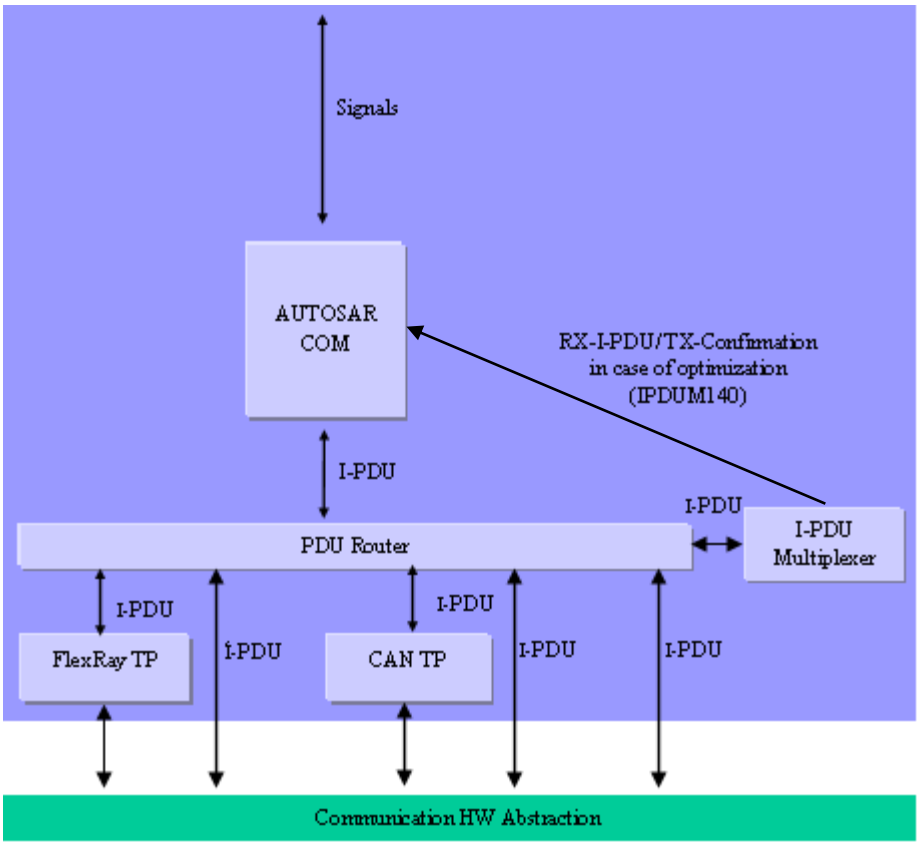

Figure 1 I-PDU Multiplexer in the AUTOSAR Architecture

5.1 AUTOSAR OS

[SWS_IpduM_00107] ⌈The IpduM shall not directly access the AUTOSAR OS.⌋ (SRS_BSW_00429)

5.2 RTE (BSW Scheduler)

The RTE includes the BSW-Scheduler (see [3]).

The IpduM module relies on the BSW-scheduler calling IpduM_MainFunctionRx and IpduM_MainFunctionTx at a period as configured in IpduMRxTimeBase or IpduMTx-TimeBase respectively.

5.3 PDU Router

The following summarizes the functionality IpduM needs from the PDU Router (for more details see Chapter 8.6):

- indication of incoming multiplexed or contained I-PDUs
- sending interface for outgoing I-PDUs (Container or Multiplexed PDUs)
- confirmation of I-PDUs which went out

The following list summarizes the functionality provided by the IpduM module for the PDU Router module:

- indication interface for incoming I-PDUs, which are de-multiplexed and for incoming Container-PDUs, which are to be disassembled
- sending interface for to be multiplexed I-PDUs and I-PDUs, which are to be assembled into a Container PDU
- confirmation interface for transmitted I-PDUs

The configuration of the PDU Router module (e.g. look-up tables) must be such that the I-PDUs, which belong to multiplexed I-PDUs and represent a static or a dynamic part of a multiplexed I-PDU, are routed to the IpduM module.

The configuration of the PDU-Router module (e.g. look-up tables) must be such that the relevant I-PDUs are routed to IpduM. These are:

- I-PDUs, which belong to multiplexed I-PDUs and represent a static or a dynamic part of a multiplexed I-PDU
- I-PDUs, which consist of static and dynamic parts to be de-multiplexed
- I-PDUs, which are to be assembled into a Container PDU
- Container PDUs to be disassembled

5.4 COM

The configuration of the IpduM module relies on a corresponding configuration of the AUTOSAR COM module. For each multiplexed I-PDU, there need to be different I-PDUs for the static part and each layout of the dynamic part. For further information configured in the COM module, see Chapter 7.1 and especially **Figure 2**.
The IpduM further assumes that the correct selector field values are already contained in the COM's modules I-PDU representing the dynamic parts. See also SWS_IpduM_00098.
The configuration of Container PDUs/ contained I-PDUs does not depend on the COM configuration.

5.5 File structure

5.5.1 Code file structure

This IpduM SWS does not define the code file structure completely.

6 Requirements traceability

Requirement	Description	Satisfied by
SRS_BSW_00003	All software modules shall provide version and identification information	SWS_IpduM_00037
SRS_BSW_00005	Modules of the µC Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	SWS_IpduM_00999
SRS_BSW_00009	All Basic SW Modules shall be documented according to a common standard.	SWS_IpduM_00104, SWS_IpduM_00105
SRS_BSW_00101	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	SWS_IpduM_00032, SWS_IpduM_00033
SRS_BSW_00162	The AUTOSAR Basic Software shall provide a hardware abstraction layer	SWS_IpduM_00999
SRS_BSW_00164	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	SWS_IpduM_00999
SRS_BSW_00168	SW components shall be tested by a function defined in a common API in the Basis-SW	SWS_IpduM_00999
SRS_BSW_00171	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	SWS_IpduM_00999
SRS_BSW_00314	All internal driver modules shall separate the interrupt frame definition from the service routine	SWS_IpduM_00999
SRS_BSW_00323	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	SWS_IpduM_00028
SRS_BSW_00325	The runtime of interrupt service routines and functions that are running in interrupt context shall be kept short	SWS_IpduM_00999
SRS_BSW_00336	Basic SW module shall be	SWS_IpduM_00999

	able to shutdown	
SRS_BSW_00337	Classification of development errors	SWS_IpduM_00026, SWS_IpduM_00153, SWS_IpduM_00162
SRS_BSW_00339	Reporting of production relevant error status	SWS_IpduM_00999
SRS_BSW_00344	BSW Modules shall support link-time configuration	SWS_IpduM_00032
SRS_BSW_00357	For success/failure of an API call a standard return type shall be defined	SWS_IpduM_00102
SRS_BSW_00369	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	SWS_IpduM_00032, SWS_IpduM_00037, SWS_IpduM_00040, SWS_IpduM_00043, SWS_IpduM_00044, SWS_IpduM_00060
SRS_BSW_00375	Basic Software Modules shall report wake-up reasons	SWS_IpduM_00999
SRS_BSW_00377	A Basic Software Module can return a module specific types	SWS_IpduM_00999
SRS_BSW_00386	The BSW shall specify the configuration for detecting an error	SWS_IpduM_00999
SRS_BSW_00405	BSW Modules shall support multiple configuration sets	SWS_IpduM_00032
SRS_BSW_00406	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	SWS_IpduM_00084
SRS_BSW_00407	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	SWS_IpduM_00037
SRS_BSW_00414	Init functions shall have a pointer to a configuration structure as single parameter	SWS_IpduM_00162, SWS_IpduM_00174
SRS_BSW_00417	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	SWS_IpduM_00999
SRS_BSW_00422	Pre-de-bouncing of error status information is done within the DEM	SWS_IpduM_00999
SRS_BSW_00423	BSW modules with AUTOSAR interfaces shall be describable with the	SWS_IpduM_00999

	means of the SW-C Template	
SRS_BSW_00427	ISR functions shall be defined and documented in the BSW module description template	SWS_IpduM_00999
SRS_BSW_00429	Access to OS is restricted	SWS_IpduM_00107
SRS_BSW_00432	Modules should have separate main processing functions for read/receive and write/transmit data path	SWS_IpduM_00999, SWS_IpduM_91001, SWS_IpduM_91002
SRS_BSW_00433	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	SWS_IpduM_00999
SRS_BSW_00437	Memory mapping shall provide the possibility to define RAM segments which are not to be initialized during startup	SWS_IpduM_00999
SRS_BSW_00438	Configuration data shall be defined in a structure	SWS_IpduM_00159
SRS_IpduM_02801	The size in bits of the selector field shall be configurable	SWS_IpduM_00173
SRS_IpduM_02802	The position of the selector field within the PDU shall be configurable	SWS_IpduM_00173
SRS_IpduM_02803	It shall be possible not to assign a SDU layout to the unused selector field values	SWS_IpduM_00011
SRS_IpduM_02807	The I-PDU Multiplexer module shall be designed in a way that it does not produce any additional runtime	SWS_IpduM_00097
SRS_IpduM_02809	The initial values of the static part shall be derived from the COM configuration	SWS_IpduM_00067, SWS_IpduM_00068, SWS_IpduM_00098, SWS_IpduM_00143
SRS_IpduM_02810	The PduR shall be configured to send parts of multiplexed I-PDUs to the IPduM on sender side	SWS_IpduM_00089, SWS_IpduM_00090, SWS_IpduM_00091
SRS_IpduM_02811	There shall be three different triggering conditions configurable that define when the combined multiplexed I-PDUs are sent to the lower layer	SWS_IpduM_00021, SWS_IpduM_00168

SRS_IpduM_02812	The PduR shall be configured to send multiplexed I-PDUs for de-multiplexing to the IPduM after they were received from the lower layer	SWS_IpduM_00041, SWS_IpduM_00042, SWS_IpduM_00086, SWS_IpduM_00140
SRS_IpduM_02813	The PduR shall be configured to send confirmations related to multiplexed I-PDUs to IPduM after receiving them from the lower layer	SWS_IpduM_00022
SRS_IpduM_02814	The confirmation shall depend upon selector field	SWS_IpduM_00087, SWS_IpduM_00088, SWS_IpduM_00152
SRS_IpduM_02816	On sender side the IPduM shall combine the static and the appropriate dynamic part within IPduM	SWS_IpduM_00015, SWS_IpduM_00017, SWS_IpduM_00169, SWS_IpduM_00171, SWS_IpduM_00172, SWS_IpduM_00223, SWS_IpduM_00224, SWS_IpduM_00225, SWS_IpduM_00226
SRS_IpduM_02817	On receiver side the IPduM extracts the static and dynamic parts of the multiplexed I-PDU	SWS_IpduM_00040, SWS_IpduM_00224, SWS_IpduM_00227
SRS_IpduM_02818	The IPduM confirms to COM the static part of the multiplexed I-PDU and the dynamic part	SWS_IpduM_00022
SRS_IpduM_02820	Dynamic I-PDU Mapping	SWS_IpduM_00175, SWS_IpduM_00179, SWS_IpduM_00180, SWS_IpduM_00181, SWS_IpduM_00182, SWS_IpduM_00183, SWS_IpduM_00184, SWS_IpduM_00185, SWS_IpduM_00186, SWS_IpduM_00187, SWS_IpduM_00188, SWS_IpduM_00189, SWS_IpduM_00190, SWS_IpduM_00191, SWS_IpduM_00192, SWS_IpduM_00193, SWS_IpduM_00194, SWS_IpduM_00195, SWS_IpduM_00196, SWS_IpduM_00199, SWS_IpduM_00200, SWS_IpduM_00201, SWS_IpduM_00202, SWS_IpduM_00203, SWS_IpduM_00207, SWS_IpduM_00208, SWS_IpduM_00210, SWS_IpduM_00211, SWS_IpduM_00212, SWS_IpduM_00213, SWS_IpduM_00214, SWS_IpduM_00215, SWS_IpduM_00216, SWS_IpduM_00217, SWS_IpduM_00220, SWS_IpduM_00228, SWS_IpduM_00229, SWS_IpduM_00230, SWS_IpduM_00231
SRS_IpduM_02821	The temporal order of I-PDUs shall be preserved	SWS_IpduM_00209, SWS_IpduM_00219, SWS_IpduM_00221, SWS_IpduM_00222
SRS_IpduM_02822	Two different Header Sizes shall be supported	SWS_IpduM_00177
SRS_IpduM_02823	The position of I-PDUs inside a Container shall be dynamic	SWS_IpduM_00178, SWS_IpduM_00248, SWS_IpduM_00249
SRS_IpduM_02824	The ID used in the header	SWS_IpduM_00204, SWS_IpduM_00205,

	shall be independent of the Container	SWS_IpduM_00206, SWS_IpduM_00207, SWS_IpduM_00250, SWS_IpduM_00251
SRS_IpduM_02825	Static I-PDU Mapping	SWS_IpduM_00232, SWS_IpduM_00233, SWS_IpduM_00234, SWS_IpduM_00235, SWS_IpduM_00236, SWS_IpduM_00237, SWS_IpduM_00238, SWS_IpduM_00240, SWS_IpduM_00241, SWS_IpduM_00242, SWS_IpduM_00245, SWS_IpduM_00246, SWS_IpduM_00247

7 Functional specification

7.1 General

There are two different approaches of multiplexing several I-PDUs into one resulting PDU being transferred on the bus:

I-PDU Multiplexing means using the same I-PDU ID transferred from the PDU Router to the Communication Hardware Abstraction Layer with more than one unique layout of this I-PDU; see also [1].

Multiple PDU to Container Mapping means collecting several I-PDUs into one Container PDU. This Container PDU is then transferred via PduR as one (large) I-PDU. This way advantage of the larger frame sizes of newer bus systems can be taken, allowing an efficient usage of the bandwidth in combination with smaller I-PDU sizes (usually 8 bytes).

[SWS_IpduM_00097] ⌈The IpduM shall be implemented so that no other modules depend on it and that it is be possible to build a system without the IpduM module if it is not needed.⌋ (SRS_IpduM_02807)

7.2 I-PDU Multiplexing

7.2.1 Definitions and Layout

A multiplexed I-PDU consists of a static part and a dynamic part, where the static part consists of zero or more signals or signal groups. The dynamic part consists of the selector field and one or more signals or signal groups; see **Figure 2**.
The dynamic part of an I-PDU is comparable with a union of the programming language ***C***. Depending on the value of the selector field inside the I-PDU, the actual layout of the I-PDU is selected.

The position of the static and the dynamic part are configurable per I-PDU. The static and the dynamic part can be subdivided into different segments.
Only one selector field can be defined for each multiplexed I-PDU. The value of the selector field defines how the content of the dynamic part of the I-PDU will be interpreted. The selector field has a configurable size between 1 and 16 contiguous bits and its position can be defined by configuration, see ECUC_IpduM_00054.
See Chapter 10.2.1 for an overview of the IpduM configuration. Chapter 10.4 defines the configuration rules.

Multiplexing of PDUs is originally known from CAN, but it is not restricted to this communication system. The IpduM is layered next to the PDU Router above the interface layer (Communication Hardware Abstraction) in the AUTOSAR layer architecture and therefore this feature could be used for all bus systems, which can be handled by the PDU Router, for example FlexRay.

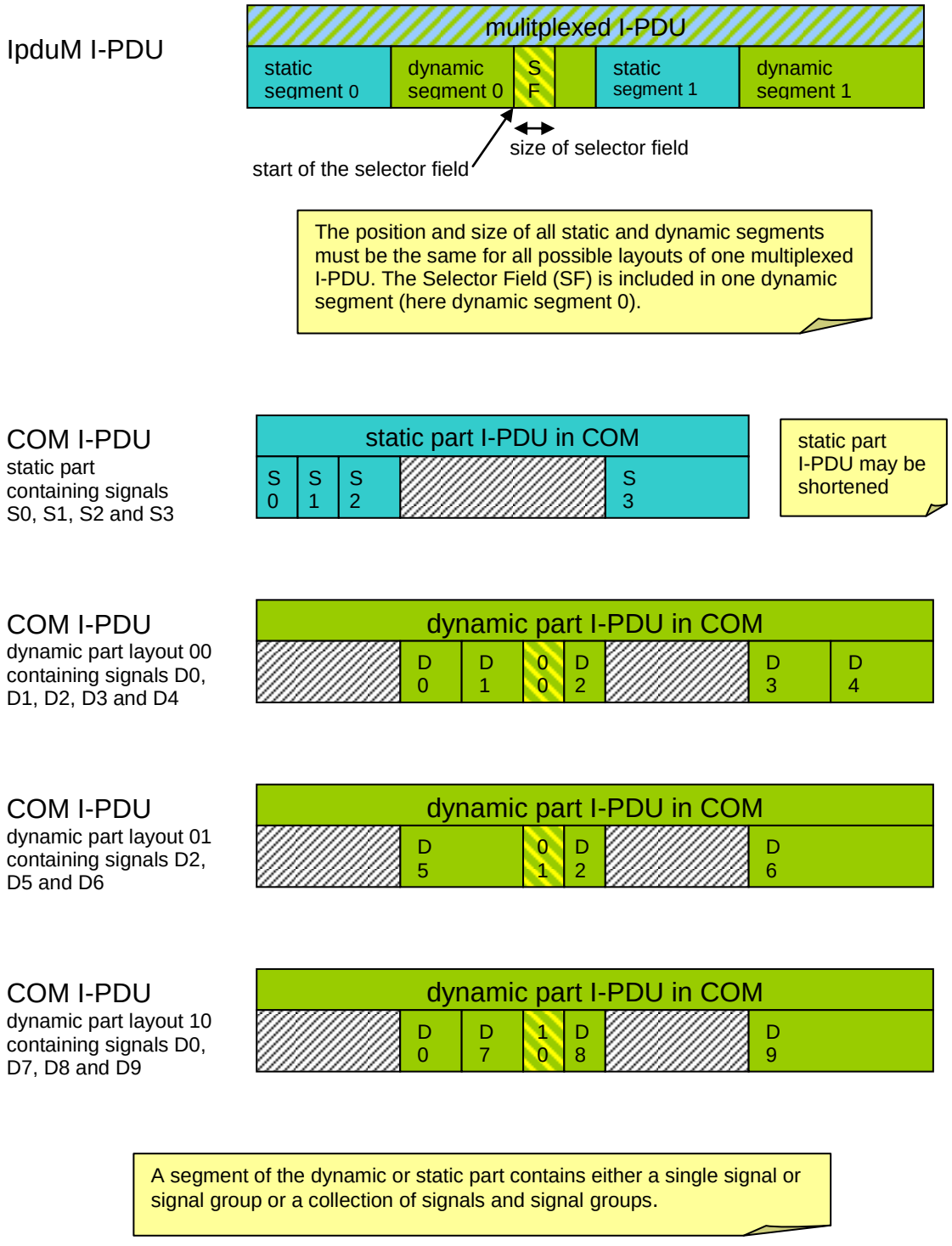

Figure 2 Possible layout of a multiplexed I-PDU with shortened static part

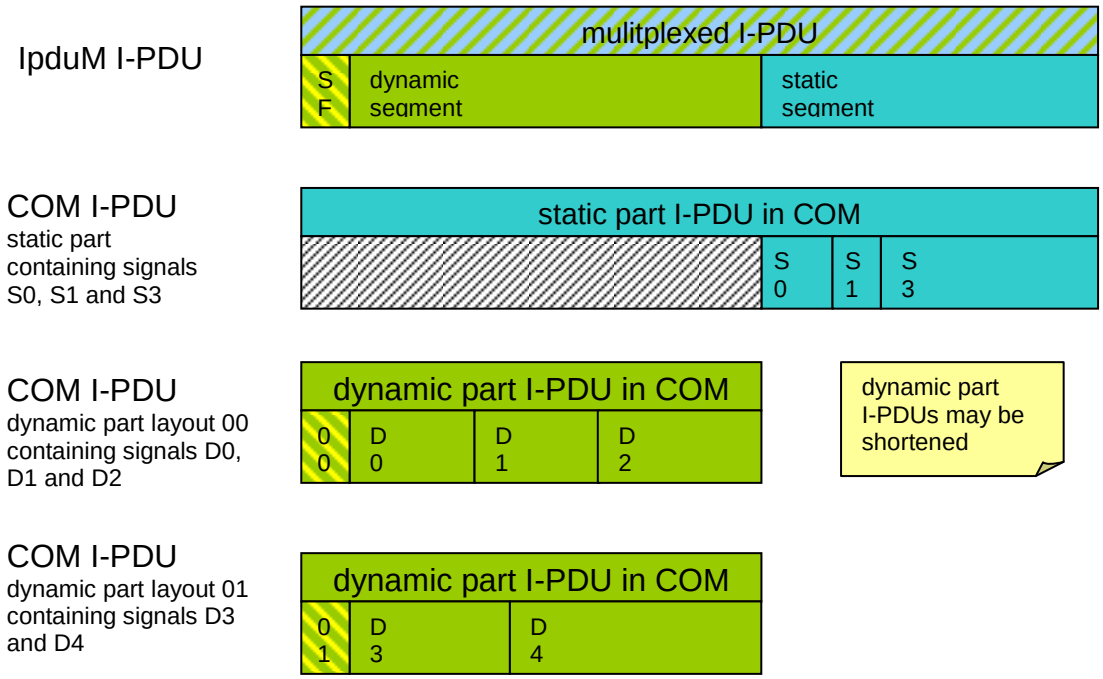

Figure 3 Possible layout of a multiplexed I-PDU with shortened dynamic parts

7.2.2 General

There is one COM I-PDU for the static part and one COM I-PDU for each layout of the dynamic part of one multiplexed IpduM I-PDU, so the IpduM combines at most two I-PDUs of COM.

[SWS_IpduM_00098] [The IpduM module shall not set the selector field.] (SRS_IpduM_02809)

The IpduM module relies on the configuration of the COM module. For each dynamic layout, an I-PDU needs to be configured in COM. Such I-PDUs already have to contain the correct selector field value. The selector field values in COM can be initialized by configuring them as signals that are initialized with an init value but are never written after initialization.

[SWS_IpduM_00173] [The IpduM shall respect the IpduMByteOrder when interpreting the selector field value.] (SRS_IpduM_02801, SRS_IpduM_02802)

For a detailed description of the transmission and reception of a multiplexed I-PDU see Chapter 7.2.4 and 7.2.5.

[SWS_IpduM_00140] [It shall be allowed to optimize the Rx- and Tx-Confirmation path from the IpduM module via the PDU Router module to the COM layer to call the COM API directly from the IpduM module without including the PDU Router. This

shall be indicated by setting the published parameter IpduMRxDirectComInvocation to TRUE, see ECUC_IpduM_00142.] (SRS_IpduM_02812)

In case of the COM invocation, optimization as defined above IpduM.c needs to include Com.h.

7.2.3 Initialization

The IpduM module provides an initialization function IpduM_Init defined in SWS_IpduM_00032. This function initializes all internal global variables and the buffers of the IpduM I-PDUs. For more details, see Chapter 8.3.1.

The environment of the IpduM shall call IpduM_Init before calling any other function of the IpduM module.

The implementer has to ensure that IPDUM_E_UNINIT is returned in development mode in case an API function (except IpduM_MainFunctionTx, IpduM_MainFunctionRx and IpduM_GetVersionInfo) is called before the module is initialized.

For the I-PDU data transmission pathway through the IpduM module, a buffer is allocated inside the IpduM module. This buffer needs to be initialized because it might be transmitted before it has been fully populated with data by the COM module. The initialization data of this buffer is derived from the initial values of the COM module's configuration as follows:

1) **[SWS_IpduM_00067]** ⌈The IpduM shall initialize its internal transmit buffers with the configured pattern IpduMIPduUnusedAreasDefault.⌋ (SRS_IpduM_02809)

2) **[SWS_IpduM_00068]** ⌈The initial signal values of the initial dynamic part shall be set according to initial values of the referenced COM I-PDU (IpduMInitialDynamicPart -> IpduMTxDynamicPart -> IpduMTxDynamicPduRef).⌋ (SRS_IpduM_02809)

3) **[SWS_IpduM_00143]** ⌈The initial signal values of the static part shall be set according to the intial values of the referenced COM I-PDU (IpduMTxStaticPart -> IpduMTxStaticPduRef)⌋ (SRS_IpduM_02809)

The selector field is contained within one segment of the intial dynamic part and therefore is initialized implicitly.

For optimization, the initial bit pattern for the buffer can be worked out at configuration-time and then copied at run-time.

7.2.4 Transmission

Inside COM, there are separated I-PDUs for the static part and one for each dynamic part of a multiplexed I-PDU.

The static part and the dynamic parts are treated in COM as separate I-PDUs with their own I-PDU IDs.

[SWS_IpduM_00015] ⌈For a multiplexed I-PDU IpduM shall merge the corresponding two COM I-PDUs representing the associated static part and the last received dynamic part into one single IpduM I-PDU with a new unique I-PDU ID. IpduM shall send out this new IpduM I-PDU to the PDU Router module, see also **Figure 1**.⌋ (SRS_IpduM_02816)

For details about the trigger of the transmission, see Chapter 7.2.4.2.

All control functionalities like deadline monitoring of the COM I-PDUs and update-bit evaluation are out of the scope of the IpduM and have to be done by the COM layer. For details about the timing-behavior of the new combined I-PDU see Chapter 7.2.4.2.

7.2.4.1 Transmission request

The IpduM module provides an IpduM_Transmit function so that the PDU Router is able to initiate the transmission of an I-PDU; see SWS_IpduM_00043.

[SWS_IpduM_00017] ⌈The function IpduM_Transmit shall assemble the multiplexed I-PDU, using the related static and dynamic part, and transmit it according to the trigger conditions/ modes as defined in SWS_IpduM_00021 and ECUC_IpduM_00125.⌋ (SRS_IpduM_02816)

As defined in Chapter 7.2.3, each outgoing I-PDU has an initial value so that, should an I-PDU be transmitted by the IpduM module before both static and dynamic parts have been sent from COM to the IpduM, a value defined by the configuration is transmitted.

[SWS_IpduM_00152] ⌈As long as no transmission confirmation for the IpduM I-PDU is received (regardless of the result), the function IpduM_Transmit shall return E_NOT_OK for any new transmission request from the upper layer with a COM I-PDU belonging to the same IpduM I-PDU.⌋ (SRS_IpduM_02814)

The IpduM module relies here on a configured transmission configuration in the lower layer.

In case a multiplexed I-PDU is only triggered for sending by either updating the dynamic or static part, the non-triggering part might be overwritten if updated multiple times between two transmissions.

7.2.4.2 Transmission trigger

The IpduM module receives the static and the dynamic part of a multiplexed I-PDU by separated two transmission requests as two single COM I-PDUs from the PDU Router module.

[SWS_IpduM_00021] [The IpduM module shall be configurable to send a transmission request for the new multiplexed I-PDU to the PDU Router because of the following trigger conditions/ modes:

- receiving a static part
- receiving a dynamic part
- receiving a static or a dynamic part
- does not trigger transmission because of receiving anything of this I-PDU (IpduMTxTriggerMode None) in case of TriggerTransmit

For configuration, see ECUC_IpduM_00052.] (SRS_IpduM_02811)

The four trigger conditions/ modes defined by SWS_IpduM_00021 allow controlling the transmission mode of the new assembled I-PDU by the transmission modes of the single I-PDUs sent by COM, see also [5].

Not all of four trigger conditions/ modes defined by SWS_IpduM_00021 guarantee the minimum delay time between consecutive transmissions of different instances of multiplexed I-PDUs, because if the transmission is triggered by static and dynamic part or only by the dynamic part, COM does not take care for the minimum delay time. COM treats the static part and the different dynamic parts as unrelated standalone I-PDUs.

The configuration “does not trigger transmission because of receiving anything” is needed if an I-PDU is only sent out because of a TriggerTransmit of a lower layer. With the API IpduM_TriggerTransmit it is possible for lower layers to trigger a send out of an I-PDU.

In case the IpduMTxTriggerMode is None and the lower layer triggers the transmission via IpduM_TriggerTransmit, the IpduMTxConfirmationPduId needs to be configured since this ID is also used for resolving the I-PDU in case of IpduM_TriggerTransmit, see also ECUC_IpduM_00158.

7.2.4.3 Just-In-Time update of parts

Sometimes it may be unwanted that the IpduM module not just sends out the locally stored parts, since these parts may contain outdated information e.g. update-bits. Therefore, the IpduM supports a per part configurable just-in-time update mechanism.

[SWS_IpduM_00168] [In case the transmission of a multiplexed I-PDU is triggered by the update of one part and IpduMJitUpdate is configured to true for the second part, the IpduM module shall update the second part via PduR_IpduMTriggerTransmit before the multiplexed I-PDU is sent out via PduR_IpduMTransmit.] (SRS_IpduM_02811)

[SWS_IpduM_00169] [In case the contents of a multiplexed I-PDU is requested via IpduM_TriggerTransmit, the IpduM module shall update all parts which have IpduMJitUpdate configured to *true* before returning the contents of the multiplexed I-PDU.] (SRS_IpduM_02816)

[SWS_IpduM_00223] ⌈In case the IpduM shall update the dynamic part just-in-time, the latest dynamic part sent by the upper layer shall be updated or the dynamic part referenced by IpduMInitialDynamicPart if no dynamic part was sent before.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00171] ⌈In case the transmission of a multiplexed I-PDU is triggered by the update of one part and IpduMJitUpdate is configured to *true* for the second part, the multiplexed I-PDU shall not be send if the JIT-update request via PduR_IpduMTriggerTransmit returns E_NOT_OK.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00172] ⌈In case the contents of a multiplexed I-PDU is requested via IpduM_TriggerTransmit and IpduMJitUpdate is configured to *true* for any multiplexed part, IpduM_TriggerTransmit shall return E_NOT_OK if any of the JIT-update requests via PduR_IpduMTriggerTransmit return E_NOT_OK.⌋ (SRS_IpduM_02816)

7.2.4.4 Transmission confirmation

Transmission confirmations are given to the IpduM module by the PDU Router according to the configuration of the I-PDUs in the PDU Router.

[SWS_IpduM_00022] ⌈If the IpduM receives a TxConfirmation for a specific IpduM I-PDU, it shall translate this confirmation into the corresponding confirmations for the COM I-PDUs, which were contained in the last sent out multiplexed IpduM I-PDU.⌋ (SRS_IpduM_02813, SRS_IpduM_02818)

Depending on the configuration of IpduMTxDynamicConfirmation (ECUC_IpduM_00163) and IpduMTxStaticConfirmation (ECUC_IpduM_00164), the IpduM will pass zero, one or two confirmations towards COM for one send request. The number of confirmations given to the upper layer does not depend on the IpduMTxTriggerMode.

Examples:

a) If neither IpduMTxDynamicConfirmation nor IpduMTxStaticConfirmation for the corresponding IpduMTxRequest is configured to true, no COM confirmation is generated.
b) If IpduMTxStaticConfirmation is configured to true but and IpduMTxDynamicConfirmation is configured to false (or vice versa), then only one COM confirmation is generated.
c) If both IpduMTxStaticConfirmation and IpduMTxDynamicConfirmation is configured to true, then two COM confirmations are generated; to the I-PDU representing the static part and the I-PDU representing the dynamic part.

In case two transmission confirmatios are generated, they are obviously equal, since they are derived from the same I-PDUM transmission confirmation.

7.2.5 Reception

Every I-PDU which is received by the Communication Hardware Abstraction (CAN Interface, Lin Interface, FlexRay Interface) is given to the PDU Router. The PDU Router routes multiplexed I-PDUs to the IpduM module. The IpduM module separately routes the static and dynamic parts of the multiplexed I-PDU to their destinations.

It is known at configuration-time which incoming I-PDU IDs correspond to multiplexed I-PDUs with a static part configured. The I-PDU ID is all that is necessary to work out if there is a static part present.

As all multiplexed I-PDUs contain a dynamic part this part always has to be routed.

There are no requirements to handle or notify wrongly configured parts. Hence, if the received I-PDU contains segments not configured for reception on this ECU, they will be ignored silently. Furthermore, if an I-PDU is configured with a PduLength of 0, it will also be ignored silently, since no meaningful processing can be configured.

This situation might occur in a gateway setting, if a multiplexed I-PDU is always routed onto another bus by the PDU Router, but contains a signal in one dynamic part that must be passed to the application. In this case, the multiplexed PDU would have to be routed to the IpduM as well.

7.2.6 Metadata handling

The requirements of this section only apply if IpduMMetaDataSupport is configured to *true*.

[SWS_IpduM_00225] ⌈If IpduMTxTriggerMode is configured to a different value than *NONE*, the IpduM shall use the MetaData of the triggering part for sending of the multiplexed I-PDU.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00226] ⌈If IpduMTxTriggerMode is configured to *NONE*, the IpduM shall use the MetaData of the last updated part for sending of the multiplexed I-PDU.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00227] ⌈On receiver side the IpduM shall forward the received MetaData along with all demultiplexed parts.⌋ (SRS_IpduM_02817)

7.3 Multiple-PDU-to-Container handling

IpduM supports a mapping of several I-PDUs to one Container PDU. Both contained and Container PDUs are *regular* PDUs from PduR's point of view. The Container

layout can either be dynamically defined using headers in front of the contained I-PDUs or statically without headers but defined static positions for contained I-PDUs.

IpduM relies on PduR beeing configured to forward send-PDUs mapped to a Container-PDU and received Container-PDUs to IpduM.

7.3.1 Dynamic Container Layout

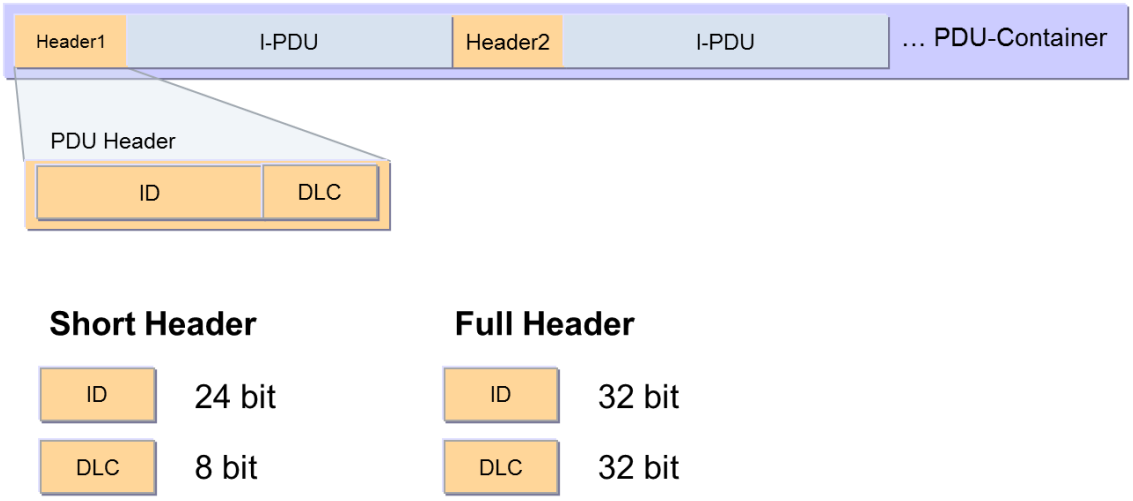

Figure 4 Layout of a dynamic Container

DRAFT: [SWS_IpduM_00175] [Inside a dynamic Container PDU IpduM shall place the header of a contained I-PDU in front of the contained I-PDU.] (SRS_IpduM_02820)

See also **Figure 4**: Layout of a dynamic Container PDU.

For dynamic Container PDUs, there is no configuration of the positions of contained I-PDUs inside the Container PDU, thus the position of an arbitrary contained I-PDU is determined by the length of payload (DLC) and headers of the preceding (added before) contained I-PDUs.

The number of I-PDUs contained in a Container PDU is limited by the maximum size of the Container PDU (PDULength of ECUC-PDU).

The order of the I-PDUs inside the Container PDU will be retained. This way all contained I-PDUs are extracted in the same order as they have been put into the Container PDU. See SWS_IpduM_00179 and SWS_IpduM_00209.

The IpduM supports two different header sizes for dynamic Container PDUs (see ECUC_IpduM_00183: IpduMContainerHeaderSize):

- IPDUM_HEADERTYPE_SHORT with 24 bit ID and 8 bit length
- IPDUM_HEADERTYPE_LONG with 32 bit ID and 32 bit length

The header sizes are configured per Container PDU via IpduMContainerHeaderSize. Thus, it is valid for the whole Container PDU. Mixing of header sizes inside one Container PDU is not supported.

[SWS_IpduM_00177] [Each I-PDU header shall consist of ID field and length field in the byte order determined by IpduMHeaderByteOrder.] (SRS_IpduM_02822)

DRAFT: [SWS_IpduM_00178] [Placing of headers and payloads of contained I-PDUs inside a dynamic Container PDU shall be contiguous without any gap.] (SRS_IpduM_02823)

Rationale: This allows iterating over a Container PDU by considering the header size and payload lengths (DLC from header).

This has to be ensured by the implementation of the container collection algorithm, since contained I-PDUs have no dedicated (configured) position inside a Container PDU.

7.3.2 Static Container Layout

To enable the static container layout, the IpduMContainerHeaderSize of the Container PDU has to be configured to *IPDUM_HEADERTYPE_NONE*.

DRAFT: [SWS_IpduM_00232] [If the IpduMContainerHeaderSize is set to *IPDUM_HEADERTYPE_NONE*, the IpduM module shall statically place the contained I-PDUs within the Container PDU according to their configured IpduMContainedTxPduOffset] (SRS_IpduM_02825)

For the Static Container Layout only contained I-PDUs with IpduMContainedTxPduCollectionSemantics set to *IPDUM_COLLECT_LAST_IS_BEST* is supported (see ECUC_IpduM_00198).

7.3.3 Transmission

[SWS_IpduM_00181] [When adding a contained I-PDU to a Container PDU which has not been triggered yet, and if IpduMContainedTxPduTrigger is set to *IPDUM_TRIGGER_ALWAYS*, the Container PDU shall be triggered immediately.] (SRS_IpduM_02820)

[SWS_IpduM_00201] [When adding the first contained I-PDU to a Container PDU with the parameter IpduMContainerTxFirstContainedPduTrigger set to TRUE, IpduM shall call PduR_IpduMTransmit.] (SRS_IpduM_02820)

Rationale: This way a transmission is requested for a time-triggered bus.

[SWS_IpduM_00184] [When adding the first contained I-PDU to a Container PDU and either IpduMContainerTxSendTimeout of the Container PDU or IpduMContainedTxPduSendTimeout of the contained I-PDU is configured greater than zero, the IpduM module shall start the transmission timer of the Container PDU. The timer

shall be initialized with the smaller non zero value of IpduMContainerTxSendTimeout and IpduMContainedTxPduSendTimeout.⌋ (SRS_IpduM_02820)

Note: If neither the IpduMContainedTxPduSendTimeout nor the IpduMContainerTx-SendTimeout is provided to the Container PDU and its Contained PDUs, the IpduM module will not trigger the Container PDU by the transmission timer of the Container PDU (no timeout-based triggering for the Container PDU).

Until the Container PDU is fetched (see SWS_IpduM_00194) or unless maximum size of the Container PDU is not exceeded further requested I-PDUs assigned to this container can be added.

[SWS_IpduM_00185] ⌈When a contained I-PDU is added to a Container PDU, the transmission timer of the Container PDU shall be updated with the contained I-PDU's timeout (IpduMContainedTxPduSendTimeout) if it is less than the remaining time of the Container PDU.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00186] ⌈When the transmission timer of the Container PDU defined by SWS_IpduM_00184 has elapsed, the Container PDU shall be triggered.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00188] ⌈When a Container PDU is triggered, IpduM shall invoke PduR_IpduMTransmit.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00192] ⌈When passing a Container PDU to PduR the Parameter Pdu-InfoPtr shall contain a pointer to the assembled Container PDU in SduDataPtr and the total length (according to SWS_IpduM_00187) in SduLength.⌋ (SRS_IpduM_02820)

7.3.3.1 Queueing

In case more than one instance of a Container PDU has to be kept by IpduM, up to IpduMContainerQueueSize instances can be stored in addition to the current in-stance. The current instance is one instance of the Container PDU that currently contained I-PDUs are being added to. After this instance has either been queued or copied to the lower layer, i.e. after a TriggerTransmit or Transmit API call depending on the configuration of IpduMContainerTxTriggerMode, no more contained I-PDUs can be added to this instance.

[SWS_IpduM_00195] ⌈If PduR_IpduMTransmit has returned *E_NOT_OK*, the same transmit request shall be repeated during the next call to IpduM_MainFunctionTx. The instance of that Container PDU is queued in the meantime.⌋ (SRS_IpduM_02820)

See also SWS_IpduM_00199.

[SWS_IpduM_00189] ⌈The IpduM shall wait for the transmission confirmation (re-gardless of the result) before invoking PduR_IpduMTransmit for the next instance of that Container PDU.⌋ (SRS_IpduM_02820)

The IpduM module relies here on a configured transmission confirmation for that Container PDU in the lower layer.

[SWS_IpduM_00190] ⌈In case the transmission confirmation for that Container PDU was received, the IpduM shall invoke PduR_IpduMTransmit for the next oldest instance of that Container PDU during the next call to IpduM_MainFunctionTx at the latest.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00191] ⌈In case IpduMContainerTxTriggerMode is set to *IPDUM_DIRECT*, and PduR_IpduMTransmit returns *E_OK* for that Container PDU, IpduM shall remove that instance from the queue.⌋ (SRS_IpduM_02820)

In this case instances of a Container-PDU can be lost, if a queue inside CanIf is used since a newer instance could overwrite a previous one. Such last-is-best behavior might not be desired in this case.

[SWS_IpduM_00196] ⌈If the IpduM receives a TxConfirmation for a specific Container PDU, it shall translate this confirmation into the corresponding confirmations for those contained I-PDUs having IpduMContainedTxPduConfirmation set to *TRUE* and were contained in the last sent out instance of the Container I-PDU.
If the same contained I-PDU is present more than once, this results in multiple TxConfirmations⌋ (SRS_IpduM_02820)

[SWS_IpduM_00199] ⌈If creating a new instance of a Container PDU would exceed IpduMContainerQueueSize the oldest instance shall be discarded. If IpduMContainerQueueSize is not configured the local instance shall be discarded. In both cases IPDUM_E_QUEUEOVFL shall be reported to DET via Det_ReportRuntimeError.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00200] ⌈A Container PDU instance shall be dropped from the queue if it is fetched by TriggerTransmit.⌋ (SRS_IpduM_02820)

7.3.3.2 Triggered Transmission and Last-is-Best semantics

[SWS_IpduM_00193] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_TRIGGERTRANSMIT*, IpduM shall keep and provide buffered data until it is fetched by a call to IpduM_TriggerTransmit.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00194] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_TRIGGERTRANSMIT*, IpduM_TriggerTransmit shall copy the oldest Conainer PDU instance in the queue. If the queue is empty/ non-existent, the current instance of the Container PDU is copied. If the current instance of the Container PDU is empty/ non-existent as well, *E_NOT_OK* is returned by IpduM_TriggerTransmit.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00220] ⌈For contained I-PDUs, with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST, IpduM shall use PduR_IpduMTriggerTransmit to fetch the PDU data from its upper layer immediately before it transfers the container I-PDU to the lower layer.⌋ (SRS_IpduM_02820)

While it seems natural to use IpduMContainedTxPduCollectionSemantics IPDUM_COLLECT_LAST_IS_BEST in combination with IpduMContainerTxTrigger-Mode IPDUM_TRIGGERTRANSMIT, it may also be used in combination with IPDUM_DIRECT.

As soon as a contained I-PDU is configured to use last-is-best semantics, the user accepts that not necessarily all instances/ values of this contained I-PDU are visible on the wire. On the other hand, queued collection semantics guarantees that every instance/ value of the contained I-PDU is visible on the wire.

7.3.4 Transmission of Dynamic Containers

The requirements within this chapter and its subchapters complement the transmission for Dynamic Containers and do not apply to Static Containers.

Due to the following requirements IpduM will make sure that instances of a contained I-PDU (same PDU-ID) are transmitted (passed to PduR inside their Container PDUs) in exactly the same order as they are passed to IpduM.

[SWS_IpduM_00179] ⌈When a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to *IPDUM_COLLECT_QUEUED* (see ECUC_IpduM_00198) is passed to IpduM via IpduM_Transmit, IpduM shall identify the associated Container PDU and append the contained I-PDU to its payload even if a previous instance of the contained I-PDU is already present in that Container PDU.⌋ (SRS_IpduM_02820)

This way a Container PDU can include more than one instance of the same I-PDU. The resulting behavior is FIFO-like in order to preserve the order of I-PDU instances being transmitted. Thus, the upper layer(s) of the receiving IpduM can implement either last-is-best or FIFO semantics.

[SWS_IpduM_00180] ⌈If a contained I-PDU has been added to a Container PDU that has not been triggered yet, and if the resulting payload is bigger than IpduMContainerTxSizeThreshold the Container PDU shall be triggered.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00182] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_DIRECT* and adding a contained I-PDU would exceed maximum size of the Container I-PDU, first the Container PDU shall be triggered. The contained I-PDU shall be added to a new instance of the Container PDU.⌋ (SRS_IpduM_02820)

SWS_IpduM_00189 has to be considered also in case both SWS_IpduM_00181 and SWS_IpduM_00182 apply.

[SWS_IpduM_00183] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_-TRIGGERTRANSMIT* and adding a contained I-PDU would exceed maximum size of the Container PDU, first the Container PDU shall be queued. Then the contained I-PDU shall be added to a new instance of the Container PDU.⌋ (SRS_IpduM_02820)

Contained I-PDUs will be added to Container PDUs with IpduMContainerTxTrigger-Mode = IPDUM_TRIGGERTRANSMIT as long as they are neither full nor queued.

[SWS_IpduM_00187] ⌈After a Container PDU is triggered or being fetched by TriggerTransmit, IpduM shall calculate the overall size of the Container PDU. The total size builds up by the total of all payloads of the contained I-PDUs plus the total length of the corresponding headers. The result shall be the payload size of the Container PDU.⌋ (SRS_IpduM_02820)

7.3.4.1 Triggered Transmission and Last-is-Best semantics

In case of contained I-PDUs with IpduMContainedTxPduCollectionSemantics set to *IPDUM_COLLECT_LAST_IS_BEST*, the IpduM module updates these I-PDUs before sending. If such contained I-PDUs have dynamic size, it can happen that the container size is not sufficient for all contained I-PDUs, if the overall size of the updated I-PDUs increases.

[SWS_IpduM_00231] ⌈If in case of updating contained I-PDUs with IpduMContainedTxPduCollectionSemantics IPDUM_COLLECT_LAST_IS_BEST, IpduMContainedTxPduPriorityHandling is set to FALSE and the container size is not sufficient for a contained I-PDU, this contained I-PDU and all following shall be shifted to the beginning of the next container instance.⌋ (SRS_IpduM_02820)

In order to preserve the order of the contained I-PDUs, also all following contained I-PDUs needs to be shifted even if there would be enough space in the current container.

[SWS_IpduM_00221] ⌈When storing contained I-PDUs into Container PDUs and IpduMContainedTxPduPriorityHandling is set to FALSE, the IpduM shall retain the order in which the contained I-PDUs are passed to IpduM. That is the first passed contained I-PDU is placed at the beginning at the container and so on. If a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST is passed multiple times, the IpduM shall store it only once at the position matching its first occurrence. ⌋ (SRS_IpduM_02821)
[SWS_IpduM_00249] ⌈ When storing contained I-PDUs with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST into Container PDUs and IpduMContainedTxPduPriorityHandling is set to TRUE, the IpduM shall collect the contained I-PDUs in respect to the IpduMContainedTxPduPriority (highest priority first and so on). I-PDUs with the same IpduMContainedTxPduPriority shall be collected in the order the send request occurred.⌋ (SRS_IpduM_02823)

Note: Multiple triggers of the same ContainedIPdu during collection of one ContainerIPdu have no effect, in case IpduMContainedTxPduCollectionSemantics is set to IPDUM_COLLECT_LAST_IS_BEST.

The requirements [SWS_IpduM_00221] and [SWS_IpduM_00249] above only defines the order of the contained I-PDUs but not the point in time when the container is constructed. For container with contained I-PDUs having last-is-best semantic, it

might be more efficient to just keep track of the order and construct the Container PDU on demand.

[SWS_IpduM_00222] ⌈In case PduR_IpduMTriggerTransmit returns *E_NOT_OK* for a contained I-PDU, IpduM shall omit this contained I-PDU silently. The associated Container PDU shall be transmitted anyway without the omitted contained I-PDU. All contained I-PDUs behind the skipped one shall be moved up by the size of the omitted contained I-PDU including its header.⌋ (SRS_IpduM_02821)

7.3.5 Transmission of Static Containers

The requirements within this chapter and its subchatpers complement the transmission for Static Containers and do not apply to Dynamic Containers.

DRAFT: [SWS_IpduM_00234] ⌈For Container PDUs with static container layout and IpduMContainerTxTriggerMode is set to IPDUM_DIRECT, the IpduM shall trigger the Container PDU when all contained I-PDUs were updated by the upper layer.⌋ (SRS_IpduM_02825)

Since, Static Container might include not updated contained I-PDUs, there are means to detect the currentness of contained I-PDUs on receiver side. Either update-bits for contained I-PDUs or an unsed area default pattern might be configured. For the concrete configuration and configuration rules, see Chapter 10.

DRAFT: [SWS_IpduM_00235] ⌈In case a contained I-PDU has a configured IpduMUpdateBitPosition, the IpduM shall ensure that the update bit of this contained I-PDU is set if and only if the contained I-PDU was successfully updated.⌋ (SRS_IpduM_02825)

DRAFT: [SWS_IpduM_00233] ⌈In case a Static Container has a configured IpduMUnusedAreasDefault, the IpduM shall ensure that all not updated areas of the Container are set to the value of IpduMUnusedAreasDefault before the Container PDU is sent.⌋ (SRS_IpduM_02825)

This allows the IpduM to handle contained I-PDUs with dynamic length also within static containers. However, the receiving IpduM is not able to detect if the SWC or the sending IpduM set the IpduMUnusedAreasDefault-value. Hence, always the complete, thus eventually filled up contained I-PDU, will be received.

It must be observed, that some bus-systems (eg. CAN-FD and FlexRay) cannot transfer PDUs of arbitrary length and might fill up the sent I-PDU to the next possible length with their own default value. Hence, the configuration of the IpduMUnusedAreasDefault value and the bus-specifc padding-pattern should be aligned.

7.3.6 Reception

There exist two possible ways how the reception of a received Container PDU can be processed by the IpduM:

- accept configured
- accept all (only for IPDUM_HEADERTYPE_LONG and IPDUM_HEADERTYPE_SHORT)

[SWS_IpduM_00202] [If IpduMContainerPduProcessing is set to *IPDUM_PROCESSING_IMMEDIATE*, the processing of the received Container PDUs shall be executed in the context of IpduM_RxIndication. Otherwise, it is deferred to the next call to IpduM_MainFunctionRx. All deferred Container PDUs shall be processed in the order of their reception.] (SRS_IpduM_02820)

[SWS_IpduM_00203] [If by a call of IpduM_RxIndication a Container PDU is received, the contained I-PDUs shall be extracted.] (SRS_IpduM_02820)

[SWS_IpduM_00205] [If for the received Container PDU the configuration parameter IpduMContainerRxAcceptContainedPdu is set to IPDUM_ACCEPT_CONFIGURED, IpduM shall expect and match only contained I-PDUs (IpduMContainedRxPdu) that reference the IpduMContainerRxPdu in IpduMContainedRxInContainerPduRef.] (SRS_IpduM_02824)

Note that it is well possible to define several IpduMContainedRxPdus with the same IpduMContainedRxPdu(Short/Long)HeaderId as long as the IpduMContainedRxPdus are assigned to different IpduMContainerRxPdus (via IpduMContainedRxInContainerPduRef) with each IpduMContainerRxPdu having IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_CONFIGURED.

[SWS_IpduM_00250][For an IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_CONFIGURED and IpduMContainerHeaderSize=IPDUM_HEADERTYPE_LONG or IPDUM_HEADERTYPE_SHORT the following constraint applies:

- IpduMContainedRxPdus with an IpduMContainedRxInContainerPduRef to that specific IpduMContainerRxPdu shall have unique IpduMContainedRxPduLongHeaderId or IpduMContainedRxPduShortHeaderId, respectively, defined in the context of that IpduMContainerRxPdu.

] (SRS_IpduM_02824)

[SWS_IpduM_00209] [Each contained I-PDU shall be notified to PduR via PduR_IpduMRxIndication. IpduM shall indicate the contained I-PDUs in the same order as the I-PDUs are located inside the Container PDU.] (SRS_IpduM_02821)

7.3.6.1 Queueing

[SWS_IpduM_00211] [If a Container PDU is received and IpduMContainerPduProcessing is set to *IPDUM_PROCESSING_DEFERRED*, the Container PDU shall be queued.] (SRS_IpduM_02820)

[SWS_IpduM_00212] [If receiving a new instance of a Container PDU would exceed IpduMContainerQueueSize the oldest instance shall be discarded and

IPDUM_E_QUEUEOVFL shall be reported to DET via Det_ReportRuntimeError.⌋ (SRS_IpduM_02820)

7.3.7 Reception of Dynamic Containers

The requirements within this chapter and its subchatpers complement the reception of Dynamic Containers and do not apply to Static Containers.

[SWS_IpduM_00204] ⌈ For each contained I-PDU of a received Container PDU where the IpduMContainerRxPdu has IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL the ID from the PDU header shall be used to identify the corresponding contained I-PDU:

- If the received Container PDU uses long or short header (IpduMContainerHeaderSize = IPDUM_HEADERTYPE_LONG or IPDUM_HEADERTYPE_SHORT, respectively) the ID shall be compared with the IpduMContainedRxPduLongHeaderId or IpduMContainedRxPduShortHeaderId, respectively, in the set of IpduMContainedRxPdus which do not have an IpduMContainedRxInContainerPduRef defined.

⌋ (SRS_IpduM_02824)

[SWS_IpduM_00206] ⌈ For a received Container PDU with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL, IpduM shall expect and match only IpduMContainedRxPdus which do not have an IpduMContainedRxInContainerPduRef defined.⌋ (SRS_IpduM_02824)

[SWS_IpduM_00251]⌈ All IpduMContainedRxPdus with no IpduMContainedRxInContainerPduRef and a defined IpduMContainedRxPduLongHeaderId or IpduMContainedRxPduShortHeaderId, shall have a unique IpduMContainedRxPduLongHeaderId or IpduMContainedRxPduShortHeaderId, respectively.⌋ (SRS_IpduM_02824)

Note that due to [SWS_IpduM_00206] it is NOT allowed to define several IpduMContainedRxPdus with the same IpduMContainedRxPduShortHeaderId / IpduMContainedRxPduLongHeaderId and no assignment to IpduMContainerRxPdu (no IpduMContainedRxInContainerPduRef defined).

Note that it is well possible to have two IpduMContainedRxPdus with no assignment to IpduMContainerRxPdu (no IpduMContainedRxInContainerPduRef defined) which have the same header id value defined, as long as one IpduMContainedRxPdu has an IpduMContainedRxPduShortHeaderId defined and the other IpduMContainedRxPdu has an IpduMContainedRxPduLongHeaderId defined.

Note that due to two ways how IpduMContainedRxPdus can interact with IpduMContainerRxPdu (IPDUM_ACCEPT_CONFIGURED and IPDUM_ACCEPT_ALL definition at the IpduMContainerRxPdu) it is well possible that IpduMContainedRxPdus with the same IpduMContainedRxPduShortHeaderId / IpduMContainedRxPduLongHeaderId exist as long as the constraints defined in [SWS_IpduM_00250], [SWS_IpduM_00251] are fulfilled.

[SWS_IpduM_00207] ⌈ If a contained I-PDU of a received IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL can not be matched according to [SWS_IpduM_00206] then this contained I-PDU shall be discarded silently.⌋ (SRS_IpduM_02820, SRS_IpduM_02824)

[SWS_IpduM_00208] ⌈For each contained I-PDU the length given in its header shall be used as the length of the corresponding I-PDU.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00210] ⌈When processing a received Container PDU and detecting a header containing the ID 0 the processing for this Container PDU shall be stopped and the remaining bytes shall be ignored.⌋ (SRS_IpduM_02820)

Rationale: A header ID of 0 means that Container PDU has been filled with padding bytes and no further data is contained.

SWS_IpduM_00210 does not mean that always a 0 has to be expected at the end of the payload.

7.3.8 Reception of Static Containers

The requirements within this chapter and its subchatpers complement the reception of Static Containers and do not apply to Dynamic Containers.

In order for the receiving IpduM module to be able to determine which of the PDUs in a received Static Container have actually been updated on the transmitter side, additional update information, so called PDU update bits within the Container Pdu, can be configured per contained I-PDU (see ECUC_IpduM_00207).

DRAFT: [SWS_IpduM_00236] ⌈In case a received contained I-PDU has a configured update bit, the IpduM module shall only process and indicate it to the upper layer if its received update-bit is set.⌋ (SRS_IpduM_02825)

The above requirement results in silently ignoring contained I-PDUs with configured but not set update bits.

With respect to SWS_IpduM_00236 expected contained I-PDUs without a configured update-bit are always processed and indicated to the upper layer. They are assumed allways valid.

7.3.9 Errorhandling

There are bus systems where it is not possible to set an arbitrary size for the transmitted L-PDU (e.g. CanFD). The valid payload length of a Container PDU can be derived from the contained headers. Therefore, the difference to the actual length of the Container PDU can be considered padding.

Assumption is that underlying bus modules are configured such that the padded values do not build up a valid header.

[SWS_IpduM_00213] ⌈When processing a received Container PDU and detecting a header where the payload length exceeds the remaining bytes of the container the processing for this Container PDU shall be stopped and the remaining bytes shall be ignored. Furthermore, *IPDUM_E_HEADER* shall be reported to DET via Det_ReportRuntimeError.⌋ (SRS_IpduM_02820)

A header with a payload length greater than the remaining byte is invalid. No further header is to be expected behind it.

[SWS_IpduM_00214] ⌈If the remaining bytes in a Container PDU are less than the configured IpduMContainerHeaderSize (ECUC_IpduM_00183) the remaining bytes shall be ignored.⌋ (SRS_IpduM_02820)

DRAFT: [SWS_IpduM_00237] ⌈When processing a received Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, the IpduM shall ignore all contained PDUs that are according to their configuration not or not completely contained in the received Container PDU. Such contained I-PDUs shall not be indicated to the upper layer. If Development Error Detection is configured (ECUC_IpduM_00132) *IPDUM_E_CONTAINER* shall be reported to DET via Det_ReportError.⌋ (SRS_IpduM_02825)

7.3.10 Metadata handling

The requirements of this section only apply if IpduMMetaDataSupport is configured to *true*.

[SWS_IpduM_00228] ⌈In case a Container PDU supports MetaData, the IpduM shall use the MetaData last collected from the contained I-PDUs when sending the Container PDU.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00229] ⌈In case the IpduM receives a Container PDU with MetaData, the IpduM shall forward the MetaData of the Container PDU along with all contained I-PDU that support MetaData.⌋ (SRS_IpduM_02820)

The IpduM does not rearrange MetaData. Thus, it only supports contained I-PDUs assigned to the same Container PDUs, which have no MetaData or have the same MetaDataType, see SWS_IpduM_00230.

7.4 Error classification

7.4.1 Development Errors

[SWS_IpduM_00026] ⌈API service called with wrong parameter:
- error code: IPDUM_E_PARAM
- value [hex]: 0x10

⌋ (SRS_BSW_00337)

[SWS_IpduM_00162] [NULL pointer checking
- error code: IPDUM_E_PARAM_POINTER
- value [hex]: 0x11

] (SRS_BSW_00337, SRS_BSW_00414)

[SWS_IpduM_00153] [API service (except IpduM_MainFunctionTx, IpduM_MainFunctionRx and IpduM_GetVersionInfo) used without module initialization
- error code: IPDUM_E_UNINIT
- value [hex]: 0x20

] (SRS_BSW_00337)

[SWS_IpduM_00174] [Invalid configuration set selection
- error code: IPDUM_E_INIT_FAILED
- value [hex]: 0x21

] (SRS_BSW_00414)

7.4.2 Runtime Errors

[SWS_IpduM_00215] [Erroneous header detected
- error code: IPDUM_E_HEADER
- value [hex]: 0x30

] (SRS_IpduM_02820)

[SWS_IpduM_00216] [Container Queue overflow
- error code: IPDUM_E_QUEUEOVFL
- value [hex]: 0x31

] (SRS_IpduM_02820)

[SWS_IpduM_00247] [Partly or erroneous container received
- error code: IPDUM_E_CONTAINER
- value [hex]: 0x32

] (SRS_IpduM_02825)

7.4.3 Transient Faults

There are no transient faults.

7.4.4 Production Errors

There are no production errors.

7.4.5 Extended Production Errors

There are no extended production errors.

7.5 Error detection and notification

[SWS_IpduM_00028] ⌈If IpduMDevErrorDetect is configured to TRUE, all IpduM APIs shall check their input parameters and report detected errors to DET via Det_ReportError. IPDUM_E_PARAM shall be reported for normal parameters and IPDUM_E_PARAM_POINTER for pointer parameters.⌋ (SRS_BSW_00323)

8 API specification

8.1 Imported types

This chapter lists all imported types and the corresponding modules.

[SWS_IpduM_00102] [

Module	Header File	Imported Type
ComStack_Types	ComStackTypes.h	PduIdType
	ComStackTypes.h	PduInfoType
Std_Types	StandardTypes.h	Std_ReturnType
	StandardTypes.h	Std_VersionInfoType

] (SRS_BSW_00357)

8.2 Type definitions

8.2.1 IpduM_ConfigType

[SWS_IpduM_00159] [

Name:	`IpduM_ConfigType`
Type:	`Structure`
Range:	Implementation specific.
Description:	This is the type of the data structure containing the initialization data for the I-PDU multiplexer.
Available via:	`IpduM.h`

] (SRS_BSW_00438)

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 IpduM_Init

[SWS_IpduM_00032] [

Service name:	IpduM_Init

Syntax:	`void IpduM_Init(`
	`  const IpduM_ConfigType* config`
	`)`

Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	config | Implementation specific structure with configuration parameters.
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Initializes the I-PDU Multiplexer.

Available via:	IpduM.h

] (SRS_BSW_00344, SRS_BSW_00405, SRS_BSW_00101, SRS_BSW_00369)

[SWS_IpduM_00033] [The function IpduM_Init shall initialize all module-related global variables.] (SRS_BSW_00101)

[SWS_IpduM_00084] [The behavior of the IpduM is unspecified until a correct call to IpduM_Init is made.] (SRS_BSW_00406)

8.3.2 IpduM_GetVersionInfo

[SWS_IpduM_00037] [

Service name:	IpduM_GetVersionInfo	
Syntax:	`void IpduM_GetVersionInfo( Std_VersionInfoType* versioninfo )`	
Service ID[hex]:	0x01	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	versioninfo	Pointer to where to store the version information of this module.
Return value:	None	
Description:	Service returns the version information of this module.	
Available via:	IpduM.h	

] (SRS_BSW_00407, SRS_BSW_00369, SRS_BSW_00003)

8.3.3 IpduM_Transmit

[SWS_IpduM_00043] [

Service name:	IpduM_Transmit	
Syntax:	`Std_ReturnType IpduM_Transmit( PduIdType TxPduId, const PduInfoType* PduInfoPtr )`	
Service ID[hex]:	0x49	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	Identifier of the PDU to be transmitted
	PduInfoPtr	Length of and pointer to the PDU data and pointer to MetaData.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Transmit request has been accepted. E_NOT_OK: Transmit request has not been accepted.
Description:	Requests transmission of a PDU.	
Available via:	IpduM.h	

] (SRS_BSW_00369)

For a detailed description read Chapter 7.2.4.1.

8.4 Call-back notifications

8.4.1 IpduM_RxIndication

[SWS_IpduM_00040] ⌈

Service name:	IpduM_RxIndication	

Syntax:	`void IpduM_RxIndication(`
	`  PduIdType RxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID[hex]:	0x42	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	RxPduId	ID of the received PDU.
	PduInfoPtr	Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Indication of a received PDU from a lower layer communication interface module.	
Available via:	`IpduM.h`	

⌋ (SRS_BSW_00369, SRS_IpduM_02817)

[SWS_IpduM_00041] ⌈If there is a static part configured in a multiplexed SDU received from the PDU Router, the function IpduM_RxIndication transforms the incoming I-PDU ID into the correct I-PDU ID for the static part's destination and then forwards the SDU via the PDU Router, see PduR_IpduMRxIndication in the PDU Router SWS.⌋ (SRS_IpduM_02812)

[SWS_IpduM_00042] ⌈When a multiplexed I-PDU is received from the PDU Router the function IpduM_RxIndication uses the incoming I-PDU ID and the selector field to find out the correct I-PDU ID for the dynamic part's destination and then forwards the I-PDU via the PDU Router, see PduR_IpduMRxIndication in the PDU Router SWS.⌋ (SRS_IpduM_02812)

[SWS_IpduM_00217] ⌈When a Container PDU is received from the PDU Router, the function IpduM_RxIndication forwards the contained I-PDUs via the PDU Router, using PduR_IpduMRxIndication (see SWS_IpduM_00105) .⌋ (SRS_IpduM_02820)

[SWS_IpduM_00086] ⌈The function IpduM_RxIndication shall be callable in interrupt context, e.g. from receive interrupt. ⌋ (SRS_IpduM_02812)

8.4.2 IpduM_TxConfirmation

[SWS_IpduM_00044] ⌈

Service name:	IpduM_TxConfirmation

Syntax:	`void IpduM_TxConfirmation(`
	`  PduIdType TxPduId,`
	`  Std_ReturnType result`
	`)`

Service ID[hex]:	0x40	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	ID of the PDU that has been transmitted.
	result	E_OK: The PDU was transmitted. E_NOT_OK: Transmission of the PDU failed.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.	
Available via:	`IpduM.h`	

] (SRS_BSW_00369)

[SWS_IpduM_00088] [The function IpduM_TxConfirmation shall translate the confirmation received from the PDU Router into confirmations for the I-PDUs which where contained in the sent multiplexed I-PDU or Container PDU.] (SRS_IpduM_02814)

These confirmations are given again to the PDU Router that has to route them to COM.

[SWS_IpduM_00087] [The function IpduM_TxConfirmation shall be callable in interrupt context, e.g. from a transmit interrupt.] (SRS_IpduM_02814)

8.4.3 IpduM_TriggerTransmit

[SWS_IpduM_00060] [

Service name:	IpduM_TriggerTransmit	

Syntax:	`Std_ReturnType IpduM_TriggerTransmit(`
	`  PduIdType TxPduId,`
	`  PduInfoType* PduInfoPtr`
	`)`	

Service ID[hex]:	0x41	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	ID of the SDU that is requested to be transmitted.
Parameters (inout):	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description:	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.	
Available via:	`IpduM.h`	

] (SRS_BSW_00369)

[SWS_IpduM_00090] [Within the function IpduM_TriggerTransmit, the IpduM shall copy the contents of its I-PDU transmit buffer to the PDU buffer given by PduInfoPtr->SduDataPtr and update PduInfoPtr->SduLength with length of the copied data accordingly.] (SRS_IpduM_02810)

[SWS_IpduM_00091] [The IpduM shall take care about the data consistency during providing the data.] (SRS_IpduM_02810)

Use case: This function is used e.g. by the LIN Master for sending out a LIN frame. In this case, the trigger transmit can be initiated by the Master schedule table itself or a received LIN header.
This function is also used by the FlexRay Interface for requesting PDUs to be sent in static part (synchronous to the FlexRay global time).

[SWS_IpduM_00089] [The function IpduM_TriggerTransmit shall be callable in interrupt context.] (SRS_IpduM_02810)

8.5 Scheduled functions

Many of the functions of the IpduM module are called synchronous in the context of the upper layer (for transmission) and in the context of the lower layer (for reception). However, some functionality is excuted deferred. Therefore, it is performed in either IpduM_MainFunctionTx or IpduM_MainFunctionRx, dependent on the respective functionality is associated to the sender or receiver side respectively.

[SWS_IpduM_91002] [

Service name:	IpduM_MainFunctionTx

Syntax:	`void IpduM_MainFunctionTx(`
	`  void`
	`)`

Service ID[hex]:	0x12
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This function performs the processing of the transmission activities that are not directly handled within the calls from PduR.
Available via:	`IpduM_SchM.h`

] (SRS_BSW_00432)

[SWS_IpduM_91001] [

Service name:	IpduM_MainFunctionRx

Syntax:	`void IpduM_MainFunctionRx(`
	`  void`
	`)`

Service ID[hex]:	0x11
Sync/Async:	Synchronous

Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This function performs the processing of the reception activities that are not directly handled within the calls from PduR.
Available via:	`IpduM_SchM.h`

] (SRS_BSW_00432)

8.6 Expected Interfaces

In this chapter, all interfaces required from other modules are listed.

8.6.1 Mandatory Interfaces

This chapter defines all interfaces that are required to fulfill the core functionality of the module.

[SWS_IpduM_00104] [

API function	Header File	Description
Det_ReportRuntimeError	Det.h	Service to report runtime errors. If a callout has been configured then this callout shall be called.

] (SRS_BSW_00009)

8.6.2 Optional Interfaces

This chapter defines all interfaces that are required to fulfill an optional functionality of the module.

[SWS_IpduM_00105] [

API function	Header File	Description
Det_ReportError	Det.h	Service to report development errors.
PduR_IpduMRxIndication	PduR_IpduM.h	Indication of a received PDU from a lower layer communication interface module.
PduR_IpduMTransmit	PduR_IpduM.h	Requests transmission of a PDU.
PduR_IpduMTriggerTransmit	PduR_IpduM.h	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.
PduR_IpduMTxConfirmation	PduR_IpduM.h	The lower layer communication interface module confirms the transmission of a PDU, or the failure to transmit a PDU.

] (SRS_BSW_00009)

8.6.3 Configurable interfaces

Not applicable

9 Sequence diagrams

9.1 Transmission of a multiplexed I-PDU and Transmit confirmation

The following sequence chart shows a transmit request initiated by the COM layer. The transmit request is for an I-PDU which has to be transmitted within a multiplexed I-PDU. In the IpduM module is configured that this transmitted I-PDU triggers the sending of the multiplexed I-PDU.

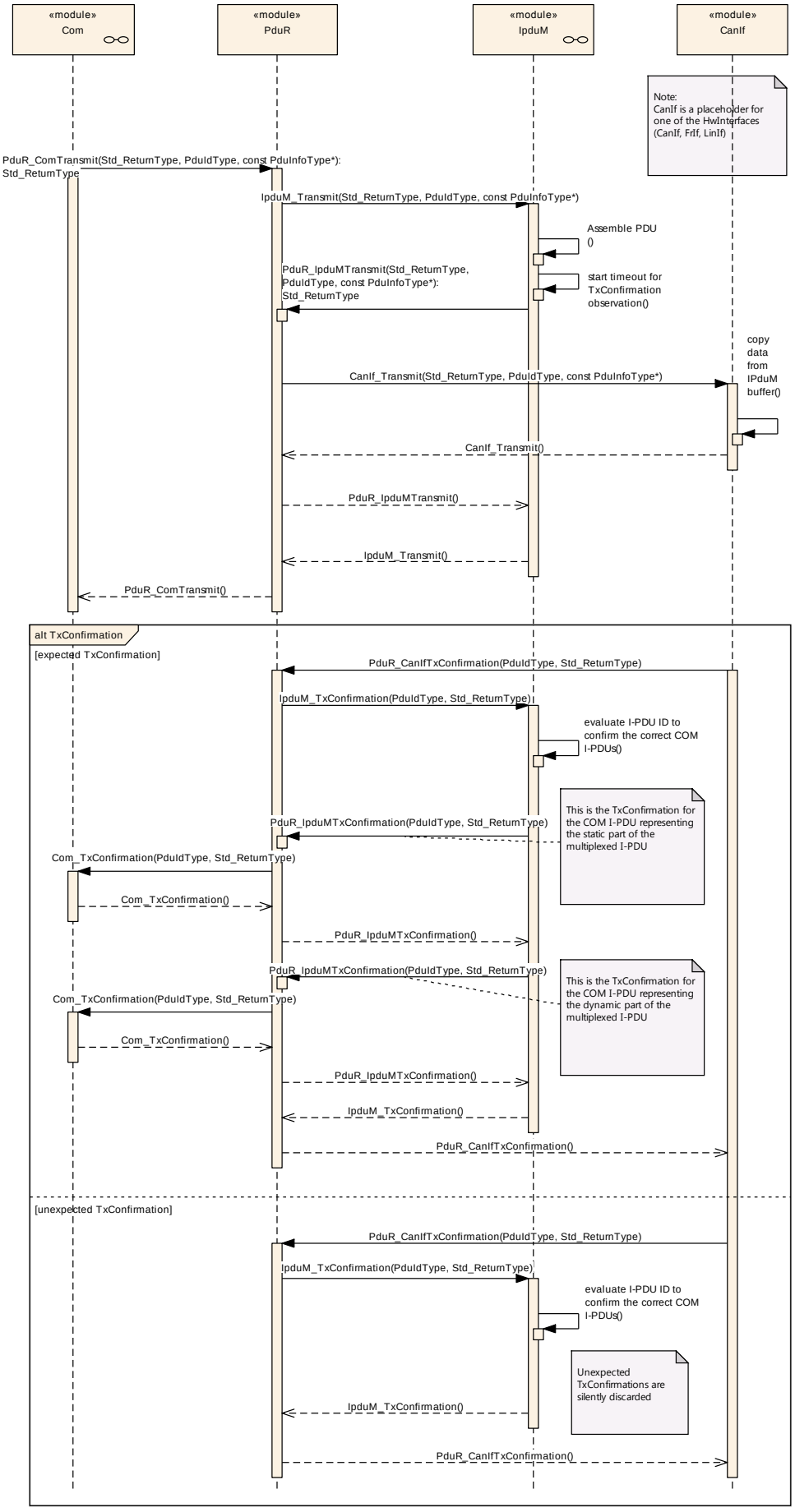
«module»
Com
«module»
PduR
«module»
IpduM
«module»
CanIf
Note:
CanIf is a placeholder for one of the HwInterfaces (CanIf, FrIf, LinIf)
PduR_ComTransmit(Std_ReturnType, PduIdType, const PduInfoType*):
Std_ReturnType
IpduM_Transmit(Std_ReturnType, PduIdType, const PduInfoType*)
Assemble PDU ()
PduR_IpduMTransmit(Std_ReturnType, PduIdType, const PduInfoType*):
Std_ReturnType
start timeout for TxConfirmation observation()
copy data from IPduM buffer()
CanIf_Transmit(Std_ReturnType, PduIdType, const PduInfoType*)
CanIf_Transmit()
PduR_IpduMTransmit()
IpduM_Transmit()
PduR_ComTransmit()
alt TxConfirmation
[expected TxConfirmation]
PduR_CanIfTxConfirmation(PduIdType, Std_ReturnType)
IpduM_TxConfirmation(PduIdType, Std_ReturnType)
evaluate I-PDU ID to confirm the correct COM I-PDUs()
This is the TxConfirmation for the COM I-PDU representing the static part of the multiplexed I-PDU
PduR_IpduMTxConfirmation(PduIdType, Std_ReturnType)
Com_TxConfirmation(PduIdType, Std_ReturnType)
Com_TxConfirmation()
PduR_IpduMTxConfirmation()
PduR_IpduMTxConfirmation(PduIdType, Std_ReturnType)
This is the TxConfirmation for the COM I-PDU representing the dynamic part of the multiplexed I-PDU
Com_TxConfirmation(PduIdType, Std_ReturnType)
Com_TxConfirmation()
PduR_IpduMTxConfirmation()
IpduM_TxConfirmation()
PduR_CanIfTxConfirmation()
[unexpected TxConfirmation]
PduR_CanIfTxConfirmation(PduIdType, Std_ReturnType)
IpduM_TxConfirmation(PduIdType, Std_ReturnType)
evaluate I-PDU ID to confirm the correct COM I-PDUs()
Unexpected TxConfirmations are silently discarded
IpduM_TxConfirmation()
PduR_CanIfTxConfirmation()

Figure 5 Transmission and confirmation of multiplexed I-PDU with triggering

9.2 Transmission of a multiplexed I-PDU without Trigger

The following sequence chart shows a transmit request initiated by the COM layer. Because of the configuration of the IpduM, no transmit request for the IpduM I-PDU takes place. For configuration see ECUC_IpudM_00052.

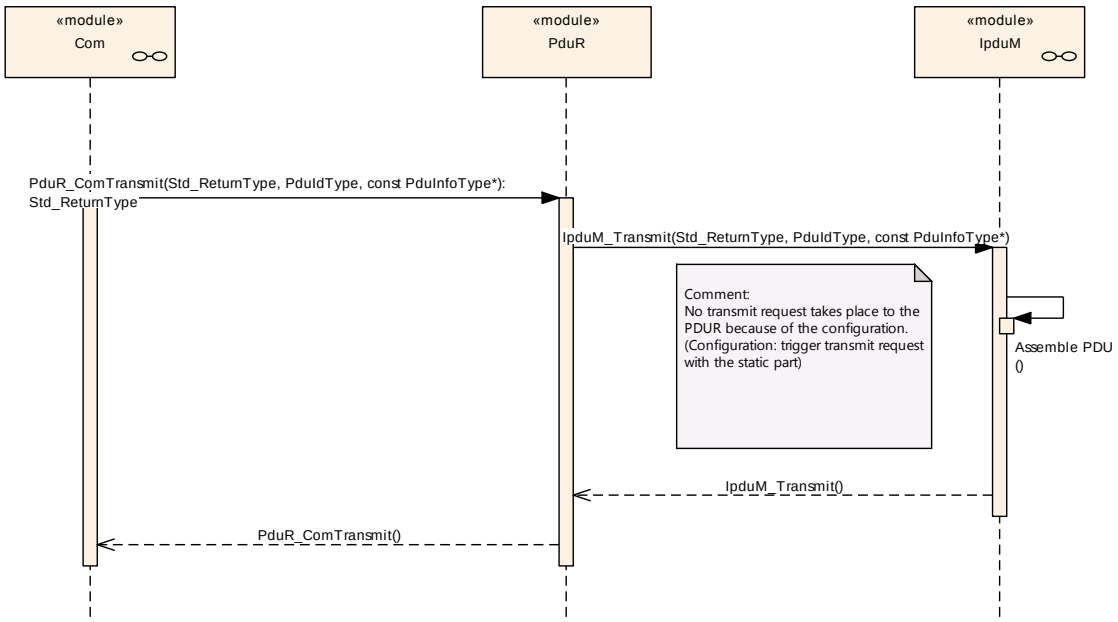

Figure 6 Transmission of a multiplexed I-PDU without triggering

9.3 Reception of the multiplexed I-PDU

The following sequence chart shows a reception of a multiplexed I-PDU. The I-PDU contains a static and a dynamic part and both are configured to create an RxIndication to the PDU Router module.

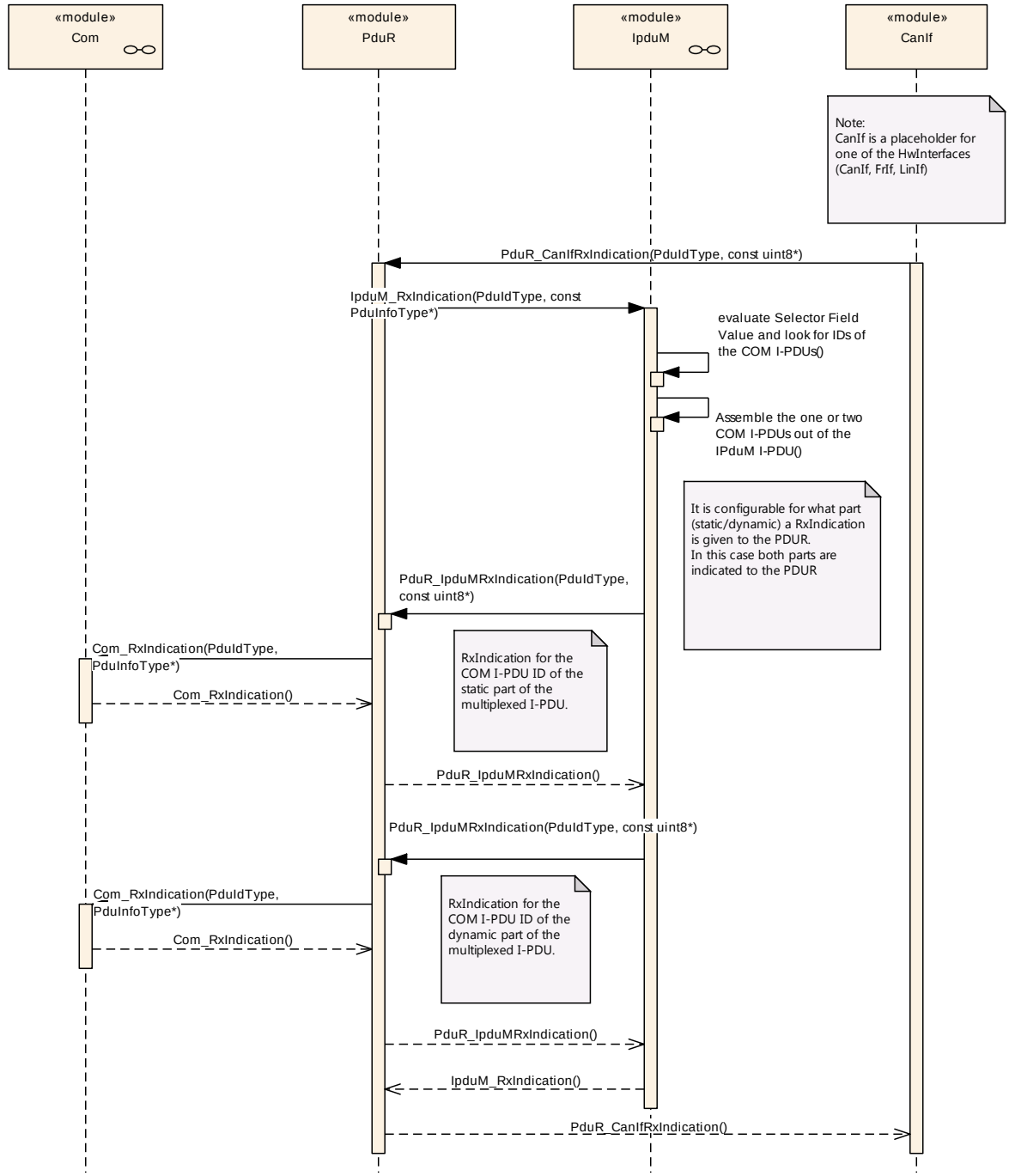

Figure 7 Reception of a multiplexed I-PDU

9.4 Trigger Transmit

The following sequence chart shows a Trigger Transmit request from an interface layer.

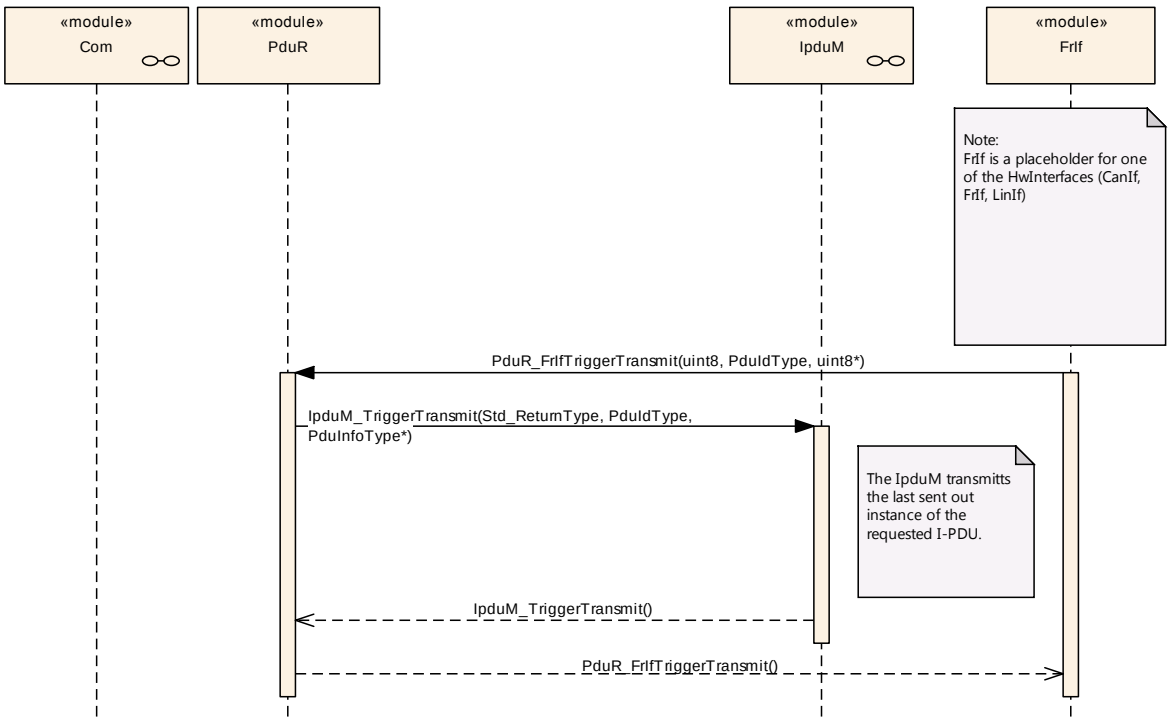

Figure 8 Trigger Transmit request from interface layer

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers.

Chapter 10.2 specifies the structure (containers) and the parameters of the module IpduM.

Chapter 10.3 specifies published information of the module IpduM.

10.1 How to read this chapter

For details, refer to the chapter 10.1 Introduction to configuration specification in SWS_BSWGeneral.

10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 7 and Chapter 8.

10.2.1 Configuration overview

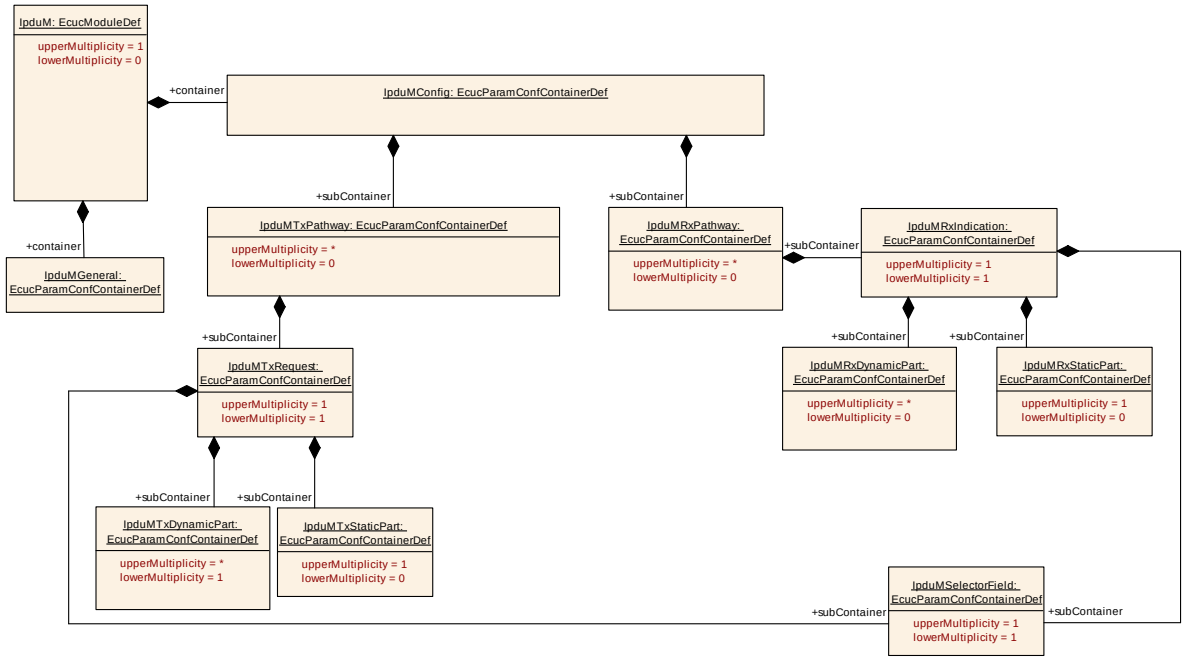

Figure 9 IpduM Configuration Overview

10.2.2 IpduM

SWS Item	**ECUC_IpduM_00204 :**
Module Name	**IpduM**
Module Description	Configuration of the IpduM (Ipdu Multiplexer) module.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMConfig	1	This container contains the sub containers of the IpduM module. • The IpduMTxPathway subcontainer includes information about sent I-PDUs. • The IpduMRxPathway includes information about received I-PDUs. • The IpduMContainerTxPdu and IpduMContainedTxPdu include information about the sending of ContainerPdus. • The IpduMContainerRxPdu and IpduMContainedRxPdu include information about the reception of ContainerPdus.
IpduMGeneral	1	Contains the general configuration parameters of IpduM.
IpduMPublishedInformation	1	Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.

10.2.3 IpduMConfig

SWS Item	**ECUC_IpduM_00059 :**
Container Name	IpduMConfig
Description	This container contains the sub containers of the IpduM module. • The IpduMTxPathway subcontainer includes information about sent I-PDUs. • The IpduMRxPathway includes information about received I-PDUs. • The IpduMContainerTxPdu and IpduMContainedTxPdu include information about the sending of ContainerPdus. • The IpduMContainerRxPdu and IpduMContainedRxPdu include information about the reception of ContainerPdus.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00166 :**
Name	IpduMMaxTxBufferSize
Parent Container	IpduMConfig
Description	Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.
Multiplicity	0..1

Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00165 :**		
Name	IpduMMaxTxPathwayCnt		
Parent Container	IpduMConfig		
Description	Maximum number of transmitted IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMContainedRxPdu	0..*	Configuration of a received contained Pdu.
IpduMContainedTxPdu	0..*	Configuration of a sender ContainedPdu.
IpduMContainerRxPdu	0..*	Configuration of a receiver ContainerPdu which may collect several ContainedPdus.
IpduMContainerTxPdu	0..*	Configuration of a transmitted container Pdu.
IpduMRxPathway	0..*	includes information about received I-PDUs
IpduMTxPathway	0..*	includes information about sent I-PDUs

10.2.4 IpduMGeneral

SWS Item	**ECUC_IpduM_00130 :**
Container Name	IpduMGeneral
Description	Contains the general configuration parameters of IpduM.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00209 :**		
Name	IpduMContainedTxPduPriorityHandling		
Parent Container	IpduMGeneral		
Description	This parameter enables/disables handling of priority for IpduMContainedTxPdu's with IpduMContainedTxPduCollectionSemantics IPDUM_LAST_IS_BEST. true: enabled false: disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00132 :**		
Name	IpduMDevErrorDetect		
Parent Container	IpduMGeneral		
Description	Switches the development error detection and notification on or off. • true: detection and notification is enabled. • false: detection and notification is disabled.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00197 :**		
Name	IpduMHeaderByteOrder		
Parent Container	IpduMGeneral		
Description	This parameter defines the ByteOrder of the headers inside a Container I-PDU.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	IPDUM_BIG_ENDIAN	Headers inside a Container I-PDU shall be ordered big endian.	
	IPDUM_LITTLE_ENDIAN	Headers inside a Container I-PDU shall be ordered little endian.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Depen-	scope: local		

dency	

SWS Item	**ECUC_IpduM_00205 :**		
Name	IpduMMetaDataSupport		
Parent Container	IpduMGeneral		
Description	This parameter enables/disables the support of meta-data feature. true: enabled false: disabled		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00200 :**		
Name	IpduMRxTimeBase		
Parent Container	IpduMGeneral		
Description	The period between successive calls to IpduM_MainFunctionRx in seconds. This parameter may be used by the IpduM generator to transform the values of the reception related timing configuration parameters of the IpduM module to internal implementation specific counter or tick values. The IpduM module's internal timing handling is implementation specific. The IpduM module (generator) may rely on the fact that IpduM_MainFunctionRx is scheduled according to the value configured here.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00133 :**
Name	IpduMStaticPartExists
Parent Container	IpduMGeneral
Description	This is to allow optimizations in the case the IpduM will never be used with a static part.

	Note that this is a pre-compile option. If this is set to False then it will not be possible to add static parts after compilation. True: A static part may exist. False: A static part will never exist.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00201 :**		
Name	IpduMTxTimeBase		
Parent Container	IpduMGeneral		
Description	The period between successive calls to IpduM_MainFunctionTx in seconds. This parameter may be used by the IpduM generator to transform the values of the reception related timing configuration parameters of the IpduM module to internal implementation specific counter or tick values. The IpduM module's internal timing handling is implementation specific. The IpduM module (generator) may rely on the fact that IpduM_MainFunctionTx is scheduled according to the value configured here.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	]0 .. INF[		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00134 :**		
Name	IpduMVersionInfoApi		
Parent Container	IpduMGeneral		
Description	Active/Deactivate the version information API. true: version information activated false: version information deactivated		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	

Scope / Dependency	scope: local

No Included Containers

10.2.5 IpduMTxPathway

SWS Item	**ECUC_IpduM_00070 :**		
Container Name	IpduMTxPathway		
Description	Contains the configuration parameters transmitted I-PDUs by the IpduM module.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMTxRequest	1	configuration for a TxRequest

10.2.6 IpduMTxRequest

SWS Item	**ECUC_IpduM_00052 :**
Container Name	IpduMTxRequest
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.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00162 :**		
Name	IpduMByteOrder		
Parent Container	IpduMTxRequest		
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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

Scope / Dependency	scope: local

SWS Item	**ECUC_IpduM_00121 :**		
Name	IpduMIPduUnusedAreasDefault		
Parent Container	IpduMTxRequest		
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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00158 :**		
Name	IpduMTxConfirmationPduId		
Parent Container	IpduMTxRequest		
Description	Handle Id used by the PduR for confirmation (IpduM_TxConfirmation) and for TriggerTransmit (IpduM_TriggerTransmit). The existence of this parameter is essential for the PduR generation tool to actually find a symbolicNameValue for the OutgoingPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00125 :**	
Name	IpduMTxTriggerMode	
Parent Container	IpduMTxRequest	
Description	Selects whether to send the multiplexed I-PDU immediately or at some later date.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	DYNAMIC_PART_TRIGGER	Writing the I-PDU representing the dynamic part does trigger a sending of the I-PDU.
	NONE	Only the buffer in the IpduM are written but not send is triggered,

			used for IpduM I-PDUs which are requested by TriggerTransmit.
	STATIC_OR_DYNAMIC_PART_TRIGGER		Writing the I-PDU representing the static or the dynamic part does trigger a sending of the I-PDU.
	STATIC_PART_TRIGGER		Writing the I-PDU representing the static part does trigger a sending of the I-PDU.
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00157 :**		
Name	IpduMInitialDynamicPart		
Parent Container	IpduMTxRequest		
Description	Reference to the dynamic part that shall be used to initialize this multiplexed TX-I-PDU.		
Multiplicity	1		
Type	Reference to [IpduMTxDynamicPart]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00120 :**		
Name	IpduMOutgoingPduRef		
Parent Container	IpduMTxRequest		
Description	Reference to the PDU defining the outgoing I-PDU. When the outgoing I-PDU is sent this is the I-PDU ID to give it. It is the IpduM I-PDU ID of the assembled I-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMSelectorField	1	Specifies the position of the selector field in the outgoing I-PDU.
IpduMTxDynamicPart	1..*	This (These) included container(s) must exist for each unique selector field value for this outgoing IpduM I-PDU.
IpduMTxDynamicSegment	1..*	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 IpduMTxDynamicPart 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.
IpduMTxStaticPart	0..1	This included container configures the static part, if present.
IpduMTxStaticSegment	0..*	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.

10.2.7 IpduMTxDynamicPart

SWS Item	**ECUC_IpduM_00056 :**		
Container Name	IpduMTxDynamicPart		
Description	Configuration parameters for an instance of a TxRequest call into the IpduM. When a Tx Request with the IpduMTxDynamicHandleId 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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00167 :**		
Name	IpduMJitUpdate		
Parent Container	IpduMTxDynamicPart		
Description	If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00163 :**		
Name	IpduMTxDynamicConfirmation		
Parent Container	IpduMTxDynamicPart		
Description	A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the dynamic part is generated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00127 :**		
Name	IpduMTxDynamicHandleId		
Parent Container	IpduMTxDynamicPart		
Description	This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured dynamic segments are copied and the IpduMTxTriggerMode is honored.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00126 :**		
Name	IpduMTxDynamicPduRef		
Parent Container	IpduMTxDynamicPart		
Description	Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.8 IpduMTxDynamicSegment

SWS Item	**ECUC_IpduM_00168 :**
Container Name	IpduMTxDynamicSegment
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 crea-

	ted 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 IpduMTxDynamicPart 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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Parent Container	IpduMTxDynamicSegment		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Parent Container	IpduMTxDynamicSegment		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.9 IpduMTxStaticPart

SWS Item	**ECUC_IpduM_00082 :**
Container Name	IpduMTxStaticPart
Description	Configuration parameters for an instance of a Tx_Request call into the IpduM. When a Tx Request with the IpduMTxStaticHandleId 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 TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the static part if it exists.

Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00167 :**		
Name	IpduMJitUpdate		
Parent Container	IpduMTxStaticPart		
Description	If configured to true fetch the data of this part Just-In-Time via the trigger-Transmit API of the PduR.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00164 :**		
Name	IpduMTxStaticConfirmation		
Parent Container	IpduMTxStaticPart		
Description	A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the static part is generated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00129 :**		
Name	IpduMTxStaticHandleId		
Parent Container	IpduMTxStaticPart		
Description	This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured static segments are copied and the IpduMTxTriggerMode is honored.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00128 :**		

Name	IpduMTxStaticPduRef		
Parent Container	IpduMTxStaticPart		
Description	Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.10 IpduMTxStaticSegment

SWS Item	**ECUC_IpduM_00171 :**		
Container Name	IpduMTxStaticSegment		
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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Parent Container	IpduMTxStaticSegment		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**	
Name	IpduMSegmentPosition	
Parent Container	IpduMTxStaticSegment	
Description	Segments bit position in the multiplexed Pdu.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 2031	

Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.11 IpduMRxPathway

SWS Item	**ECUC_IpduM_00071 :**		
Container Name	IpduMRxPathway		
Description	Contains the configuration parameters received I-PDUs by the IpduM module.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMRxIndication	1	configuration for RxIndication

10.2.12 IpduMRxIndication

SWS Item	**ECUC_IpduM_00047 :**
Container Name	IpduMRxIndication
Description	Contains the configuration for incoming RxIndication calls.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00162 :**	
Name	IpduMByteOrder	
Parent Container	IpduMRxIndication	
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.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	--
	LITTLE_ENDIAN	--
Post-Build Variant Value	true	

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00109 :**		
Name	IpduMRxHandleId		
Parent Container	IpduMRxIndication		
Description	This is the I-PDU ID of the incoming I-PDU. If an incoming RxIndication's I-PDU ID matches this value then it is unpacked according to the specification in this container.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00108 :**		
Name	IpduMRxIndicationPduRef		
Parent Container	IpduMRxIndication		
Description	Reference to the received Pdu representation in the ECU Configuration Description exchange file.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMRxDynamicPart	0..*	Each of these containers contains the configuration for one value of the selector field for the incoming I-PDU's dynamic part.
IpduMRxDynamicSegment	0..*	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 IpduMRxDynamicPart 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.
IpduMRxStaticPart	0..1	This contains the configuration for the incoming I-PDU's static part. If the incoming I-PDU has no static part then this is omitted.
IpduMRxStaticSegment	0..*	The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several seg-

		ments. 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 IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.
IpduMSelectorField	1	This contains the location of the selector field. At run-time, the selector field is used to select which dynamic part is unpacked.

10.2.13 IpduMRxDynamicPart

SWS Item	**ECUC_IpduM_00048 :**		
Container Name	IpduMRxDynamicPart		
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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00113 :**		
Name	IpduMRxSelectorValue		
Parent Container	IpduMRxDynamicPart		
Description	This is the selector value that this container refers to.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00112 :**
Name	IpduMOutgoingDynamicPduRef
Parent Container	IpduMRxDynamicPart
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.

Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.14 IpduMRxDynamicSegment

SWS Item	**ECUC_IpduM_00170 :**		
Container Name	IpduMRxDynamicSegment		
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 IpduMRxDynamicPart 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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Parent Container	IpduMRxDynamicSegment		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Parent Container	IpduMRxDynamicSegment		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE

	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.15 IpduMRxStaticPart

SWS Item	**ECUC_IpduM_00049 :**		
Container Name	IpduMRxStaticPart		
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.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00115 :**		
Name	IpduMOutgoingStaticPduRef		
Parent Container	IpduMRxStaticPart		
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.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.16 IpduMRxStaticSegment

SWS Item	**ECUC_IpduM_00169 :**
Container Name	IpduMRxStaticSegment
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 IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.
Post-Build Variant Multiplicity	true

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Parent Container	IpduMRxStaticSegment		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Parent Container	IpduMRxStaticSegment		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.17 IpduMSelectorField

SWS Item	**ECUC_IpduM_00054 :**		
Container Name	IpduMSelectorField		
Description	This contains the location and the length of the selector field.		
Configuration Parameters			

SWS Item	**ECUC_IpduM_00160 :**		
Name	IpduMSelectorFieldLength		
Parent Container	IpduMSelectorField		
Description	Length of the selector field in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 16		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE

	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00161 :**		
Name	IpduMSelectorFieldPosition		
Parent Container	IpduMSelectorField		
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.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

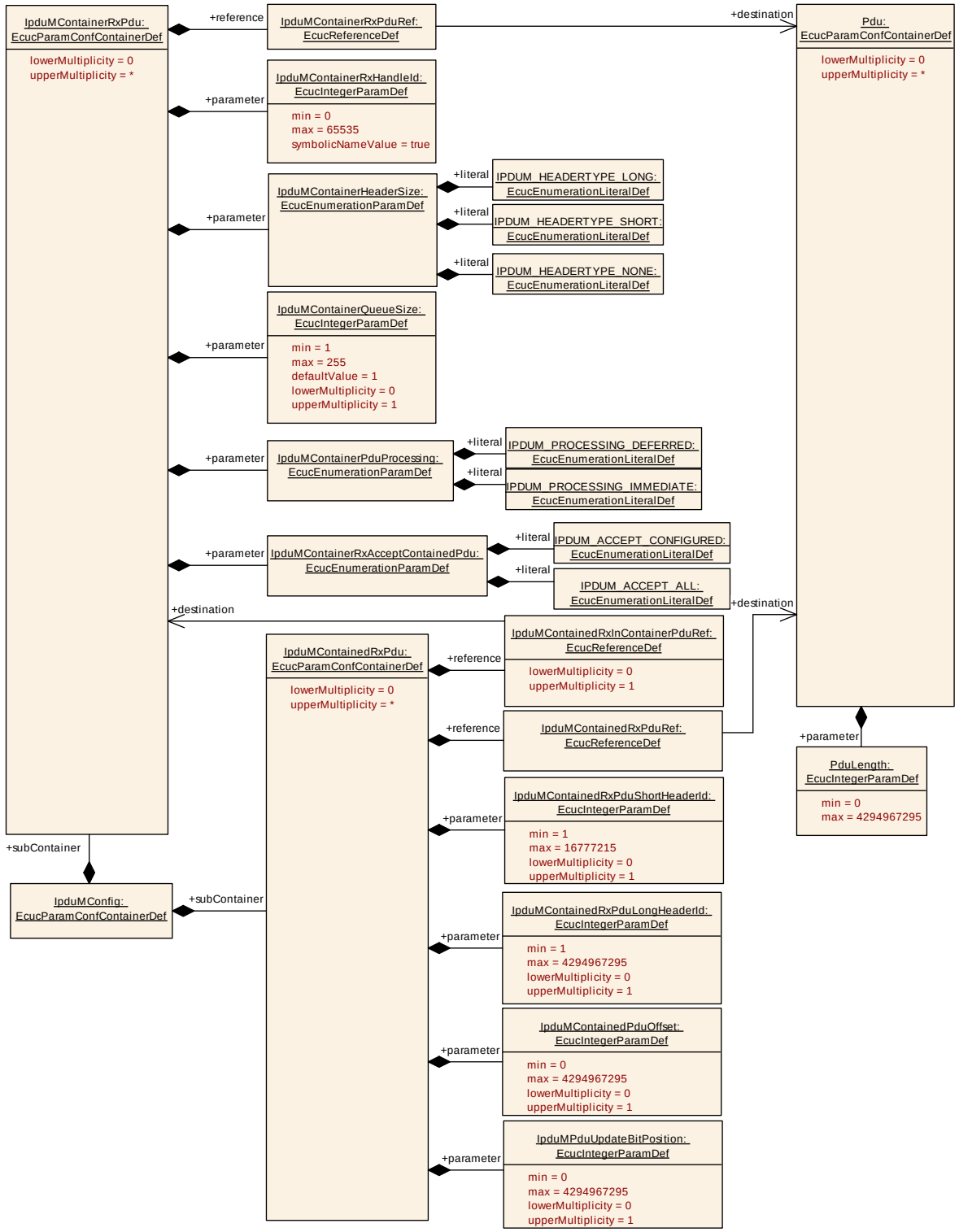

Figure 10 Configuration Overview RxContainer

10.2.18 IpduMContainerRxPdu

SWS Item	**ECUC_IpduM_00188 :**
Container Name	IpduMContainerRxPdu
Description	Configuration of a receiver ContainerPdu which may collect several Con-

	tainedPdus.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00183 :**		
Name	IpduMContainerHeaderSize		
Parent Container	IpduMContainerRxPdu		
Description	Defines the layout of the header information (header id and length).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_HEADERTYPE_LONG	Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
	IPDUM_HEADERTYPE_NONE	Static Container Layout **Tags:** atp.Status=draft	
	IPDUM_HEADERTYPE_SHORT	Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00184 :**		
Name	IpduMContainerPduProcessing		
Parent Container	IpduMContainerRxPdu		
Description	Defines whether the handling of this ContainerPdu shall be done in the context of the caller (IMMEDIATE) or in the next call to IpduM_MainFunctionRx (DEFERRED).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_PROCESSING_DEFERRED	--	
	IPDUM_PROCESSING_IMMEDIATE	--	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00185 :**
Name	IpduMContainerQueueSize
Parent Container	IpduMContainerRxPdu
Description	Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.
Multiplicity	0..1
Type	EcucIntegerParamDef

Range	1 .. 255		
Default value	1		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00186 :**		
Name	IpduMContainerRxAcceptContainedPdu		
Parent Container	IpduMContainerRxPdu		
Description	Defines for the received IpduMContainerRxPdu whether the list of referencing IpduM-ContainedRxPdus (via the reference IpduMContainedPduContainerRefRx) is a closed set.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_ACCEPT_ALL	The IpduMContainedRxPdus which are referencing this IpduMContainerRxPdu are expected inside this IpduMContai-nerRxPdu, but there may also occur other Pdus inside this IpduMContai-nerRxPdu as well. This also supports the case where no IpduMContai-nedRxPdu references the IpduMCon-tainerRxPdu.	
	IPDUM_ACCEPT_CONFIGURED	Only the IpduMContainedRxPdus which are referencing this IpduMCon-tainerRxPdu are expected inside this IpduMContainerRxPdu.	
Post-Build Variant Value	true		
Value Configura-tion Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Depen-dency	scope: local		

SWS Item	**ECUC_IpduM_00187 :**		
Name	IpduMContainerRxHandleId		
Parent Container	IpduMContainerRxPdu		
Description	Handle Id used by the PduR for RxIndication.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00189 :**
Name	IpduMContainerRxPduRef

Parent Container	IpduMContainerRxPdu		
Description	Reference to the Pdu which represents the container and is used for reception.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.19 IpduMContainedRxPdu

SWS Item	**ECUC_IpduM_00174 :**		
Container Name	IpduMContainedRxPdu		
Description	Configuration of a received contained Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00206 :**		
Name	IpduMContainedPduOffset		
Parent Container	IpduMContainedRxPdu		
Description	Static offset (in bytes) of the ContainedPdu. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE. - only the ContainedPdu with the highest offset within a ContainerPdu may have variable length.		

SWS Item	**ECUC_IpduM_00203 :**		
Name	IpduMContainedRxPduLongHeaderId		
Parent Container	IpduMContainedRxPdu		
Description	LongHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		

Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: Only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_LONG		

SWS Item	**ECUC_IpduM_00202 :**		
Name	IpduMContainedRxPduShortHeaderId		
Parent Container	IpduMContainedRxPdu		
Description	ShortHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 16777215		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: Only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT		

SWS Item	**ECUC_IpduM_00207 :**		
Name	IpduMPduUpdateBitPosition		
Parent Container	IpduMContainedRxPdu		
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). **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE.		

SWS Item	**ECUC_IpduM_00173 :**
Name	IpduMContainedRxInContainerPduRef

Parent Container	IpduMContainedRxPdu		
Description	Optional reference to an IpduMContainerRxPdu this IpduMContainedRxPdu may be received in. If this IpduMContainedRxPdu shall be received in exactly one IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_CONFIGURED then the IpduMContainedRxInContainerPduRef shall be defined. If this IpduMContainedRxPdu can be received in any IpduMContainerRxPdu with IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL then the IpduMContainedRxInContainerPduRef shall NOT be defined.		
Multiplicity	0..1		
Type	Reference to [IpduMContainerRxPdu]		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00175 :**		
Name	IpduMContainedRxPduRef		
Parent Container	IpduMContainedRxPdu		
Description	Reference to the Pdu which represents this ContainedPdu and is used for reception indication.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

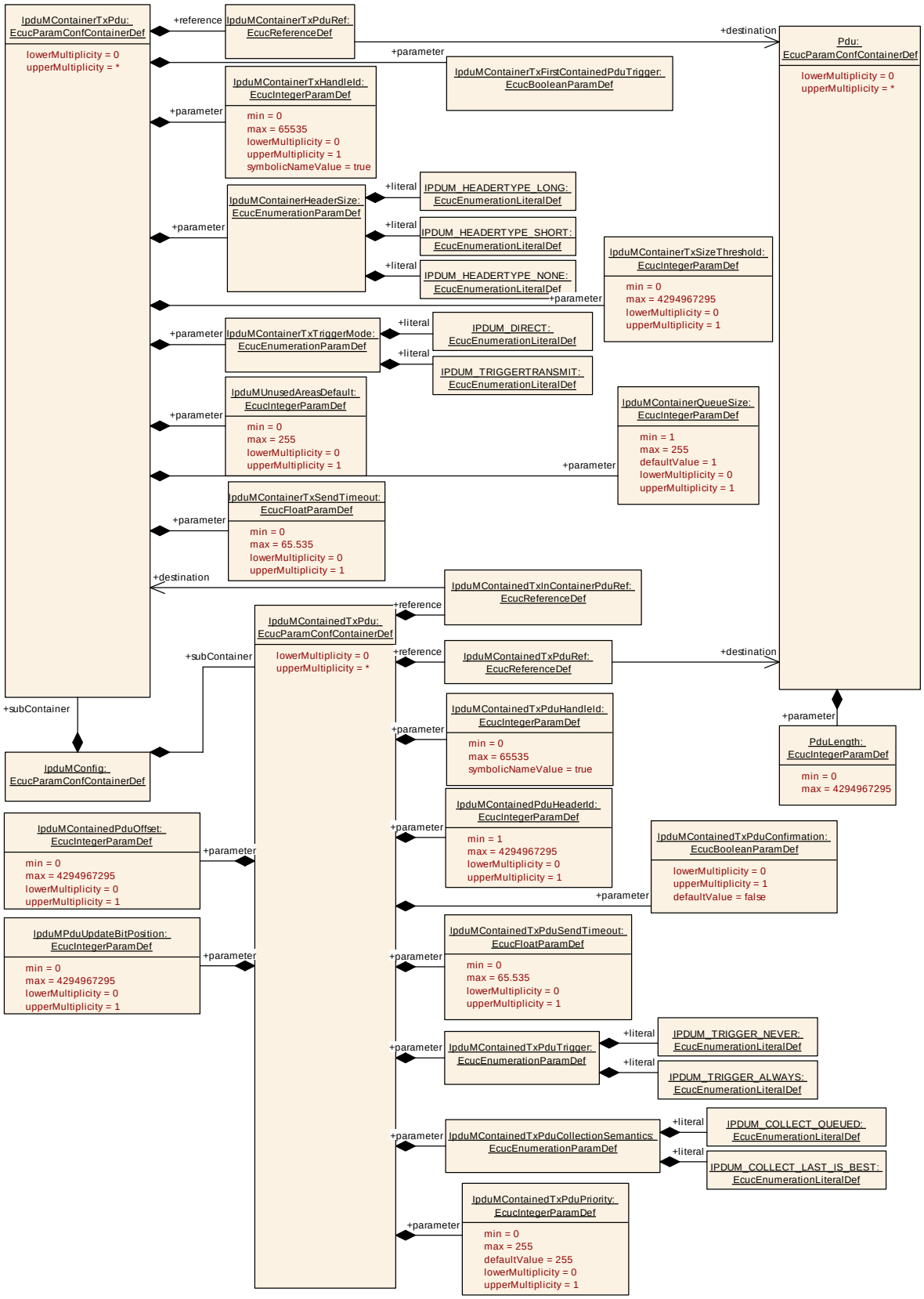

Figure 11 Configuration Overview TxContainer

10.2.20 IpduMContainerTxPdu

SWS Item	**ECUC_IpduM_00192 :**		
Container Name	IpduMContainerTxPdu		
Description	Configuration of a transmitted container Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00183 :**	
Name	IpduMContainerHeaderSize	
Parent Container	IpduMContainerTxPdu	
Description	Defines the layout of the header information (header id and length).	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	IPDUM_HEADERTYPE_LONG	Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit
	IPDUM_HEADERTYPE_NONE	Static Container Layout **Tags:** atp.Status=draft
	IPDUM_HEADERTYPE_SHORT	Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit
Post-Build Variant Value	true	
Value Configuration Class	Pre-compile time	X VARIANT-PRE-COMPILE
	Link time	X VARIANT-LINK-TIME
	Post-build time	X VARIANT-POST-BUILD
Scope / Dependency	scope: local	

SWS Item	**ECUC_IpduM_00185 :**		
Name	IpduMContainerQueueSize		
Parent Container	IpduMContainerTxPdu		
Description	Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default value	1		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00199 :**

Name	IpduMContainerTxFirstContainedPduTrigger		
Parent Container	IpduMContainerTxPdu		
Description	Defines if the transmission of this IpduMContainerTxPdu shall be requested right after the first IpduMContainedTxPdu was put into it.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00191 :**		
Name	IpduMContainerTxHandleId		
Parent Container	IpduMContainerTxPdu		
Description	Handle Id used by the PduR for TxConfirmation and for TriggerTransmit of the ContainerPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00194 :**		
Name	IpduMContainerTxSendTimeout		
Parent Container	IpduMContainerTxPdu		
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.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 65.535]		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00195 :**
Name	IpduMContainerTxSizeThreshold
Parent Container	IpduMContainerTxPdu

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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT or IPDUM_HEADERTYPE_LONG		

SWS Item	**ECUC_IpduM_00196 :**		
Name	IpduMContainerTxTriggerMode		
Parent Container	IpduMContainerTxPdu		
Description	Defines whether this ContainerPdu is fetched via trigger transmit.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_DIRECT	The IpduM sends this ContainerPdu when this ContainerPdu is triggered.	
	IPDUM_TRIGGERTRANSMIT	This ContainerPdu is stored in the IpduM and fetched via trigger transmit.	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00208 :**		
Name	IpduMUnusedAreasDefault		
Parent Container	IpduMContainerTxPdu		
Description	IpduM fills not updated areas of the Container PDU with this byte-pattern. **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: Only valid if IpduMContainerHeaderSize is set to		

	IPDUM_HEADERTYPE_NONE / should be aligned to bus-specific padding value if available.

SWS Item	**ECUC_IpduM_00193 :**		
Name	IpduMContainerTxPduRef		
Parent Container	IpduMContainerTxPdu		
Description	Reference to the Pdu which represents the container and is used for transmission.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.21 IpduMContainedTxPdu

SWS Item	**ECUC_IpduM_00177 :**		
Container Name	IpduMContainedTxPdu		
Description	Configuration of a sender ContainedPdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00172 :**		
Name	IpduMContainedPduHeaderId		
Parent Container	IpduMContainedTxPdu		
Description	Header Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT or IPDUM_HEADERTYPE_LONG.		

SWS Item	**ECUC_IpduM_00206 :**
Name	IpduMContainedPduOffset
Parent Container	IpduMContainedTxPdu
Description	Static offset (in bytes) of the ContainedPdu. **Tags:** atp.Status=draft

Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE. - only the ContainedPdu with the highest offset within a ContainerPdu may have variable length.		

SWS Item	**ECUC_IpduM_00198 :**		
Name	IpduMContainedTxPduCollectionSemantics		
Parent Container	IpduMContainedTxPdu		
Description	Defines whether this IpduMContainedTxPdu shall be collected using a last-is-best or queued semantics.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_COLLECT_LAST_IS_BEST		The IpduMContainedTxPdu data will be fetched via TriggerTransmit just before the transmission executes.
	IPDUM_COLLECT_QUEUED		The IpduMContainedTxPdu data will instantly be stored to the IpduMContainerTxPdu in the context of the Transmit API.
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00178 :**		
Name	IpduMContainedTxPduConfirmation		
Parent Container	IpduMContainedTxPdu		
Description	This Parameter determines whether for this contained I-PDU a TxConfirmation shall be provided. If set to TRUE a TxConfirmation is issued. It is not used when an I-PDU is requested using the trigger transmit API. If this Parameter is omitted, the default value shall be used.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00179 :**		
Name	IpduMContainedTxPduHandleId		
Parent Container	IpduMContainedTxPdu		
Description	Handle Id of the ContainedPdu.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00210 :**		
Name	IpduMContainedTxPduPriority		
Parent Container	IpduMContainedTxPdu		
Description	Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	255		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: The IpduMContainedTxPduPriority shall only be considered if IpduMContainedTxPduPriorityHandling is set to TRUE.		

SWS Item	**ECUC_IpduM_00181 :**		
Name	IpduMContainedTxPduSendTimeout		
Parent Container	IpduMContainedTxPdu		
Description	Defines a ContainedPdu specific sender timeout which can reduce the ContainerPdu timer when this ContainedPdu is put inside the Contai-nerPdu. Defined in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	[0 .. 65.535]		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00182 :**		
Name	IpduMContainedTxPduTrigger		
Parent Container	IpduMContainedTxPdu		
Description	Defines whether this Pdu triggers the sending of the ContainerPdu.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_TRIGGER_ALWAYS	This Pdu directly triggers the sending of the ContainerPdu.	
	IPDUM_TRIGGER_NEVER	This Pdu does not triggers the sending of the ContainerPdu (other trigger criteria might still trigger sending of the ContainerPdu).	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00207 :**		
Name	IpduMPduUpdateBitPosition		
Parent Container	IpduMContainedTxPdu		
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). **Tags:** atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE.		

SWS Item	**ECUC_IpduM_00176 :**		
Name	IpduMContainedTxInContainerPduRef		
Parent Container	IpduMContainedTxPdu		
Description	Reference to the container Pdu which this contained Pdu shall be collected in.		
Multiplicity	1		
Type	Reference to [IpduMContainerTxPdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00180 :**
Name	IpduMContainedTxPduRef

Parent Container	IpduMContainedTxPdu		
Description	Reference to the Pdu which represents this ContainedPdu and is used for transmission.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.3 Published Information

For details refer to the Chapter 10.3 Published Information in SWS_BSWGeneral.

10.3.1 IpduMPublishedInformation

SWS Item	**ECUC_IpduM_00141 :**
Container Name	IpduMPublishedInformation
Description	Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00142 :**		
Name	IpduMRxDirectComInvocation		
Parent Container	IpduMPublishedInformation		
Description	If set to TRUE the COM invocation optimization as defined in IPDUM140 is implemented.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Published Information	X	All Variants
Scope / Dependency	scope: local		

No Included Containers

10.4 Configuration Rules

10.4.1 Selector Field

[SWS_IpduM_00011] ⌈The number of values used of the selector field, i.e. values used to distinguish between different I-PDU layouts, does not have to be the whole range of possible values.⌋ (SRS_IpduM_02803)

Example: The size of a selector field with 3 bits leads to 2^3 possible selector field values; it shall be allowed to use only an arbitrary subset of these values. The used subset needs no to be contiguous.

10.4.2 Placement of static and dynamic parts

[SWS_IpduM_00224] ⌈All static and dynamic parts shall be configured to allocate exactly the same bits in the multiplexed and the de-multiplexed I-PDUs.⌋ (SRS_IpduM_02816, SRS_IpduM_02817)

The above requirement assures that the IpduM does not need to shift or swap any bits or bytes but is able to construct the (de-) multiplexed I-PDUs by masking and copying operations. The IpduM handles all segments as opaque data and performs no endianness or sign conversion of the copied data.

10.4.3 Multiple PDU to Container Mapping

[SWS_IpduM_00219] ⌈IpduM shall reject configurations in which the transmit properties (see ECUC_IpduM_00198: IpduMContainedTxPduCollectionSemantics) of the contained I-PDUs which are assigned to a specific Container PDU are mixed. A Container PDU shall contain either solely I-PDUs with IPDUM_COLLECT_LAST_IS_BEST or solely I-PDUs with IPDUM_COLLECT_QUEUED semantic.⌋ (SRS_IpduM_02821)

Container PDUs that have only I-PDUs assigned with IPDUM_COLLECT_LAST_IS_BEST semantic to can be realized buffer efficiently.

[SWS_IpduM_00230] ⌈IpduM shall reject configurations in which contained I-PDU supporting MetaData have a different MetaDataType from the MetaDataType of the Container PDU.⌋
(SRS_IpduM_02820)

The above requirement implies that multiple contained I-PDUs supporting MetaData assigned to the same Container PDU have the same MetaDataType.

DRAFT: [SWS_IpduM_00238] ⌈For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, all contained I-PDUs shall have IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST.⌋ (SRS_IpduM_02825)

DRAFT: [SWS_IpduM_00241] ⌈For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, all contained I-PDUs shall have a configured IpduMContainedTxPduOffset.⌋ (SRS_IpduM_02825)

DRAFT: [SWS_IpduM_00242] ⌈For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE and IpduMUnusedAreasDefault not set, all contained I-PDUs shall have a configured IpduMPduUpdateBitPosition.⌋ (SRS_IpduM_02825)

DRAFT: [SWS_IpduM_00240] ⌈Contained I-PDUs with a configured IpduMPduUpdateBitPosition shall only be assigned to Container PDUs with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE.⌋ (SRS_IpduM_02825)

DRAFT: [SWS_IpduM_00246] ⌈Only the last contained IPdu (according to IpduMContainedPduOffset) of a ContainerIPdu with static container layout (i.e. IpduMContainerHeaderSize set to *IPDUM_HEADERTYPE_NONE*) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.⌋ (SRS_IpduM_02825)

This constraint is in line with similar constraints in Com (SWS_Com_00754 and SWS_Com_00755) and in the FrIf (SWS_FrIf_05092).

DRAFT: [SWS_IpduM_00245] ⌈All IpduMPduUpdateBitPositions shall be configured to their own not otherwise occupied bit position.⌋ (SRS_IpduM_02825)

10.4.4 Priority

[SWS_IpduM_00248] ⌈ All contained I-Pdus with Collection Semantic IPDUM_COLLECT_LAST_IS_BEST and IpduMContainedTxPduPriorityHandling is set to TRUE, shall have an IpduMContainedTxPduPriority. If the IpduMContainedTxPduPriority is not configured, the IpduMContainedTxPduPriority shall be set to default value 255 (lowest available priority).⌋ (SRS_IpduM_02823)

11 Not applicable requirements

[SWS_IpduM_00999] [These requirements are not applicable to this specification.] (SRS_BSW_00171, SRS_BSW_00375, SRS_BSW_00437, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00427, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00336, SRS_BSW_00339, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00386, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00314, SRS_BSW_00377)