Document Title	Specification of Intrusion Detection System Manager for Adaptive Platform
Document Category	Software specification (SWS)
Document UID	978
Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Support of APMC configuration • Minor improvements and clarifications
2025-11-27	R25-11	AUTOSAR Release Management	• Minor improvements and clarifications
2024-11-27	R24-11	AUTOSAR Release Management	• Introduce QualifiedEventsReceiver and ReportingModeProvider. • Support versioning of context data. • Update document structure.
2023-11-23	R23-11	AUTOSAR Release Management	• Introduce ContextDataProvider. • Introduce TimestampProvider. • Clarifications regarding event ordering, interaction with DM, and the relationship between PortPrototype and SecurityEventType.
2022-11-24	R22-11	AUTOSAR Release Management	• No content changes
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• 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 7
2 Acronyms and Abbreviations 8
3 Related documentation 9
3.1 Input documents & related standards and norms . . . 9
3.2 Further Applicable Specification . . . 10
4 Constraints and assumptions 11
4.1 Known limitations . . . 11
5 Dependencies to other Functional Clusters 12
5.1 Provided Interfaces . . . 12
5.2 Required Interfaces . . . 13
5.3 Protocol layer dependencies . . . 14
6 Requirements Tracing 15
7 Functional specification 16
7.1 Event Generation . . . 16
7.1.1 Context Data Endianess . . . 17
7.2 Reporting Mode . . . 17
7.2.1 Reconfiguration of Reporting Mode . . . 17
7.3 Context Data Modification . . . 18
7.4 Filter Chain . . . 18
7.4.1 Function Group State Filter . . . 19
7.4.2 Sampling Filter . . . 19
7.4.3 Aggregation Filter . . . 19
7.4.4 Threshold Filter . . . 21
7.4.5 Qualification . . . 21
7.5 Timestamp . . . 21
7.6 Propagation of QSEvs . . . 22
7.7 Propagation of QSEvs to an Application . . . 23
7.8 Authenticity of Transmitted QSevs . . . 23
7.9 Rate & Traffic Limitation . . . 24
7.10 Access Control . . . 24
7.11 Diagnostic Access . . . 25
7.11.1 Access to Persisted Events . . . 25
7.12 IdsM Provided SEvs . . . 25
7.13 Functional cluster life-cycle . . . 26
7.13.1 Startup . . . 26
7.13.2 Shutdown . . . 26
7.13.3 Daemon crash . . . 27
7.14 Reporting . . . 27

7.14.1 Security Events . . . 27
7.14.2 Log Messages . . . 27
7.14.3 Violation Messages . . . 28
7.14.4 Production Errors . . . 29
8 API specification 30
8.1 PortInterface to API class binding . . . 30
8.2 Header: ara/idsm/common.h . . . 31
8.2.1 Non-Member Types . . . 31
8.3 Header: ara/idsm/context_data_provider.h . . . 33
8.3.1 Class: ContextDataProvider . . . 33
8.3.1.1 Public Member Functions . . . 33
8.3.1.1.1 Special Member Functions . . . 33
8.3.1.1.2 Constructors . . . 35
8.3.1.1.3 Member Functions . . . 35
8.4 Header: ara/idsm/event_reporter.h . . . 37
8.4.1 Class: EventReporter . . . 37
8.4.1.1 Public Member Functions . . . 37
8.4.1.1.1 Constructors . . . 37
8.4.1.1.2 Member Functions . . . 38
8.5 Header: ara/idsm/idsm_error_domain.h . . . 40
8.5.1 Non-Member Types . . . 40
8.5.2 Non-Member Functions . . . 41
8.5.2.1 Other . . . 41
8.5.3 Class: IdsmErrorDomain . . . 42
8.5.3.1 Public Member Types . . . 42
8.5.3.2 Public Member Functions . . . 43
8.5.3.2.1 Special Member Functions . . . 43
8.5.3.2.2 Member Functions . . . 43
8.5.4 Class: IdsmException . . . 44
8.5.4.1 Public Member Functions . . . 45
8.5.4.1.1 Constructors . . . 45
8.6 Header: ara/idsm/qualified_events_receiver.h . . . 45
8.6.1 Class: QualifiedEventsReceiver . . . 45
8.6.1.1 Public Member Functions . . . 46
8.6.1.1.1 Special Member Functions . . . 46
8.6.1.1.2 Constructors . . . 47
8.6.1.1.3 Member Functions . . . 48
8.7 Header: ara/idsm/reporting_mode_provider.h . . . 49
8.7.1 Class: ReportingModeProvider . . . 49
8.7.1.1 Public Member Functions . . . 50
8.7.1.1.1 Constructors . . . 50
8.7.1.1.2 Member Functions . . . 50
8.8 Header: ara/idsm/timestamp_provider.h . . . 51

8.8.1 Class: TimestampProvider . . . 51
8.8.1.1 Public Member Functions . . . 52
8.8.1.1.1 Special Member Functions . . . 52
8.8.1.1.2 Constructors . . . 53
8.8.1.1.3 Member Functions . . . 54
9 Service Interfaces 55
10 Configuration 56
10.1 Containers and configuration parameters . . . 56
10.1.1 IdsM Instantiation . . . 56
10.1.1.1 IdsM Instantiation . . . 56
10.1.1.2 IdsmGeneral . . . 58
10.1.1.3 IdsTimeBaseRefCollection . . . 61
10.1.2 IdsM Context Provider Mapping . . . 61
10.1.2.1 IdsmContextProviderMappingCollection . . . 61
10.1.2.2 IdsmContextProviderMapping . . . 62
10.1.3 IdsM Security Event Mapping . . . 63
10.1.3.1 IdsmSecurityEventMappingCollection . . . 63
10.1.3.2 IdsmSecurityEventMapping . . . 64
10.1.4 IdsM Timestamp Provider Mapping . . . 65
10.1.4.1 IdsmTimestampProviderMappingCollection . . . 65
10.1.4.2 IdsmTimestampProviderMapping . . . 66
10.1.5 IdsM Reporting Mode Provider Mapping . . . 67
10.1.5.1 IdsmReportingModeProviderMappingCollection . . . 67
10.1.5.2 IdsmReportingModeProviderMapping . . . 68
10.1.6 IdsM Qualified Event Receiver Mapping . . . 68
10.1.6.1 IdsmQualifiedEventReceiverMappingCollection . . . 68
10.1.6.2 IdsmQualifiedEventReceiverMapping . . . 69
10.1.7 Correspondence Table . . . 70
10.2 Default Values . . . 71
10.3 Semantic Constraints . . . 71
A Mentioned Manifest Elements 72
B Mentioned APMC Elements from other Functional Clusters 85
C Demands and constraints on Base Software (normative) 86
D Platform Extension Interfaces (normative) 87
E Not implemented requirements 88
F History of Constraints and Specification Items 89
F.1 Constraint and Specification Item History of this document according to AUTOSAR Release R22-11 . . . 89
F.1.1 Added Specification Items in R22-11 . . . 89
F.1.2 Changed Specification Items in R22-11 . . . 89

F.1.3 Deleted Specification Items in R22-11 . . . 89
F.2 Constraint and Specification Item History of this document according to AUTOSAR Release R23-11 . . . 89
F.2.1 Added Specification Items in R23-11 . . . 89
F.2.2 Changed Specification Items in R23-11 . . . 89
F.2.3 Deleted Specification Items in R23-11 . . . 90
F.3 Constraint and Specification Item History of this document according to AUTOSAR Release R24-11 . . . 90
F.3.1 Added Specification Items in R24-11 . . . 90
F.3.2 Changed Specification Items in R24-11 . . . 90
F.3.3 Deleted Specification Items in R24-11 . . . 90
F.3.4 Added Constraints in R24-11 . . . 91
F.3.5 Changed Constraints in R24-11 . . . 91
F.3.6 Deleted Constraints in R24-11 . . . 91
F.4 Constraint and Specification Item History of this document according to AUTOSAR Release R25-11 . . . 91
F.4.1 Added Specification Items in R25-11 . . . 91
F.4.2 Changed Specification Items in R25-11 . . . 91
F.4.3 Deleted Specification Items in R25-11 . . . 91
F.4.4 Added Constraints in R25-11 . . . 91
F.4.5 Changed Constraints in R25-11 . . . 91
F.4.6 Deleted Constraints in R25-11 . . . 92
F.5 Constraint and Specification Item History of this document according to AUTOSAR Release R26-11 . . . 92
F.5.1 Added Specification Items in R26-11 . . . 92
F.5.2 Changed Specification Items in R26-11 . . . 92
F.5.3 Deleted Specification Items in R26-11 . . . 92
F.5.4 Added Constraints in R26-11 . . . 92
F.5.5 Changed Constraints in R26-11 . . . 92
F.5.6 Deleted Constraints in R26-11 . . . 93

1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the AUTOSAR Adaptive Functional Cluster IdsM.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations that are only relevant to Intrusion Detection System Manager. A general list of acronyms and abbreviations is available in the [1, AUTOSAR glossary].

Acronym	Description:
Filter Chain	A set of consecutive filters which is applied to Security Events-
Intrusion Detection System	An Intrusion Detection System is a security control which detects and processes security events.
Intrusion Detection System Manager	The Intrusion Detection System Manager handles security events reported by security sensors.
Intrusion Detection System Reporter	The Intrusion Detection System Reporter handles qualified security events received from Idsm instances.
Security Extract	The Security Extract specifies which security events are handled by IdsM instances and their configuration parameters.
Security Event Type	A security event type can be identified by its security event type ID. Instances of security event types are called security events and share the same security event type ID.
Security Events	Onboard Security Events are instances of security event types which are reported by BSW or SWC to the IdsM.
Security Event Memory	A user defined diagnostic event memory which is independent from the primary diagnostic event memory.
Security Sensors	BSW or SWC which report security events to the Idsm.
Qualified Security Events	Security events which pass their filter chain are regarded as Qualified Security Events.
Security Incident and Event Management	Process for handling a confirmed security incident
Security Operation Centre	Organization of security and domain experts who are analyzing security events and contributing to mitigation of threats.
DID	Data Identifier according to Unified Diagnostic Services
DTC	Diagnostics Trouble Code
FC	Functional Cluster
IDS	Intrusion Detection System
IdsM	Intrusion Detection System Manager
IdsR	Intrusion Detection System Reporter
SecXT	Security Extract
SEv	Security Event
QSEv	Qualified Security Event
Sem	Security Event Memory
SIEM	Security Incident and Event Management
SOC	Security Operation Centre
SWCL	Software Cluster

Table 2.1: Acronyms and abbreviations used in the scope of this Document

3 Related documentation

This document is part of the AUTOSAR IDS specification and covers the software specification for the `Adaptive Platform`. For other aspects of the IDS specification, please refer to the following documents:

- **Protocol Requirements on transmission of qualified security events (PRS IDS) [2]:** Specifies the communication protocol between for the transmission of security events.
- **Security Extract Template [3]:** Specifies the Security Extract.

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_FO_TR_Glossary

[2] Specification of Intrusion Detection System Protocol
AUTOSAR_FO_PRS_IntrusionDetectionSystem

[3] Security Extract Template
AUTOSAR_FO_TPS_SecurityExtractTemplate

[4] General Specification of Adaptive Platform
AUTOSAR_AP_SWS_General

[5] Explanation of Adaptive Platform Software Architecture
AUTOSAR_AP_EXP_SWArchitecture

[6] General Requirements specific to Adaptive Platform
AUTOSAR_AP_RS_General

[7] Technical Report on Security Events Specification
AUTOSAR_FO_TR_SecurityEventsSpecification

[8] Specification of Cryptography
AUTOSAR_AP_SWS_Cryptography

[9] Specification of Adaptive Platform Core
AUTOSAR_AP_SWS_Core

[10] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

[11] Specification of Execution Management
AUTOSAR_AP_SWS_ExecutionManagement

3.2 Further Applicable Specification

AUTOSAR provides a general specification [4] which is also applicable for this functional cluster. It shall be considered an additional and required specification for implementing this functional cluster.

4 Constraints and assumptions

There are no known constraints and assumptions.

4.1 Known limitations

There are no known limitations for this specification.

5 Dependencies to other Functional Clusters

This chapter provides an overview of the dependencies to other Functional Clusters in the AUTOSAR Adaptive Platform. Section 5.1 “Provided Interfaces” lists the interfaces provided by Intrusion Detection System Mananger to other Functional Clusters. Section 5.2 “Required Interfaces” lists the interfaces required by Intrusion Detection System Mananger.

A detailed technical architecture documentation of the AUTOSAR Adaptive Platform is provided in [5].

5.1 Provided Interfaces

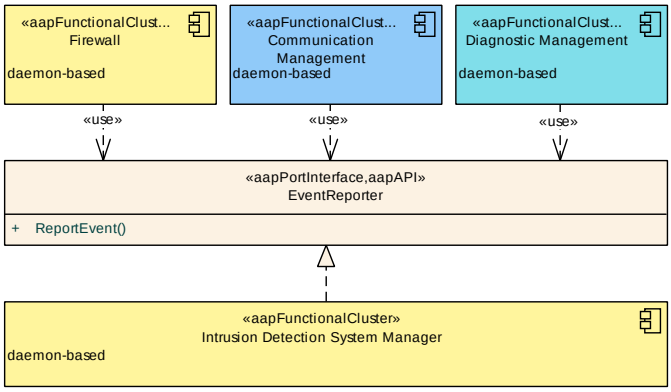


Figure 5.1: Interfaces provided by Intrusion Detection System Mananger to other Functional Clusters

Figure 5.1 shows interfaces provided by Intrusion Detection System Mananger to other Functional Clusters within the AUTOSAR Adaptive Platform. Table 5.1 provides a complete list of interfaces provided to other Functional Clusters within the AUTOSAR Adaptive Platform.

Interface	*Functional Cluster*	*Purpose*
`EventReporter`	Communication Management	`Communication Management` may use this interface to report security events.
	Cryptography	This interface is used to report standardized security events.
	Diagnostic Management	`Diagnostic Management` uses this interface to report standardized security events.
	Execution Management	`Execution Management` shall use this interface to report security events.
	Firewall	The `Firewall` uses this interface to report standardized security events.
	Platform Health Management	`Platform Health Management` uses this interface to report security events.

△

Interface	*Functional Cluster*	*Purpose*
	State Management	`State Management` shall use this interface to report security events.
	Time Synchronization	`Time Synchronization` uses this interface to report security events.
	Update and Configuration Management	This interface is used to report security events.

Table 5.1: Interfaces provided to other Functional Clusters

5.2 Required Interfaces

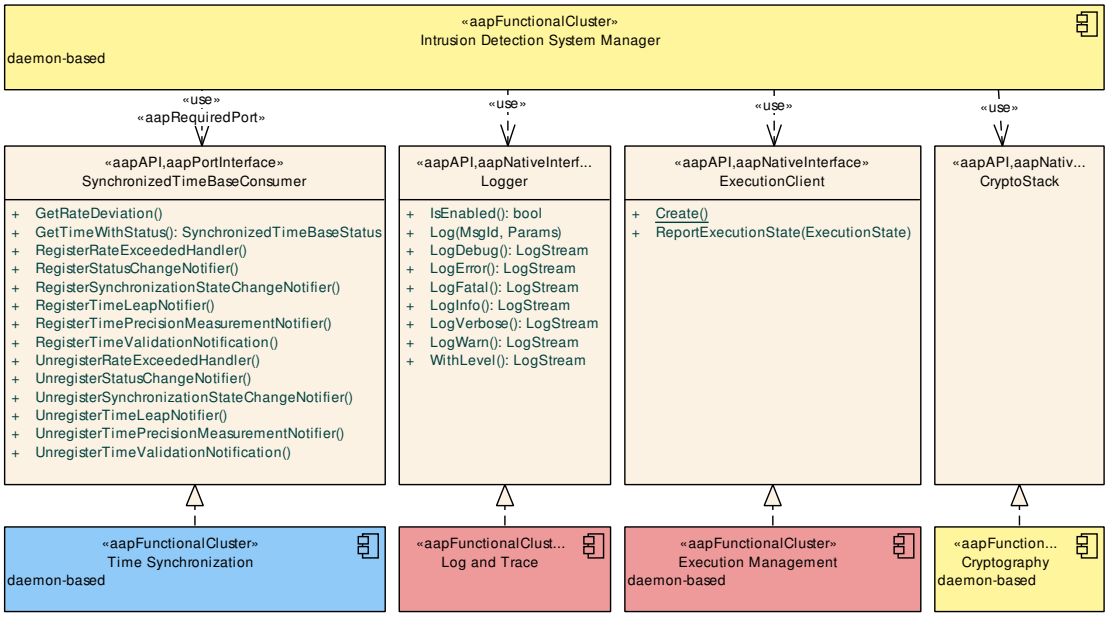


Figure 5.2: Interfaces required by Intrusion Detection System Mananger from other Functional Clusters

Figure 5.2 shows interfaces required by Intrusion Detection System Mananger from other Functional Clusters within the AUTOSAR Adaptive Platform. Table 5.2 provides a complete list of required interfaces from other Functional Clusters within the AUTOSAR Adaptive Platform.

Functional Cluster	*Interface*	*Purpose*
Cryptography	CryptoStack	`Adaptive Intrusion Detection System Manager` uses this interface to sign security events.
Execution Management	ExecutionClient	
Log and Trace	Logger	`Adaptive Intrusion Detection System Manager` shall use this interface to log standardized messages.

▽

△

Functional Cluster	*Interface*	*Purpose*
Time Synchronization	SynchronizedTimeBaseConsumer	`Adaptive Intrusion Detection System Manager` shall use this interface to determine timestamps of security events.

Table 5.2: Interfaces required from other Functional Clusters

5.3 Protocol layer dependencies

Security events generated via the `IdsM` API can be transmitted to the `IdsR` using the protocol specified in PRS IDS [2].

6 Requirements Tracing

The following tables reference the requirements specified in AUTOSAR RS General [6] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_AP_00120]	Method and Function names	[SWS_AIDSM_10702] [SWS_AIDSM_10703] [SWS_AIDSM_10705] [SWS_AIDSM_10707] [SWS_AIDSM_10708] [SWS_AIDSM_10709] [SWS_AIDSM_10710] [SWS_AIDSM_10711] [SWS_AIDSM_10712]
[RS_AP_00121]	Parameter names	[SWS_AIDSM_10703] [SWS_AIDSM_10705] [SWS_AIDSM_10711] [SWS_AIDSM_10712]
[RS_AP_00122]	Type names	[SWS_AIDSM_10204] [SWS_AIDSM_10704] [SWS_AIDSM_10706]
[RS_AP_00127]	Usage of ara::core types	[SWS_AIDSM_10204] [SWS_AIDSM_10704] [SWS_AIDSM_10706]
[RS_AP_00130]	AUTOSAR Adaptive Platform shall represent a rich and modern programming environment	[SWS_AIDSM_10204] [SWS_AIDSM_10702] [SWS_AIDSM_10703] [SWS_AIDSM_10704] [SWS_AIDSM_10705] [SWS_AIDSM_10706] [SWS_AIDSM_10707] [SWS_AIDSM_10708] [SWS_AIDSM_10709] [SWS_AIDSM_10710] [SWS_AIDSM_10711] [SWS_AIDSM_10712]
[RS_AP_00142]	Handling of unsuccessful operations	[SWS_AIDSM_20000] [SWS_AIDSM_20001]
[RS_AP_00173]	InstanceSpecifierAlreadyInUse Violation	[SWS_AIDSM_10401] [SWS_AIDSM_10501] [SWS_AIDSM_10601] [SWS_AIDSM_10801]
[RS_AP_00177]	AraNotInitializedViolation	[SWS_AIDSM_10301] [SWS_AIDSM_10401] [SWS_AIDSM_10501] [SWS_AIDSM_10601] [SWS_AIDSM_10801]
[RS_AP_00180]	UnresolvableInstanceSpecifier Violation	[SWS_AIDSM_10301] [SWS_AIDSM_10401] [SWS_AIDSM_10501] [SWS_AIDSM_10601] [SWS_AIDSM_10801]

Table 6.1: Requirements Tracing

7 Functional specification

This chapter specifies the functional behavior of the IdsM for the Adaptive Platform.

7.1 Event Generation

SWCLs and FCs can generate new security events using the IdsM API. All event types that can be generated by a SWCL are configured in the manifest and linked to a PortPrototype of the SWCL. Generating new events involves three steps:

1. Construct an InstanceSpecifier object using the shortName path of the PortPrototype referencing the event type as the parameter.
2. Construct an ara::idsm::EventReporter object by passing the InstanceSpecifier.
3. Call the ara::idsm::EventReporter::ReportEvent function on the ara::idsm::EventReporter object.

Using the ara::idsm::EventReporter::ReportEvent function, an application can optionally provide a timestamp, a counter, and/or context data.

[SWS_AIDSM_00102] EventReporter Constructor ⌈The constructor ara::idsm::EventReporter::EventReporter shall create an ara::idsm::EventReporter instance that reports security events defined by the SecurityEventDefinition that is referenced by the SecurityEventMapping (via the SecurityEventForeignRef) which also references

- the RPortPrototype (via the IdsmRPortPrototypeInExecutableInstanceRef) identified by the parameter instanceSpecifier and
- the calling Process (via the IdsmProcessForeignRef).

⌋

[SWS_AIDSM_00103] Event Reporting ⌈The functions ara::idsm::EventReporter::ReportEvent, ara::idsm::EventReporter::ReportEvent, ara::idsm::EventReporter::ReportEvent, ara::idsm::EventReporter::ReportEvent, ara::idsm::EventReporter::ReportEvent, and ara::idsm::EventReporter::ReportEvent shall trigger processing of the security event identified by the instanceSpecifier passed to the constructor of the ara::idsm::EventReporter object.⌋

7.1.1 Context Data Endianess

SEv context data is generally structured (see e.g. [7]) and can contain multiple elements. On the IdsM side, the internal structure of the context data is not visible, as the context data is provided by the IDS sensor as a byte array.

In order to to fulfill [PRS_Ids_00004], all context data elements need to be provided in big endian byte order. This needs to be considered when constructing the SEv context data, i.e., the IDS sensor needs to take care of the endianess before submitting the SEv to the IdsM.

7.2 Reporting Mode

[SWS_AIDSM_00201] Reporting Mode ⌈`IdsM` shall determine the default reporting mode of every reported `SEv` from the `SecXT` model parameter `SecurityEventContextProps.defaultReportingMode`.⌋

[SWS_AIDSM_00202] Reporting Mode Options ⌈

Reporting Mode Level	*Related Behavior*
OFF	`IdsM` shall discard the `SEv` without further processing.
BRIEF	If the `SEv` has been reported including context data, `IdsM` shall shall discard the context data from further processing, transmission, and storage.
DETAILED	If the `SEv` has been reported including context data, `IdsM` shall keep the context data for potential transmission or persisting of the `QSEv`.
BRIEF_BYPASSING_FILTERS	`IdsM` shall report or persist the `SEv` without context data without further application of any filter chain.
DETAILED_BYPASSING_FILTERS	`IdsM` shall report or persist the `SEv` with context data (if provided by the sensor) without further application of any filter chain.

⌋

7.2.1 Reconfiguration of Reporting Mode

[SWS_AIDSM_00203] Get current reporting mode ⌈The function `ara::idsm::ReportingModeProvider::GetReportingMode` shall provide the current reporting mode of the `SEv` specified by the parameter `eventId`.⌋

[SWS_AIDSM_00204] Set current reporting mode ⌈The function `ara::idsm::ReportingModeProvider::SetReportingMode` shall change the reporting mode of the `SEv` specified by the parameter `eventId` to the mode specified by the parameter `reportingMode`.⌋

7.3 Context Data Modification

[SWS_AIDSM_01501] Context Data Modification ⌈If `ContextProviderMapping` exists and an application registered a `ara::idsm::ContextDataProvider` via a call to `ara::idsm::ContextDataProvider::Offer`, then `IdsM` shall call the function `ara::idsm::ContextDataProvider::ModifyContextData` and use the modified context data for further processing of the SEv.⌋

[SWS_AIDSM_01503] Context Data Offset ⌈If IdsM calls the function `ara::idsm::ContextDataProvider::ModifyContextData`, the parameter contextData shall be a span that contains the original context data at the offset `originalContextDataOffset`, which has been set at the constructor of the `ara::idsm::ContextDataProvider`. The size of the span shall be the size of the original context data plus the parameter `additionalBytes`.⌋

7.4 Filter Chain

Filter chains are configured using the `SecXT` model element `SecurityEventFilterChain`.

[SWS_AIDSM_00301] Filter chain selection ⌈When a `SEv` is reported, the `IdsM` shall apply the filter chain that is mapped to the `SecurityEventDefinition` of the reported `SEv` via the `SecurityEventContextMapping`.⌋

[SWS_AIDSM_00302] Filter chain evaluation ⌈`IdsM` shall evaluate the filter chain after evaluating the reporting mode.⌋

[SWS_AIDSM_00303] Possible Filters ⌈Each filter chain may consist of the following filters:

- Function Group State Filter
- Forward-Every-nth Filter
- Aggregation Filter
- Threshold Filter

⌋

[SWS_AIDSM_00304] Filter chain configuration ⌈Each filter can be activated by aggregating the respective Filter object at the `SecurityEventFilterChain` object in the model.⌋

[SWS_AIDSM_00305] Filter chain order ⌈`IdsM` shall evaluate all activated filter in the order Function Group State Filter, Forward-Every-nth Filter, Aggregation Filter, Threshold Filter.⌋

[SWS_AIDSM_00306] Dropping of SEvs ⌈If the evaluation of one filter leads to dropping the SEv, IdsM shall not evaluate any additional filter.⌋

After successful evaluation of the configured filter chain, we define the security event as qualified (QSEv).

7.4.1 Function Group State Filter

[SWS_AIDSM_00401] Function Group State Filter ⌈If IdsM evaluates the Function Group State Filter and the current function group state equals one of the states referenced by SecurityEventStateFilter.blockIfStateActiveAp, then IdsM shall drop the SEv.⌋

Note: The Function Group State Filter is comparable to the Machine State filter specified for IdsM on Classic Platform.

7.4.2 Sampling Filter

[SWS_AIDSM_00501] Sampling Filter ⌈If IdsM evaluates the sampling filter for a SEv, IdsM shall drop all the SEvs but every *n-th* per SecurityEventDefinition, where n is defined by SecurityEventOneEveryNFilter.n.⌋

An implementation will typically maintain one counter per SecurityEventDefinition that will be incremented when an SEv of given type is evaluated by the sampling filter. If the counter equals n the SEv is not dropped and the counter is reset to 0.

[SWS_AIDSM_00502] Sampling Filter Initialization ⌈IdsM shall initialize the sampling filter for a SEv so that the first received SEv per SecurityEventDefinition is forwarded.⌋

Example: SecurityEventOneEveryNFilter.n is set to 3 for a certain event type, then SEvs 1, 4, 7, ... will be forwarded by the IdsM (1 describing the first SEv reported after reset).

7.4.3 Aggregation Filter

All SEv of a given type occurring within a configured time interval are aggregated into one SEv with an additional counter information attached that indicates how often the event occurred in the time interval.

[SWS_AIDSM_00600] Configuration of Aggregation Filter ⌈The integrator shall configure the parameter SecurityEventAggregationFilter.minimumInterval-

Length to be the duration of the interval during which SEvs of the given type shall be aggregated.⌋

[SWS_AIDSM_00608] Start of Intervals ⌈The intervals used for the Aggregation Filter and the Threshold Filter shall first start after initialization of IdsM. Subsequent intervals shall start with the end of the previous interval of the same filter.⌋

[SWS_AIDSM_00601] No Event Forwarding During Interval ⌈The aggregation filter shall not forward (i.e., to the next filter) any incoming SEv during the aggregation interval.⌋

At the end of each aggregation interval, the aggregation filter shall implement the following logic for each Security Event Type:

[SWS_AIDSM_00602] End of Interval: No Event ⌈If no SEv of the same event type has been received by the aggregation filter in the past aggregation interval, no action shall be taken.⌋

[SWS_AIDSM_00603] End of Interval: One or More Events ⌈If one or more SEv of the same event type have been received by the aggregation filter in the past aggregation interval, a SEv shall be forwarded to the next filter in the chain.⌋

[SWS_AIDSM_00604] End of Interval: Count ⌈If the SEv is forwarded to the next filter in the filter chain, the count parameter of the SEv shall equal the sum of all count parameters of all SEvs of given event type processed by the aggregation filter in the past time interval.⌋

[SWS_AIDSM_00605] End of Interval: First Context Data ⌈If the SEv is forwarded to the next filter in the filter chain and if SecurityEventAggregationFilter.contextDataSource equals IDSM_FILTERS_CTX_USE_FIRST, then the context data shall equal the first context data of an SEv of given type that has been received at the aggregation filter in the past time interval.⌋

[SWS_AIDSM_00606] End of Interval: Last Context Data ⌈If the SEv is forwarded to the next filter in the filter chain and if SecurityEventAggregationFilter.contextDataSource equals IDSM_FILTERS_CTX_USE_LAST, then the context data shall equal the last context data of an SEv of given type that has been received at the aggregation filter in the past time interval.⌋

[SWS_AIDSM_00607] End of Interval: Timestamp ⌈If the SEv is forwarded to the next filter in the filter chain, the timestamp shall be taken from the same SEv from which the context data comes from (configured via SecurityEventAggregationFilter.contextDataSource).⌋

7.4.4 Threshold Filter

[SWS_AIDSM_00701] Event Dropping Below Threshold ⌈The threshold filter shall drop an SEv of given type if the sum of count parameters of all SEvs of given type that were processed by the threshold filter in the current threshold interval is smaller than the configured parameter SecurityEventThresholdFilter.thresholdNumber.⌋

[SWS_AIDSM_00702] Event Forwarding Above Threshold ⌈The threshold filter shall forward an SEv of given type if the sum of count parameters of all SEvs of given type that were processed by the threshold filter in the current threshold interval is equal to or greater than the configured parameter SecurityEventThresholdFilter.thresholdNumber.⌋

7.4.5 Qualification

[SWS_AIDSM_00703] Event Forwarding Above Threshold ⌈Filter algorithms shall not cause the SEv count to overflow. If the maximum value is reached, the value shall not be incremented any further.⌋

After a SEv has successfully passed the last configured filter of the filter chain, it is considered a QSEv. Depending on the configuration, the QSEv can be transmitted to the IdsR and/or persisted locally.

7.5 Timestamp

Timestamps are optional and can be provided to the IdsM in different ways.

[SWS_AIDSM_00801] Timestamps are optional ⌈If IdsmInstance.timestampFormat is not set, IdsM shall not add a timestamp to a QSEv and shall ignore timestamps provided via the timestamp parameter of the event reporting interface.⌋

[SWS_AIDSM_00802] Timestamps provided by the stack ⌈If IdsmInstance.timestampFormat equals "'AUTOSAR"' and the ara::idsm::EventReporter::ReportEvent function is called without a timestamp parameter, then Idsm shall add a timestamp from the TimeSync::TimeBaseResource referenced via the IdsTimeBaseForeignRef to stored and transmitted QSEvs.⌋

The format of the timestamp to be added is specified in [2].

[SWS_AIDSM_00803] Timestamp provided via event reporting interface ⌈If IdsmInstance.timestampFormat is set and the ara::idsm::EventReporter::ReportEvent function is called with a timestamp parameter, then Idsm shall use this provided timestamp parameter for transmission or storage of the QSEv.⌋

[SWS_AIDSM_00804] Timestamp provided via application software ⌈If IdsmInstance.timestampFormat does not equal "AUTOSAR" and the ara::idsm::EventReporter::ReportEvent function is called without a timestamp parameter, then IdsM shall add a timestamp that is provided by a application software through the ara::idsm::TimestampProvider::GetTimestamp callback to the QSEv.⌋

[SWS_AIDSM_00805] Timestamp configured but not provided ⌈If IdsmInstance.timestampFormat does not equal "AUTOSAR", but the ara::idsm::EventReporter::ReportEvent function is called without a timestamp parameter and no TimestampProvider has been registered, then IdsM shall not add a timestamp to the QSEv.⌋

[SWS_AIDSM_00806] Truncation of timestamp parameter ⌈If the ara::idsm::EventReporter::ReportEvent function is called with a timestamp parameter, then IdsM shall truncate this value by the 2 most-significant bits, i.e., only keep the 62 least-significant bits for further use.⌋

It is possible that the report event function is called in an order that does not match with the timestamp provided, i.e., the later call contains an older timestamp. This means that the persisted and transmitted events may contain timestamps that are not necessarily ordered.

7.6 Propagation of QSEvs

[SWS_AIDSM_00901] QSEv transmission ⌈If a PlatformModuleEthernetEndpointConfiguration is referenced from the IdsPlatformInstantiation.idsDesign in the role networkInterface, IdsM shall transmit QSEvs using the IDS protocol defined in [2] from the local endpoint configured via the PlatformModuleEthernetEndpointConfiguration referenced by the IdsFunctionalClusterDesign in the role networkInterface to the remote endpoint configured via the RemoteEndpointConfiguration referenced by the PlatformModuleEthernetEndpointConfiguration in the role remoteConfig.⌋

[SWS_AIDSM_00902] Message ID ⌈IdsM shall set the Message ID field of the IDS Message Separation Header to all zero (0x00000000).⌋

[SWS_AIDSM_00904] Protocol Version ⌈If unversioned context data has been provided to IdsM, IdsM shall use version 1 of the IDS network protocol defined in [2]. Otherwise, IdsM shall use version 2 of the IDS network protocol.⌋

The Module Instance ID is used on CP to distinguish between events raised by multiple instances of the same BSW component. On AP, there can only be one instance of a functional cluster. Therefore, the Module Instance ID is not used on AP:

[SWS_AIDSM_00903] IdsM Module Instance ID ⌈IdsM shall set the Module Instance ID field of the IDS Header to all zero.⌋

In case that distinguishing different processes is required for a certain security event, identifying information is placed in the event's context data.

7.7 Propagation of QSEvs to an Application

In addition to QSEvs being propagated via the IDS network protocol, QSEvs can also be propagated to an application. Both propagation mechanisms are independent, e.g., the same QSEv may be propagated via the network protocol and to an application.

[SWS_AIDSM_01601] QSEv transmission to application ⌈If a `QualifiedEventReceiverMapping` exists and an application registered a `ara::idsm::QualifiedEventsReceiver` via a call to `ara::idsm::QualifiedEventsReceiver::Offer`, then `IdsM` shall call the function `ara::idsm::QualifiedEventsReceiver::OnEventQualification` upon qualification of a QSEv with unversioned context data, or the function `ara::idsm::QualifiedEventsReceiver::OnEventQualification` upon qualification of a QSEv with versioned context data or without context data.⌋

7.8 Authenticity of Transmitted QSevs

`IdsM` can optionally protect the authenticity of transmitted `QSEv`s using cryptographic signatures or message authentication codes.

[SWS_AIDSM_01001] Authenticating QSEv ⌈If the container `IdsmAuthenticationSupport` exists, then `IdsM` shall calculate a cryptographic authenticator for each `QSEv` transmitted to the IdsR and to each locally persisted `QSEv`.⌋

[SWS_AIDSM_01003] Authentication Success and Failure ⌈If authenticator generation according to [SWS_AIDSM_01001] succeeds, IdsM shall attach the generated authenticator to each QSEv. If authenticator generation fails, IdsM shall continue processing the QSEv without a authenticator.⌋

[SWS_AIDSM_01004] Authenticator Generation Failed ⌈When the IdsM fails to calculate the authenticator or authenticator generation failure occurs for other reasons, IdsM shall log the DltMessage AuthenticatorGenerationFailed.⌋

Over which data the authenticator shall be computed and how the authenticator shall be included in the message transmitted to the `IdsR` is specified in [2]. Which authenticator primitive and which key shall be used can be configured using the parameters of `IdsmAuthenticationSupport`:

[SWS_AIDSM_01002] Primitive, Key and Format ⌈`IdsM` shall determine the authentication algorithm, the key, and the authentication format from the following configuration:

1. Use the authentication algorithm specified in the parameter AuthenticationPrimitive.
2. Use the authentication key identified by the CryptoKeySlot that is referenced by the AuthenticationKeySlotRef.
3. Use the output format specified by the parameter AuthenticationFormat.

⌋

The naming scheme for the authentication algorithm to be used is specified in SWS Cryptography [8].

7.9 Rate & Traffic Limitation

[SWS_AIDSM_01101] Rate and Traffic Limitation ⌈Before sending a QSEv to the IdsR, IdsM shall apply rate and traffic limitation that can lead to dropping the QSEv.⌋

[SWS_AIDSM_01103] Rate Limitation ⌈IdsM shall drop an QSEv from transmission, if its transmission would cause the number of QSEvs transmitted in the current interval, which is specified in IdsmRateLimitation.timeInterval, to exceed the maximum number of transmission configured as IdsmRateLimitation.maxEventsInInterval.⌋

[SWS_AIDSM_01104] Traffic Limitation ⌈IdsM shall drop an QSEv from transmission, if its transmission would cause the number of bytes transmitted in the current interval, which is specified in IdsmTrafficLimitation.timeInterval, to exceed the maximum number of bytes configured as IdsmTrafficLimitation.maxBytesInInterval.⌋

7.10 Access Control

The generation of security events, modification of context data, and provision of timestamps is subject to access control, i.e., it can be restricted which processes can perform these tasks.

[SWS_AIDSM_01201] EventReporter Access Control ⌈IdsM shall restrict the Processes that can report an event type identified by a SecurityEventDefinition, which is referenced by a SecurityEventMapping, to the Process referenced by the same SecurityEventMapping via the IdsmProcessForeignRef.⌋

[SWS_AIDSM_01202] TimestampProvider Access Control ⌈IdsM shall restrict the Processes that can provide timestamps via the TimestampProvider interface to the Process referenced by an TimestampProviderMapping via the IdsmProcessForeignRef.⌋

[SWS_AIDSM_01203] ContextDataProvider Access Control ⌈IdsM shall restrict the Processes that can modify context data via the ContextDataProvider interface to the Process referenced by an ContextProviderMapping via the IdsmProcessForeignRef.⌋

[SWS_AIDSM_01204] ReportingModeProvider Access Control ⌈IdsM shall restrict the Processes that can modify and query the reporting mode via the ReportingModeProvider interface to the Process ReportingModeProviderMapping via the IdsmProcessForeignRef.⌋

[SWS_AIDSM_01205] QualifiedEventsReceiver Access Control ⌈IdsM shall restrict the Processes that can receive qualified security events via the QualifiedEventsReceiver interface to the Process referenced by an QualifiedEventReceiverMapping via the IdsmProcessForeignRef.⌋

[SWS_AIDSM_01206] Access Control Security Event ⌈If access to resources was not granted by IDSM due to [SWS_AIDSM_01201], [SWS_AIDSM_01202], [SWS_AIDSM_01203], [SWS_AIDSM_01204], or [SWS_AIDSM_01205], the Security Event SEV_ACCESS_CONTROL_IDSM_IAM_ACCESS_DENIED shall be raised.⌋

7.11 Diagnostic Access

IdsM allows diagnostic access to support two use-cases: First, persisted events can be read via diagnostic access. Second, a reconfiguration of the reporting mode via diagnostic access is possible.

7.11.1 Access to Persisted Events

Each security event references a diagnostic event, which in turn references a DTC.

[SWS_AIDSM_01301] Access to Persisted Events ⌈If a QSEv has been successfully qualified and the QSEv is configured to be persisted (i.e., SecurityEventContextProps.persistentStorage == True) and mapped to a DiagnosticEvent via DiagnosticEventToSecurityEventMapping, then IdsM shall report the status of the referenced DiagnosticEvent to kFailed and, if the ReportingMode is DETAILED or DETAILED_BYPASSING_FILTERS, additionally store the provided context data and timestamp in the DiagnosticEvent's snapshot record.⌋

7.12 IdsM Provided SEvs

IdsM itself can also be used as a Security Event sensor. The security events reported by the IdsM module are listed in Section 7.14.1.

[SWS_AIDSM_01402] Buffer availability ⌈`IdsM` shall ensure that `IdsM` internal events can be processed even though no buffers are available.⌋

An implementation could achieve this by, e.g., pre-allocating memory buffers for `IdsM` provided events.

[SWS_AIDSM_01403] Bypass limitation filter ⌈`IdsM` internal `SEv`s shall not be filtered by rate and traffic limitation filter.⌋

7.13 Functional cluster life-cycle

This section defines behavior of this functional cluster during its life-cycle. Please note that there is a general behavior for ara::core::Initialize and ara::core::Deinitialize defined in [9] by [SWS_CORE_15005] and [SWS_CORE_90022].

[SWS_AIDSM_00003] Queuing of IDS events ⌈IDS events that are reported while the IdsM is not able to process them (e.g. daemon communication is not established) shall be queued. The queue size is defined by `IdsmGeneral.IdsmQueueSize`. If this size is exceeded, the oldest entries shall be discarded. If `IdsmGeneral.IdsmQueueSize` is not defined, the queue size is zero.⌋

7.13.1 Startup

Using `ara::core::Initialize`, the application can initialize its ara::idsm library.

[SWS_AIDSM_00001] Initialization ⌈When `ara::core::Initialize` is called, IdsM shall read in the manifest information and prepare the access structures necessary to generate events from the application.⌋

Access structures may encompass the communication channel between the application process and the stack process (if there is any) or other resource required by the IdsM.

7.13.2 Shutdown

Using `ara::core::Deinitialize`, the application can deinitialize its ara::idsm library.

[SWS_AIDSM_00002] Deinitialization ⌈When `ara::core::Deinitialize` is called, the IdsM shall close all accquired handles and free all access structures.⌋

The application is expected not to call any API of `IdsM` before `ara::core::Initialize` or after `ara::core::Deinitialize`.

7.13.3 Daemon crash

This section is intentionally left empty.

7.14 Reporting

7.14.1 Security Events

This chapter contains all standardized Security Events of this Functional Cluster.

[SWS_AIDSM_01401] Security events for IDSM (AP) ⌈

Name	*Description*	*ID*
SEV_IDSM_NO_EVENT_BUFFER_AVAILABLE	A SEv cannot be handled because there are no more event buffers available to process the event.	46
SEV_IDSM_NO_CONTEXT_DATA_BUFFER_AVAILABLE	The context data of an incoming event cannot be stored because there are no more context data buffers available.	47
SEV_IDSM_COMMUNICATION_ERROR	An error occurred when sending a QSEv via PDU.	49
SEV_IDSM_NO_QUALIFIED_EVENT_BUFFER_AVAILABLE	A security event raised when a QSEv has to be dropped due to insufficient QSEv buffers available.	87
SEV_ACCESS_CONTROL_IDSM_IAM_ACCESS_DENIED	Access of an application to a resource provided by Intrusion Detection System Management was denied.	136

⌋

[SWS_AIDSM_02001] Security event context data definition: SEV_ACCESS_CONTROL_IDSM_IAM_ACCESS_DENIED ⌈

SEV Name	*SEV_ACCESS_CONTROL_IDSM_IAM_ACCESS_DENIED*	
ID	136	
Description	Access of an application to a resource provided by Intrusion Detection System Management was denied.	
Context Data Version	1	
Context Data	*Data Type*	*Allowed Values*
UserId	uint32	

⌋

7.14.2 Log Messages

This chapter contains all Log Messages (i.e. DLT messages) of this Functional Cluster.

[SWS_AIDSM_20001] Log Message AuthenticatorGenerationFailed

Status: DRAFT
Upstream requirements: RS_AP_00142

⌈

Dlt-Message	AuthenticatorGenerationFailed		
Description	When the AIDSM FC failed to calculate the authenticator.		
MessageId	0x80005000		
MessageType Info	DLT_LOG_WARN		
Dlt-Argument	***ArgumentDescription***	***ArgumentType***	***ArgumentUnit***
No Dlt-arguments available			

⌋

7.14.3 Violation Messages

This section lists all violation messages (i.e., DLT messages logged for Violations according to [AP_SWS_GENERAL_00021]) defined by this functional cluster.
Please note that concrete implementations might additionally implement Non-Standardized Violations (see also [AP_SWS_GENERAL_00003]).

[AP_SWS_GENERAL_13215] Violation Message UnresolvableInstanceSpecifier Violation

Dlt-Message	UnresolvableInstanceSpecifierViolation		
Description	InstanceSpecifier either cannot be resolved in the model in the context of your executable or process, or it refers to a model element that is neither a PortPrototype of the expected type nor a FunctionalCluster InteractsWithFunctionalClusterMapping of the expected type.		
MessageId	0x80001fe8		
MessageType Info	DLT_LOG_FATAL		
Dlt-Argument	***ArgumentDescription***	***ArgumentType***	***ArgumentUnit***
modeledProcessId	Meta-model identifier of the process that caused the violation, i.e., its short name path with '/' as a separator	uint8 [encoding UTF-8]	
location	An implementation-defined identifier of the location where the violation was detected, for example {Class Name}:{MethodName}, {filename}:{linenumber}, etc.	uint8 [encoding UTF-8]	
instanceSpecifier	InstanceSpecifier used to try to create the object.	uint8 [encoding UTF-8]	
className	Name of the class that was instantiated.	uint8 [encoding UTF-8]	

[AP_SWS_GENERAL_13006] Violation Message InstanceSpecifierAlreadyInUse Violation

Dlt-Message	InstanceSpecifierAlreadyInUseViolation
Description	Violation message that is sent in case a constructor in the ara framework was called with an Instance Specifier already in use in this process.

▽

△

MessageId	0x80001ff9		
MessageType Info	DLT_LOG_FATAL		
Dlt-Argument	***ArgumentDescription***	***ArgumentType***	***ArgumentUnit***
modeledProcessId	Meta-model identifier of the process that caused the violation, i.e., its short name path with '/' as a separator.	uint8 [encoding UTF-8]	
location	An implementation-defined identifier of the location where the violation was detected, for example {filename}:{linenumber}.	uint8 [encoding UTF-8]	
instanceSpecifier	InstanceSpecifier used to try to create the object.	uint8 [encoding UTF-8]	
className	Name of the class that was instantiated.	uint8 [encoding UTF-8]	

[SWS_AIDSM_20000] Violation Message InvalidOriginalContextDataOffsetViolation

Status: DRAFT
Upstream requirements: RS_AP_00142

⌈

Dlt-Message	InvalidOriginalContextDataOffsetViolation		
Description	Violation message that is sent in case the parameter originalContextDataOffset is larger than the parameter additionalBytes in the constructor of the class ContextDataProvider.		
MessageId	0x80005fff		
MessageType Info	DLT_LOG_FATAL		
Dlt-Argument	***ArgumentDescription***	***ArgumentType***	***ArgumentUnit***
modeledProcessId	Meta-model identifier of the process that caused the violation, i.e., its short name path with '/' as a separator.	uint8 [encoding UTF-8]	
location	An implementation-defined identifier of the location where the violation was detected, for example {filename}:{linenumber}.	uint8 [encoding UTF-8]	
originalContextDataOffset	Original Context Data Offset value passed as input parameter.	uint32	NoUnit
additionalBytes	additionalBytes value passed as input parameter.	uint32	NoUnit

⌋

7.14.4 Production Errors

This chapter contains all Production Errors i.e. Diagnostic Events of this Functional Cluster.

8 API specification

This chapter provides a reference of the APIs defined by this functional cluster. The API is described in the following chapters in tables. Table 8.1 explains the content that is described in such an API table.

Kind:	Defines the kind of the declaration that this API table describes. The following values are supported: • class (Declaration of a class) • function (Declaration of a member or non-member function) • struct (Declaration of a structure) • type alias (Declaration of a type alias) • enumeration (Declaration of an enumeration) • variable (Declaration of a variable)	
Port Interfaces:	States that the C++ API `class` is the related C++ API binding for the given modeled sub-class of `PortInterface`	
Header File:	Defines the header file to be included according to [AP_SWS_GENERAL_90001]	
Forwarding Header File:	Defines the forwarding header file to be included according to [AP_SWS_GENERAL_90001]	
Scope:	Defines the scope that may be a C++ `namespace` (in case of a `class` or non-member function) or a `class` declaration (in case of a member)	
Symbol:	C++ symbol name	
Thread Safety:	Defines whether a function is thread-safe, not thread-safe, or conditional according to [AP_SWS_GENERAL_13200] and [AP_SWS_GENERAL_13202]	
Syntax:	Description of C++ syntax	
Template Param:	Template parameter (0..*)	Template parameter(s) used to parameterize the template
Parameters (in):	Parameter declaration (0..*)	Parameter(s) that are passed to the function
Parameters (out):	Parameter declaration (0..*)	Parameter(s) that are returned to the caller
Return Value:	Return type	Type of the value that the function returns
Exception Safety:	Defines whether a function is exception-safe, not exception safe or conditionally exception safe	
Exceptions:	List of C++ `Exception`s that may be thrown by the function	
Violations:	List of violations that may raised by the function	
Errors:	Error type (0..*)	List of defined `ara::core::ErrorCode`s that may be returned by the function with their recoverability class defined in [RS_AP_00160]. APIs can be extended with vendor-specific error codes. These are not standardized by AUTOSAR
Description:	Brief description of the function	

Table 8.1: Explanation of an API table

8.1 PortInterface to API class binding

This table shows the `API class` binding for each `PortInterface` owned by this functional cluster and those functions taking an `ara::core::InstanceSpecifier` argument, designated to “construct” that class. These constructing functions may be any combination of special-member constructors, named constructor members or non-member factory constructors.

Port Interface	API Class / Function
IdsmContextProviderInterface	[SWS_AIDSM_10500] Definition of API class ara::idsm::ContextDataProvider
	[SWS_AIDSM_10501] Definition of API function ara::idsm::ContextDataProvider::ContextDataProvider
IdsmQualifiedEventReceiverInterface	[SWS_AIDSM_10800] Definition of API class ara::idsm::QualifiedEventsReceiver
	[SWS_AIDSM_10801] Definition of API function ara::idsm::QualifiedEventsReceiver::QualifiedEventsReceiver
IdsmReportingModeProviderInterface	[SWS_AIDSM_10600] Definition of API class ara::idsm::ReportingModeProvider
	[SWS_AIDSM_10601] Definition of API function ara::idsm::ReportingModeProvider::ReportingModeProvider
IdsmTimestampProviderInterface	[SWS_AIDSM_10400] Definition of API class ara::idsm::TimestampProvider
	[SWS_AIDSM_10401] Definition of API function ara::idsm::TimestampProvider::TimestampProvider
SecurityEventReportInterface	[SWS_AIDSM_10101] Definition of API class ara::idsm::EventReporter
	[SWS_AIDSM_10301] Definition of API function ara::idsm::EventReporter::EventReporter

Table 8.2: PortInterface (sub-class) to API class / function binding

8.2 Header: ara/idsm/common.h

8.2.1 Non-Member Types

[SWS_AIDSM_10201] Definition of API type ara::idsm::ContextDataType ⌈

Kind:	type alias
Header file:	#include "ara/idsm/common.h"
Scope:	namespace ara::idsm
Symbol:	ContextDataType
Syntax:	`using ContextDataType = ara::core::Span<const std::uint8_t>;`
Description:	ContextDataType used for sending context data to the IdsM .

⌋

[SWS_AIDSM_10203] Definition of API type ara::idsm::CountType ⌈

Kind:	type alias
Header file:	#include "ara/idsm/common.h"
Scope:	namespace ara::idsm
Symbol:	CountType
Syntax:	`using CountType = std::uint16_t;`
Description:	CountType used for setting optional count for events pre-qualified by sensors .

⌋

[SWS_AIDSM_10205] Definition of API type ara::idsm::EventIdType ⌈

Kind:	type alias
Header file:	#include "ara/idsm/common.h"
Scope:	namespace ara::idsm
Symbol:	EventIdType
Syntax:	`using EventIdType = std::uint16_t;`
Description:	EventIdType for an event .

⌋

[SWS_AIDSM_10207] Definition of API enum ara::idsm::ReportingModeType ⌈

Kind:	enumeration	
Header file:	#include "ara/idsm/common.h"	
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"	
Scope:	namespace ara::idsm	
Symbol:	ReportingModeType	
Underlying type:	std::uint8_t	
Syntax:	`enum class ReportingModeType : std::uint8_t {...};`	
Values:	kOff	--
	kBrief	--
	kDetailed	--
	kBriefBypassingFilters	--
	kDetailedBypassing Filters	--
Description:	Defines an enumeration class for the Reporting Modes. See SWS_AIDSM_00201 for definition. .	

⌋

[SWS_AIDSM_10202] Definition of API type ara::idsm::TimestampType ⌈

Kind:	type alias
Header file:	#include "ara/idsm/common.h"
Scope:	namespace ara::idsm
Symbol:	TimestampType
Syntax:	`using TimestampType = std::uint64_t;`
Description:	TimestampType used for setting optional sensor-specific timestamp for events.
Notes:	Only 62 least-significant bits are used as timestamp value and stored or transmitted, respectively

⌋

[SWS_AIDSM_10206] Definition of API type ara::idsm::VersionedContextData Type ⌈

Kind:	type alias
Header file:	#include "ara/idsm/common.h"
Scope:	namespace ara::idsm
Symbol:	VersionedContextDataType

▽

△

Syntax:	`using VersionedContextDataType = std::pair<ContextDataType, std::uint16_t>;`
Description:	VersionedContextDataType used for sending a versioned context data to the IdsM.

⌋

8.3 Header: ara/idsm/context_data_provider.h

8.3.1 Class: ContextDataProvider

[SWS_AIDSM_10500] Definition of API class ara::idsm::ContextDataProvider ⌈

Kind:	class
Port Interfaces:	`IdsmContextProviderInterface`
Header file:	#include "ara/idsm/context_data_provider.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	ContextDataProvider
Syntax:	`class ContextDataProvider {...};`
Description:	Class for providing context data to the IdsM .

⌋

8.3.1.1 Public Member Functions

8.3.1.1.1 Special Member Functions

[SWS_AIDSM_10503] Definition of API function ara::idsm::ContextDataProvider::ContextDataProvider ⌈

Kind:	function	
Header file:	#include "ara/idsm/context_data_provider.h"	
Scope:	`class ara::idsm::ContextDataProvider`	
Syntax:	`ContextDataProvider (ContextDataProvider &&ra) noexcept;`	
Parameters (in):	ra	The ContextDataProvider object to be moved.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Move constructor for ContextDataProvider.	

⌋

[SWS_AIDSM_10504] Definition of API function ara::idsm::ContextDataProvider::ContextDataProvider ⌈

Kind:	function
Header file:	#include "ara/idsm/context_data_provider.h"
Scope:	`class ara::idsm::ContextDataProvider`
Syntax:	`ContextDataProvider (const ContextDataProvider &) noexcept=delete;`
Description:	The copy constructor for ContextDataProvider shall not be used.

⌋

[SWS_AIDSM_10506] Definition of API function ara::idsm::ContextDataProvider::operator= ⌈

Kind:	function
Header file:	#include "ara/idsm/context_data_provider.h"
Scope:	`class ara::idsm::ContextDataProvider`
Syntax:	`ContextDataProvider & operator= (const ContextDataProvider &) noexcept=delete;`
Description:	The copy assignment operator for ContextDataProvider shall not be used.

⌋

[SWS_AIDSM_10505] Definition of API function ara::idsm::ContextDataProvider::operator= ⌈

Kind:	function	
Header file:	#include "ara/idsm/context_data_provider.h"	
Scope:	`class ara::idsm::ContextDataProvider`	
Syntax:	`ContextDataProvider & operator= (ContextDataProvider &&ra) noexcept;`	
Parameters (in):	ra	The ContextDataProvider object to be moved.
Return value:	ContextDataProvider &	The moved ContextDataProvider object.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	Move assignment operator for ContextDataProvider.	

⌋

[SWS_AIDSM_10502] Definition of API function ara::idsm::ContextDataProvider::~ContextDataProvider ⌈

Kind:	function
Header file:	#include "ara/idsm/context_data_provider.h"
Scope:	`class ara::idsm::ContextDataProvider`
Syntax:	`virtual ~ContextDataProvider () noexcept;`
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Destructor for ContextDataProvider.

⌋

8.3.1.1.2 Constructors

[SWS_AIDSM_10501] Definition of API function ara::idsm::ContextDataProvider::ContextDataProvider

Upstream requirements: RS_AP_00173, RS_AP_00180, RS_AP_00177

⌈

Kind:	function	
Header file:	#include "ara/idsm/context_data_provider.h"	
Scope:	`class ara::idsm::ContextDataProvider`	
Syntax:	`explicit ContextDataProvider (const ara::core::InstanceSpecifier &instance, std::size_t additionalBytes, std::size_t originalContextDataOffset) noexcept;`	
Parameters (in):	instance	instance specifier identifying the PPortPrototype of a IdsmContextDataProviderInterface
	additionalBytes	The number of bytes to be additionally allocated by IdsM for the context data buffer.
	originalContextDataOffset	The offset of the original context data in the context data buffer. This value has to be smaller or equal to the parameter additionalBytes.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Violations:	`InvalidOriginalContextDataOffsetViolation`	If the parameter originalContextDataOffset is larger than the parameter additionalBytes.
	`UnresolvableInstanceSpecifierViolation`	The InstanceSpecifier cannot be resolved.
	`InstanceSpecifierAlreadyInUseViolation`	The InstanceSpecifier is already in use.
	`AraNotInitializedViolation`	Violation message that is sent in case a constructor or function checks for an initialized ara and identifies that ara is not initialized.
Description:	Creation of a ContextDataProvider.	

⌋

8.3.1.1.3 Member Functions

[SWS_AIDSM_10509] Definition of API function ara::idsm::ContextDataProvider::ModifyContextData ⌈

Kind:	function	
Header file:	#include "ara/idsm/context_data_provider.h"	
Scope:	`class ara::idsm::ContextDataProvider`	
Syntax:	`virtual ara::core::Result< std::size_t > ModifyContextData (ara::core::Span< std::uint8_t > contextData, EventIdType event) noexcept=0;`	
Parameters (in):	event	Event ID of the QSEv

▽

△

Parameters (inout):	contextData	Span to the context data buffer in big endian format to be modified by application with a size of the original context data plus additional Bytes.
Return value:	ara::core::Result< std::size_t >	Size of modified context data.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	This function does not specify any standardized errors.	
Description:	ModifyContextData to be invoked by IdsM. IdsM will place the original context data according to the parameter originalContextDataOffset passed to the constructor of the ContextDataProvider. The application that implements this function may modify the context data arbitrarily.	

⌋

[SWS_AIDSM_10507] Definition of API function ara::idsm::ContextDataProvider::Offer ⌈

Kind:	function	
Header file:	#include "ara/idsm/context_data_provider.h"	
Scope:	`class ara::idsm::ContextDataProvider`	
Syntax:	`ara::core::Result< void > Offer () noexcept;`	
Return value:	ara::core::Result< void >	A Result, being either empty or containing any of the errors defined below.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	This function does not specify any standardized errors.	
Description:	Enables potential invocations of ModifyContextData by IdsM. Offer() ignores repeated calls without calling StopOffer() in between. .	

⌋

[SWS_AIDSM_10508] Definition of API function ara::idsm::ContextDataProvider::StopOffer ⌈

Kind:	function
Header file:	#include "ara/idsm/context_data_provider.h"
Scope:	`class ara::idsm::ContextDataProvider`
Syntax:	`void StopOffer () noexcept;`
Return value:	None
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Disables invocations of ModifyContextData. StopOffer ignores repeated calls without calling Offer() in between. .

⌋

8.4 Header: ara/idsm/event_reporter.h

8.4.1 Class: EventReporter

[SWS_AIDSM_10101] Definition of API class ara::idsm::EventReporter ⌈

Kind:	class
Port Interfaces:	SecurityEventReportInterface
Header file:	#include "ara/idsm/event_reporter.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	EventReporter
Syntax:	`class EventReporter final {...};`
Description:	Class for reporting security events to the IdsM .

⌋

8.4.1.1 Public Member Functions

8.4.1.1.1 Constructors

[SWS_AIDSM_10301] Definition of API function ara::idsm::EventReporter::EventReporter

Upstream requirements: RS_AP_00180, RS_AP_00177

⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`EventReporter (const ara::core::InstanceSpecifier &instanceSpecifier) noexcept;`	
Parameters (in):	instanceSpecifier	InstanceSpecifier of the RPortPrototype of type SecurityEventReportInterface that is mapped to the SecurityEventDefinition by means of the SecurityEventMapping (in case an Application reports the security event) or InstanceSpecifier of the FunctionalClusterToSecurityEventDefinitionMapping that maps a module instantiation to the SecurityEventDefinition (in case a module instantiation reports the security event).
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Violations:	`Unresolvable-InstanceSpecifier-Violation`	The InstanceSpecifier cannot be resolved.
	`AraNotInitialized-Violation`	Violation message that is sent in case a constructor or function checks for an initialized ara and identifies that ara is not initialized.
Description:	Construct a new Event Reporter object. Called by the sensor for each event type using the instance specified of the event type .	

⌋

8.4.1.1.2 Member Functions

[SWS_AIDSM_10304] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (ContextDataType contextData, const CountType count=1) noexcept;`	
Parameters (in):	contextData	context data in big endian format
	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event with sensor-provided context data at the IdsM. Please check chapter 7 for more information about context data endianess. .	

⌋

[SWS_AIDSM_10305] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (ContextDataType contextData, const TimestampType timestamp, const CountType count=1) noexcept;`	
Parameters (in):	contextData	context data in big endian format
	timestamp	application provided timestamp
	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event with sensor-provided context data and with a sensor-provided timestamp at the IdsM. Please check chapter 7 for more information about context data endianess. .	

⌋

[SWS_AIDSM_10306] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (VersionedContextDataType contextData, const CountType count=1) noexcept;`	
Parameters (in):	contextData	context data in big endian format

▽

△

	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event with sensor-provided, versioned context data at the IdsM. Please check chapter 7 for more information about context data endianess. .	

⌋

[SWS_AIDSM_10307] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (VersionedContextDataType contextData, const TimestampType timestamp, const CountType count=1) noexcept;`	
Parameters (in):	contextData	context data in big endian format
	timestamp	application provided timestamp
	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event with sensor-provided, versioned context data and with a sensor-provided timestamp at the IdsM. Please check chapter 7 for more information about context data endianess. .	

⌋

[SWS_AIDSM_10302] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (const CountType count=1) noexcept;`	
Parameters (in):	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event at the IdsM. .	

⌋

[SWS_AIDSM_10303] Definition of API function ara::idsm::EventReporter::ReportEvent ⌈

Kind:	function	
Header file:	#include "ara/idsm/event_reporter.h"	
Scope:	`class ara::idsm::EventReporter`	
Syntax:	`void ReportEvent (const TimestampType timestamp, const CountType count=1) noexcept;`	
Parameters (in):	timestamp	application provided timestamp
	count	optional application provided number of event occurences to be reported
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Create a new security event with a sensor-provided timestamp at the IdsM. .	

⌋

8.5 Header: ara/idsm/idsm_error_domain.h

8.5.1 Non-Member Types

[SWS_AIDSM_10204] Definition of API enum ara::idsm::IdsmErrc

Upstream requirements: RS_AP_00130, RS_AP_00122, RS_AP_00127

⌈

Kind:	enumeration	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"	
Scope:	namespace ara::idsm	
Symbol:	IdsmErrc	
Underlying type:	ara::core::ErrorDomain::CodeType	
Syntax:	`enum class IdsmErrc : ara::core::ErrorDomain::CodeType {...};`	
Values:	kUnknownEventId	= 3
		An unknown event ID was provided.
Description:	Defines the error codes for the `ara::idsm::IdsmErrorDomain`	

⌋

8.5.2 Non-Member Functions

8.5.2.1 Other

[SWS_AIDSM_10702] Definition of API function ara::idsm::GetIdsmErrorDomain

Upstream requirements: RS_AP_00120, RS_AP_00130

⌈

Kind:	function	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Scope:	namespace ara::idsm	
Syntax:	`constexpr const ara::core::ErrorDomain & GetIdsmErrorDomain () noexcept;`	
Return value:	const ara::core::Error Domain &	Reference to the `ara::idsm::IdsmErrorDomain` object
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Returns a reference to the `ara::idsm::IdsmErrorDomain` object	

⌋

[SWS_AIDSM_10703] Definition of API function ara::idsm::MakeErrorCode

Upstream requirements: RS_AP_00120, RS_AP_00121, RS_AP_00130

⌈

Kind:	function	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Scope:	namespace ara::idsm	
Syntax:	`constexpr ara::core::ErrorCode MakeErrorCode (ara::idsm::IdsmErrc code, ara::core::ErrorDomain::SupportDataType data) noexcept;`	
Parameters (in):	code	Error code number.
	data	Vendor defined data associated with the error
Return value:	ara::core::ErrorCode	An `ara::core::ErrorCode` object.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Creates an instance of `ara::core::ErrorCode`	

⌋

8.5.3 Class: IdsmErrorDomain

[SWS_AIDSM_10706] Definition of API class ara::idsm::IdsmErrorDomain

Upstream requirements: RS_AP_00130, RS_AP_00122, RS_AP_00127

⌈

Kind:	class
Header file:	#include "ara/idsm/idsm_error_domain.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	IdsmErrorDomain
Base class:	ara::core::ErrorDomain
Syntax:	`class IdsmErrorDomain final : public ara::core::ErrorDomain {...};`
Unique ID:	As per `ara::idsm::IdsmErrorDomain` in [AP_SWS_GENERAL_90023]
Description:	A class representing a firewall error domain.

⌋

8.5.3.1 Public Member Types

[SWS_AIDSM_10707] Definition of API type ara::idsm::IdsmErrorDomain::Errc

Upstream requirements: RS_AP_00120, RS_AP_00130

⌈

Kind:	type alias
Header file:	#include "ara/idsm/idsm_error_domain.h"
Scope:	`class ara::idsm::IdsmErrorDomain`
Symbol:	Errc
Syntax:	`using Errc = IdsmErrc;`
Description:	Alias for the error code value enumeration

⌋

[SWS_AIDSM_10708] Definition of API type ara::idsm::IdsmErrorDomain::Exception

Upstream requirements: RS_AP_00120, RS_AP_00130

⌈

Kind:	type alias
Header file:	#include "ara/idsm/idsm_error_domain.h"
Scope:	`class ara::idsm::IdsmErrorDomain`
Symbol:	Exception
Syntax:	`using Exception = IdsmException;`
Description:	Alias for the exception base class

⌋

8.5.3.2 Public Member Functions

8.5.3.2.1 Special Member Functions

[SWS_AIDSM_10709] Definition of API function ara::idsm::IdsmErrorDomain::IdsmErrorDomain

Upstream requirements: RS_AP_00120, RS_AP_00130

⌈

Kind:	function
Header file:	#include "ara/idsm/idsm_error_domain.h"
Scope:	`class ara::idsm::IdsmErrorDomain`
Syntax:	`IdsmErrorDomain ()=delete;`
Description:	Constructs a new `ara::idsm::IdsmErrorDomain` object

⌋

8.5.3.2.2 Member Functions

[SWS_AIDSM_10711] Definition of API function ara::idsm::IdsmErrorDomain::Message

Upstream requirements: RS_AP_00120, RS_AP_00121, RS_AP_00130

⌈

Kind:	function	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Scope:	`class ara::idsm::IdsmErrorDomain`	
Syntax:	`const char * Message (CodeType errorCode) const noexcept override;`	
Parameters (in):	errorCode	The error code number.
Return value:	const char *	The message associated with the error code
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Returns the message associated with the error code	

⌋

[SWS_AIDSM_10710] Definition of API function ara::idsm::IdsmErrorDomain::Name

Upstream requirements: RS_AP_00120, RS_AP_00130

⌈

Kind:	function
Header file:	#include "ara/idsm/idsm_error_domain.h"
Scope:	`class ara::idsm::IdsmErrorDomain`
Syntax:	`const char * Name () const noexcept override;`

▽

△

Return value:	const char *	As per `ara::idsm::IdsmErrorDomain` in [AP_SWS_GENERAL_90023]
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Retrieve the name of the error domain	

⌋

[SWS_AIDSM_10712] Definition of API function ara::idsm::IdsmErrorDomain::ThrowAsException

Upstream requirements: RS_AP_00120, RS_AP_00121, RS_AP_00130

⌈

Kind:	function	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Scope:	`class ara::idsm::IdsmErrorDomain`	
Syntax:	`void ThrowAsException (const ara::core::ErrorCode &errorCode) const noexcept(false) override;`	
Parameters (in):	errorCode	The error to throw.
Return value:	None	
Exception Safety:	not exception safe	
Thread Safety:	thread-safe	
Description:	Throws the exception associated with the error code. This function does not participate in overload resolution when C++ exceptions are disabled in the compiler toolchain.	

⌋

8.5.4 Class: IdsmException

[SWS_AIDSM_10704] Definition of API class ara::idsm::IdsmException

Upstream requirements: RS_AP_00130, RS_AP_00122, RS_AP_00127

⌈

Kind:	class
Header file:	#include "ara/idsm/idsm_error_domain.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	IdsmException
Base class:	ara::core::Exception
Syntax:	`class IdsmException : public ara::core::Exception {...};`
Description:	Defines a class for exceptions to be thrown by the API.

⌋

8.5.4.1 Public Member Functions

8.5.4.1.1 Constructors

[SWS_AIDSM_10705] Definition of API function ara::idsm::IdsmException::IdsmException

Upstream requirements: RS_AP_00120, RS_AP_00121, RS_AP_00130

⌈

Kind:	function	
Header file:	#include "ara/idsm/idsm_error_domain.h"	
Scope:	`class ara::idsm::IdsmException`	
Syntax:	`explicit IdsmException (ara::core::ErrorCode errorCode) noexcept;`	
Parameters (in):	errorCode	The error code.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Constructs a new `ara::idsm::IdsmException` containing an `ara::core::ErrorCode`	

⌋

8.6 Header: ara/idsm/qualified_events_receiver.h

8.6.1 Class: QualifiedEventsReceiver

[SWS_AIDSM_10800] Definition of API class ara::idsm::QualifiedEventsReceiver

⌈

Kind:	class
Port Interfaces:	`IdsmQualifiedEventReceiverInterface`
Header file:	#include "ara/idsm/qualified_events_receiver.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	QualifiedEventsReceiver
Syntax:	`class QualifiedEventsReceiver {...};`
Description:	Class for receiving qualified events from the IdsM .

⌋

8.6.1.1 Public Member Functions

8.6.1.1.1 Special Member Functions

[SWS_AIDSM_10803] Definition of API function ara::idsm::QualifiedEventsReceiver::QualifiedEventsReceiver ⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	
Syntax:	`QualifiedEventsReceiver (QualifiedEventsReceiver &&ra) noexcept;`	
Parameters (in):	ra	The QualifiedEventsReceiver object to be moved.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Move constructor for QualifiedEventsReceiver.	

⌋

[SWS_AIDSM_10804] Definition of API function ara::idsm::QualifiedEventsReceiver::QualifiedEventsReceiver ⌈

Kind:	function
Header file:	#include "ara/idsm/qualified_events_receiver.h"
Scope:	`class ara::idsm::QualifiedEventsReceiver`
Syntax:	`QualifiedEventsReceiver (const QualifiedEventsReceiver &)=delete;`
Description:	The copy constructor for QualifiedEventsReceiver shall not be used.

⌋

[SWS_AIDSM_10805] Definition of API function ara::idsm::QualifiedEventsReceiver::operator= ⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	
Syntax:	`QualifiedEventsReceiver & operator= (QualifiedEventsReceiver &&ra) noexcept;`	
Parameters (in):	ra	The QualifiedEventsReceiver object to be moved.
Return value:	QualifiedEventsReceiver &	The moved QualifiedEventsReceiver object .
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	Move assignment operator for QualifiedEventsReceiver.	

⌋

[SWS_AIDSM_10806] Definition of API function ara::idsm::QualifiedEventsReceiver::operator= ⌈

Kind:	function
Header file:	#include "ara/idsm/qualified_events_receiver.h"
Scope:	`class ara::idsm::QualifiedEventsReceiver`
Syntax:	`QualifiedEventsReceiver & operator= (const QualifiedEventsReceiver &)=delete;`
Description:	The copy assignment operator for QualifiedEventsReceiver shall not be used.

⌋

[SWS_AIDSM_10802] Definition of API function ara::idsm::QualifiedEventsReceiver::~QualifiedEventsReceiver ⌈

Kind:	function
Header file:	#include "ara/idsm/qualified_events_receiver.h"
Scope:	`class ara::idsm::QualifiedEventsReceiver`
Syntax:	`virtual ~QualifiedEventsReceiver () noexcept;`
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Destructor for QualifiedEventsReceiver.

⌋

8.6.1.1.2 Constructors

[SWS_AIDSM_10801] Definition of API function ara::idsm::QualifiedEventsReceiver::QualifiedEventsReceiver

Upstream requirements: RS_AP_00173, RS_AP_00180, RS_AP_00177

⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	
Syntax:	`explicit QualifiedEventsReceiver (const ara::core::InstanceSpecifier &instance) noexcept;`	
Parameters (in):	instance	instance specifier to the PPortPrototype of a IdsmQualifiedEventReceiverInterface
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Violations:	`UnresolvableInstanceSpecifierViolation`	The InstanceSpecifier cannot be resolved.
	`InstanceSpecifierAlreadyInUseViolation`	The InstanceSpecifier is already in use.

▽

△

	`AraNotInitialized-Violation`	Violation message that is sent in case a constructor or function checks for an initialized ara and identifies that ara is not initialized.
Description:	Creation of an QualifiedEventsReceiver.	

⌋

8.6.1.1.3 Member Functions

[SWS_AIDSM_10808] Definition of API function ara::idsm::QualifiedEventsReceiver::Offer ⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	
Syntax:	`ara::core::Result< void > Offer () noexcept;`	
Return value:	ara::core::Result< void >	A Result, being either empty or containing an error
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	This function does not specify any standardized errors.	
Description:	Enables potential invocations of OnEventQualification by IdsM. Offer ignores repeated calls (without calling StopOffer in between).	

⌋

[SWS_AIDSM_10807] Definition of API function ara::idsm::QualifiedEventsReceiver::OnEventQualification ⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	

Syntax:	`virtual void OnEventQualification (EventIdType event,`
	`ara::core::Optional< ContextDataType > contextData,`
	`ara::core::Optional< TimestampType > timestamp) noexcept=0;`	

Parameters (in):	event	The eventId of the qualified security event
	contextData	The optional unversioned context data of the qualified security event
	timestamp	The optional timestamp of the qualified security event
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	OnEventQualification is implementated by the application and invoked by IdsM on qualification of a security event with unversioned context data. The invocation needs to be enabled before by a call of QualifiedEventsReceiver::Offer.	

⌋

[SWS_AIDSM_10810] Definition of API function ara::idsm::QualifiedEventsReceiver::OnEventQualification ⌈

Kind:	function	
Header file:	#include "ara/idsm/qualified_events_receiver.h"	
Scope:	`class ara::idsm::QualifiedEventsReceiver`	
Syntax:	`virtual void OnEventQualification (EventIdType event, ara::core::Optional< VersionedContextDataType > contextData, ara::core::Optional< TimestampType > timestamp) noexcept=0;`	
Parameters (in):	event	The eventId of the qualified security event
	contextData	The optional versionedcontext data of the qualified security event
	timestamp	The optional timestamp of the qualified security event
Return value:	None	
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	OnEventQualification is implementated by the application and invoked by IdsM on qualification of a security event with versioned context data. The invocation needs to be enabled before by a call of QualifiedEventsReceiver::Offer.	

⌋

[SWS_AIDSM_10809] Definition of API function ara::idsm::QualifiedEventsReceiver::StopOffer ⌈

Kind:	function
Header file:	#include "ara/idsm/qualified_events_receiver.h"
Scope:	`class ara::idsm::QualifiedEventsReceiver`
Syntax:	`void StopOffer () noexcept;`
Return value:	None
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Disables invocations of OnEventQualification.

⌋

8.7 Header: ara/idsm/reporting_mode_provider.h

8.7.1 Class: ReportingModeProvider

[SWS_AIDSM_10600] Definition of API class ara::idsm::ReportingModeProvider ⌈

Kind:	class
Port Interfaces:	`IdsmReportingModeProviderInterface`
Header file:	#include "ara/idsm/reporting_mode_provider.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm

▽

△

Symbol:	ReportingModeProvider
Syntax:	`class ReportingModeProvider final {...};`
Description:	Class for providing ReportingModes to the IdsM .

⌋

8.7.1.1 Public Member Functions

8.7.1.1.1 Constructors

[SWS_AIDSM_10601] Definition of API function ara::idsm::ReportingMode Provider::ReportingModeProvider

Upstream requirements: RS_AP_00173, RS_AP_00180, RS_AP_00177

⌈

Kind:	function	
Header file:	#include "ara/idsm/reporting_mode_provider.h"	
Scope:	`class ara::idsm::ReportingModeProvider`	
Syntax:	`explicit ReportingModeProvider (const ara::core::InstanceSpecifier &instance) noexcept;`	
Parameters (in):	instance	instance specifier to the PPortPrototype of a IdsmReportingMode ProviderInterface
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Violations:	`Unresolvable-InstanceSpecifier-Violation`	The InstanceSpecifier cannot be resolved.
	`InstanceSpecifier-AlreadyInUse-Violation`	The InstanceSpecifier is already in use.
	`AraNotInitialized-Violation`	Violation message that is sent in case a constructor or function checks for an initialized ara and identifies that ara is not initialized.
Description:	Creation of an ReportingModeProvider.	

⌋

8.7.1.1.2 Member Functions

[SWS_AIDSM_10602] Definition of API function ara::idsm::ReportingMode Provider::GetReportingMode ⌈

Kind:	function
Header file:	#include "ara/idsm/reporting_mode_provider.h"
Scope:	`class ara::idsm::ReportingModeProvider`
Syntax:	`ara::core::Result< ReportingModeType > GetReportingMode (EventIdType eventId) noexcept;`

▽

△

Parameters (in):	eventId	ID of the event for which the reporting mode shall be queried.
Return value:	ara::core::Result< ReportingModeType >	A Result, being either the current reporting mode, or containing any of the errors defined below.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	IdsmErrc::kUnknown EventId	rollback_semantics
		Returned if the eventId does not identify a configured event.
Description:	Get the ReportingMode for the event identified by an Event ID. .	

⌋

[SWS_AIDSM_10603] Definition of API function ara::idsm::ReportingMode Provider::SetReportingMode ⌈

Kind:	function	
Header file:	#include "ara/idsm/reporting_mode_provider.h"	
Scope:	`class ara::idsm::ReportingModeProvider`	
Syntax:	`ara::core::Result< void > SetReportingMode (EventIdType eventId, ReportingModeType reportingMode) noexcept;`	
Parameters (in):	eventId	ID of the event for which the reporting mode shall be changed.
	reportingMode	The reporting mode to be set.
Return value:	ara::core::Result< void >	A Result, being either empty or containing any of the errors defined below.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	IdsmErrc::kUnknown EventId	rollback_semantics
		Returned if the eventId does not identify a configured event.
Description:	Set the ReportingMode for the event identified by an Event ID.	

⌋

8.8 Header: ara/idsm/timestamp_provider.h

8.8.1 Class: TimestampProvider

[SWS_AIDSM_10400] Definition of API class ara::idsm::TimestampProvider ⌈

Kind:	class
Port Interfaces:	`IdsmTimestampProviderInterface`
Header file:	#include "ara/idsm/timestamp_provider.h"
Forwarding header file:	#include "ara/idsm/idsm_fwd.h"
Scope:	namespace ara::idsm
Symbol:	TimestampProvider
Syntax:	`class TimestampProvider {...};`
Description:	Class for providing timestamps to the IdsM .

⌋

8.8.1.1 Public Member Functions

8.8.1.1.1 Special Member Functions

[SWS_AIDSM_10403] Definition of API function ara::idsm::TimestampProvider::TimestampProvider ⌈

Kind:	function	
Header file:	#include "ara/idsm/timestamp_provider.h"	
Scope:	`class ara::idsm::TimestampProvider`	
Syntax:	`TimestampProvider (TimestampProvider &&ra) noexcept;`	
Parameters (in):	ra	The TimestampProvider object to be moved.
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Description:	Move constructor for TimestampProvider.	

⌋

[SWS_AIDSM_10404] Definition of API function ara::idsm::TimestampProvider::TimestampProvider ⌈

Kind:	function
Header file:	#include "ara/idsm/timestamp_provider.h"
Scope:	`class ara::idsm::TimestampProvider`
Syntax:	`TimestampProvider (const TimestampProvider &) noexcept=delete;`
Description:	The copy constructor for TimestampProvider shall not be used.

⌋

[SWS_AIDSM_10406] Definition of API function ara::idsm::TimestampProvider::operator= ⌈

Kind:	function
Header file:	#include "ara/idsm/timestamp_provider.h"
Scope:	`class ara::idsm::TimestampProvider`
Syntax:	`TimestampProvider & operator= (const TimestampProvider &) noexcept=delete;`
Description:	The copy assignment operator for TimestampProvider shall not be used.

⌋

[SWS_AIDSM_10405] Definition of API function ara::idsm::TimestampProvider::operator= ⌈

Kind:	function	
Header file:	#include "ara/idsm/timestamp_provider.h"	
Scope:	`class ara::idsm::TimestampProvider`	
Syntax:	`TimestampProvider & operator= (TimestampProvider &&ra) noexcept;`	
Parameters (in):	ra	The TimestampProvider object to be moved.

▽

△

Return value:	TimestampProvider &	The moved TimestampProvider object.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	Move assignment operator for TimestampProvider.	

⌋

[SWS_AIDSM_10402] Definition of API function ara::idsm::TimestampProvider::~TimestampProvider ⌈

Kind:	function
Header file:	#include "ara/idsm/timestamp_provider.h"
Scope:	`class ara::idsm::TimestampProvider`
Syntax:	`virtual ~TimestampProvider () noexcept;`
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Destructor for TimestampProvider.

⌋

8.8.1.1.2 Constructors

[SWS_AIDSM_10401] Definition of API function ara::idsm::TimestampProvider::TimestampProvider

Upstream requirements: RS_AP_00173, RS_AP_00180, RS_AP_00177

⌈

Kind:	function	
Header file:	#include "ara/idsm/timestamp_provider.h"	
Scope:	`class ara::idsm::TimestampProvider`	
Syntax:	`explicit TimestampProvider (const ara::core::InstanceSpecifier &instance) noexcept;`	
Parameters (in):	instance	instance specifier to the PPortPrototype of a IdsmTimestampProviderInterface
Exception Safety:	exception safe	
Thread Safety:	thread-safe	
Violations:	`Unresolvable-InstanceSpecifier-Violation`	The InstanceSpecifier cannot be resolved.
	`InstanceSpecifier-AlreadyInUse-Violation`	The InstanceSpecifier is already in use.
	`AraNotInitialized-Violation`	Violation message that is sent in case a constructor or function checks for an initialized ara and identifies that ara is not initialized.
Description:	Creation of an TimestampProvider.	

⌋

8.8.1.1.3 Member Functions

[SWS_AIDSM_10407] Definition of API function ara::idsm::TimestampProvider::GetTimestamp ⌈

Kind:	function	
Header file:	#include "ara/idsm/timestamp_provider.h"	
Scope:	`class ara::idsm::TimestampProvider`	
Syntax:	`virtual TimestampType GetTimestamp () noexcept=0;`	
Return value:	TimestampType	The application provided Timestamp
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Description:	GetTimestamp to be invoked by IdsM. The invocation needs to be enabled before by a call of TimestampProvider::Offer.	

⌋

[SWS_AIDSM_10408] Definition of API function ara::idsm::TimestampProvider::Offer ⌈

Kind:	function	
Header file:	#include "ara/idsm/timestamp_provider.h"	
Scope:	`class ara::idsm::TimestampProvider`	
Syntax:	`ara::core::Result< void > Offer () noexcept;`	
Return value:	ara::core::Result< void >	A Result, being either empty or containing any of the errors defined below.
Exception Safety:	exception safe	
Thread Safety:	non_threadsafe	
Errors:	This function does not specify any standardized errors.	
Description:	Enables potential invocations of GetTimestamp by IdsM. Offer() ignores repeated calls without calling StopOffer() in between. .	

⌋

[SWS_AIDSM_10409] Definition of API function ara::idsm::TimestampProvider::StopOffer ⌈

Kind:	function
Header file:	#include "ara/idsm/timestamp_provider.h"
Scope:	`class ara::idsm::TimestampProvider`
Syntax:	`void StopOffer () noexcept;`
Return value:	None
Exception Safety:	exception safe
Thread Safety:	non_threadsafe
Description:	Disables invocations of GetTimestamp. StopOffer ignores repeated calls without calling Offer() in between. .

⌋

9 Service Interfaces

This functional cluster does not define any provided or required service interfaces.

10 Configuration

The configuration model of this functional cluster is defined in [10]. This chapter defines the default values for attributes and semantic constraints for elements specified in [10] that are part of the configuration model of this functional cluster.

10.1 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters are described in Chapter 7 and Chapter 8.

10.1.1 IdsM Instantiation

10.1.1.1 IdsM Instantiation

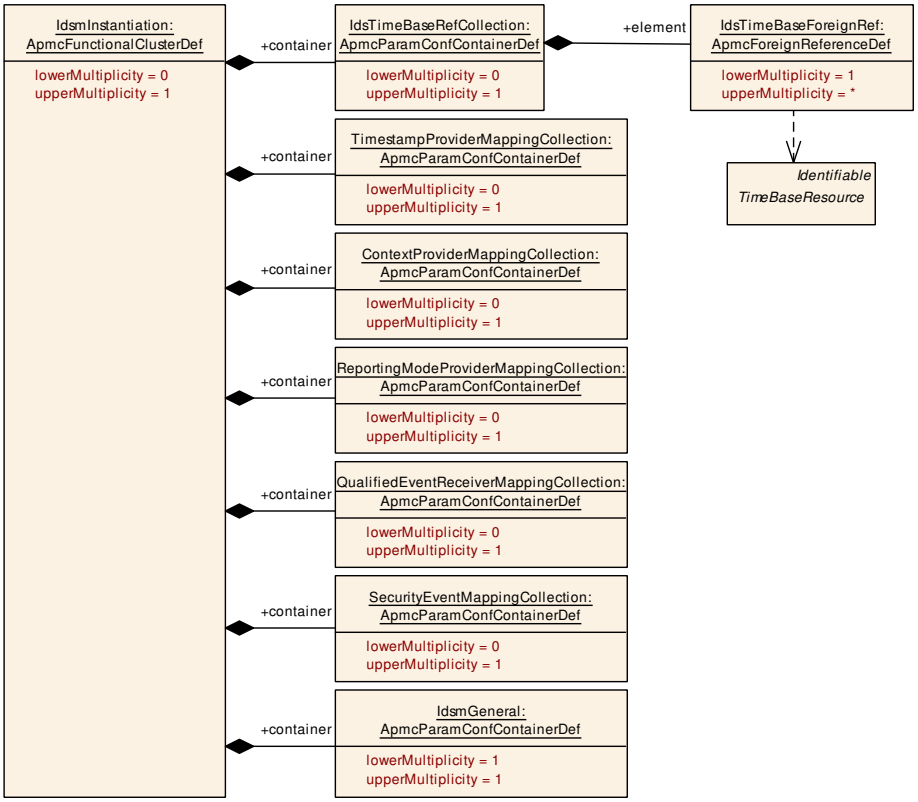


Figure 10.1: Intrusion Detection Management Instantiation Configuration

[APMC_AIDSM_00001] Definition of ApmcFunctionalClusterDef IdsmInstantiation ⌈

Functional Cluster Name	IdsmInstantiation
Description	This functional cluster defines the configuration for the intrusion detection management on a specific machine.

Included Containers		
Container Name	**Multiplicity**	**Description**
ContextProviderMappingCollection	0..1	This container represents the collection of context provider mappings in the context of the enclosing intrusion detection management.
IdsTimeBaseRefCollection	0..1	This container represents the collection of reference to a `TimeBaseResource` for obtaining time stamps.
IdsmGeneral	1	This container holds some general configuration items that apply for the entire functional cluster.
QualifiedEventReceiverMapping Collection	0..1	This container represents the collection of qualified event receiver mappings in the context of the enclosing intrusion detection management.
ReportingModeProviderMapping Collection	0..1	This container represents the collection of reporting mode provider mappings in the context of the enclosing intrusion detection management.
SecurityEventMappingCollection	0..1	This container represents the collection of security event mappings within the enclosing intrusion detection management.
TimestampProviderMapping Collection	0..1	This container represents the collection of timestamp provider mappings in the context of the enclosing intrusion detection management.

⌋

10.1.1.2 IdsmGeneral

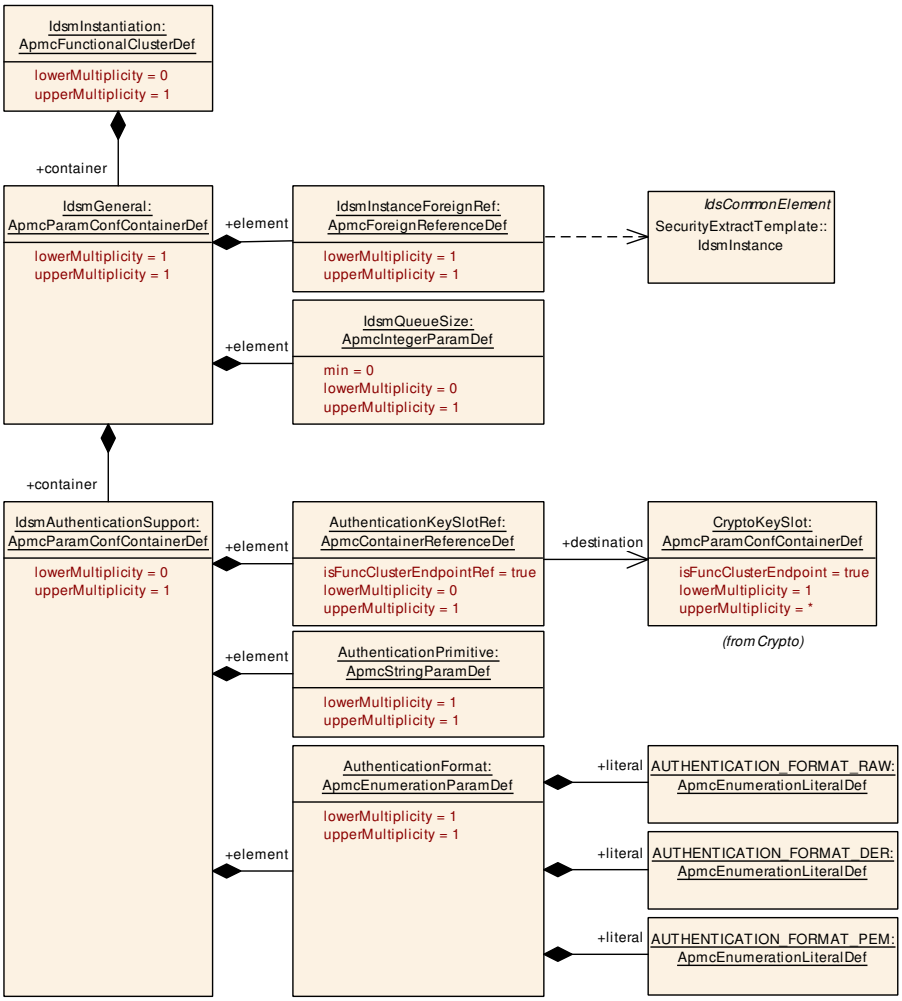


Figure 10.2: General configuration for the intrusion detection system manager

[APMC_AIDSM_00026] Definition of ApmcParamConfContainerDef IdsmGeneral ⌈

Container Name	IdsmGeneral
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container holds some general configuration items that apply for the entire functional cluster.
Multiplicity	1
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmQueueSize	0..1	[APMC_AIDSM_00027]
IdsmInstanceForeignRef	1	[APMC_AIDSM_00025]

Included Containers		
Container Name	**Multiplicity**	**Description**
IdsmAuthenticationSupport	0..1	This container provides the configuration elements for the authentication support in the context of IdsM.

⌋

[APMC_AIDSM_00027] Definition of ApmcIntegerParamDef IdsmQueueSize ⌈

Parameter Name	IdsmQueueSize	
Parent Container	IdsmGeneral	
Upstream mapping defined	–	
Description	Length of the queue in which IDS events can be stored before processing, if immediate processing is not possible.	
Multiplicity	0..1	
Type	ApmcIntegerParamDef	
Range	0 .. 18446744073709551615	
Default value	–	

⌋

[APMC_AIDSM_00025] Definition of ApmcForeignReferenceDef IdsmInstanceForeignRef ⌈

Parameter Name	IdsmInstanceForeignRef
Parent Container	IdsmGeneral
Upstream mapping defined	–
Description	This foreign reference identifies the `IdsmInstance` that is associated with the enclosing IdsmInstantiation.
Multiplicity	1
Type	Foreign reference to IDSM-INSTANCE

⌋

[APMC_AIDSM_00029] Definition of ApmcParamConfContainerDef IdsmAuthenticationSupport ⌈

Container Name	IdsmAuthenticationSupport
Parent Container	IdsmGeneral
Upstream mapping defined	–
Description	This container provides the configuration elements for the authentication support in the context of IdsM.
Multiplicity	0..1
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
AuthenticationFormat	1	[APMC_AIDSM_00031]
AuthenticationPrimitive	1	[APMC_AIDSM_00030]
AuthenticationKeySlotRef	0..1	[APMC_AIDSM_00028]

No Included Containers

⌋

[APMC_AIDSM_00031] Definition of ApmcEnumerationParamDef Authentication Format ⌈

Parameter Name	AuthenticationFormat	
Parent Container	IdsmAuthenticationSupport	
Upstream mapping defined	–	
Description	This parameter describes several dedicated options for the configuration of the authentication format.	
Multiplicity	1	
Type	ApmcEnumerationParamDef	
Range	AUTHENTICATION_FORMAT_DER	The DER format is used if the signature is supposed to be stored in the DER (distinguished encoding rules) format, see~[SWS_CRYPT_04210] in [8].
	AUTHENTICATION_FORMAT_PEM	The PEM format is used if the signature is supposed to be stored in the PEM (privacy-enhanced electronic mail) format, see [SWS_CRYPT_04211] in [8].
	AUTHENTICATION_FORMAT_RAW	The RAW format is used if the signature is supposed to be stored in raw format, see [SWS_CRYPT_60022] in [8].

⌋

[APMC_AIDSM_00030] Definition of ApmcStringParamDef AuthenticationPrimitive ⌈

Parameter Name	AuthenticationPrimitive
Parent Container	IdsmAuthenticationSupport
Upstream mapping defined	–
Description	This parameter configures a textual identification of the cryptographic primitive used to implement the authentication support.
Multiplicity	1
Type	ApmcStringParamDef
Default value	–
Verbatim String	–

⌋

[APMC_AIDSM_00028] Definition of ApmcContainerReferenceDef Authentication KeySlotRef ⌈

Parameter Name	AuthenticationKeySlotRef
Parent Container	IdsmAuthenticationSupport
Upstream mapping defined	–
Description	This reference identifies the `CryptoKeySlot` that provides the authentication service to the IdsM.
Multiplicity	0..1
Type	Reference to CryptoKeySlot

⌋

10.1.1.3 IdsTimeBaseRefCollection

[APMC_AIDSM_00023] Definition of ApmcParamConfContainerDef IdsTimeBaseRefCollection ⌈

Container Name	IdsTimeBaseRefCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of reference to a `TimeBaseResource` for obtaining time stamps.
Multiplicity	0..1
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsTimeBaseForeignRef	1..*	[APMC_AIDSM_00024]

No Included Containers

⌋

[APMC_AIDSM_00024] Definition of ApmcForeignReferenceDef IdsTimeBaseForeignRef ⌈

Parameter Name	IdsTimeBaseForeignRef
Parent Container	IdsTimeBaseRefCollection
Upstream mapping defined	–
Description	This reference identifies a `TimeBaseResource` used for obtaining time stamps.
Multiplicity	1..*
Type	Foreign reference to TIME-BASE-RESOURCE

⌋

10.1.2 IdsM Context Provider Mapping

10.1.2.1 IdsmContextProviderMappingCollection

[APMC_AIDSM_00011] Definition of ApmcParamConfContainerDef ContextProviderMappingCollection ⌈

Container Name	ContextProviderMappingCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of context provider mappings in the context of the enclosing intrusion detection management.
Multiplicity	0..1
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
ContextProviderMapping	1..*	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `PPortPrototype` that is typed by a `IdsmContextProviderInterface`.

⌋

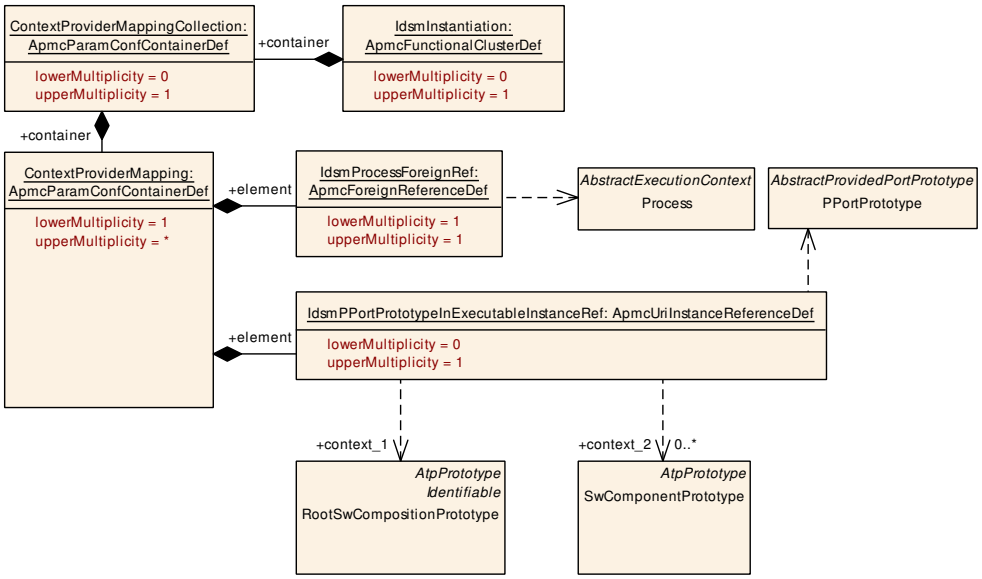


Figure 10.3: Intrusion Detection Management Context Provider Mapping Configuration

10.1.2.2 IdsmContextProviderMapping

[APMC_AIDSM_00012] Definition of ApmcParamConfContainerDef Context ProviderMapping ⌈

Container Name	ContextProviderMapping
Parent Container	ContextProviderMappingCollection
Upstream mapping defined	–
Description	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `PPortPrototype` that is typed by a `IdsmContextProviderInterface`.
Multiplicity	1..*
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmPPortPrototypeInExecutableInstanceRef	0..1	[APMC_AIDSM_00014]
IdsmProcessForeignRef	1	[APMC_AIDSM_00013]

No Included Containers

⌋

[APMC_AIDSM_00014] Definition of ApmcUriInstanceReferenceDef IdsmPPort PrototypeInExecutableInstanceRef ⌈

Parameter Name	IdsmPPortPrototypeInExecutableInstanceRef
Parent Container	ContextProviderMapping, TimestampProviderMapping
Upstream mapping defined	–
Description	This instanceRef identifies the PPortPrototype That is applicable for the enclosing mapping.
Multiplicity	0..1
Type	Instance reference to P-PORT-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE*

⌋

[APMC_AIDSM_00013] Definition of ApmcForeignReferenceDef IdsmProcess ForeignRef ⌈

Parameter Name	IdsmProcessForeignRef
Parent Container	ContextProviderMapping, QualifiedEventReceiverMapping, ReportingModeProvider Mapping, SecurityEventMapping, TimestampProviderMapping
Upstream mapping defined	–
Description	This reference identifies the Process in which context the enclosing mapping is placed.
Multiplicity	1
Type	Foreign reference to PROCESS

⌋

10.1.3 IdsM Security Event Mapping

10.1.3.1 IdsmSecurityEventMappingCollection

[APMC_AIDSM_00002] Definition of ApmcParamConfContainerDef Security EventMappingCollection ⌈

Container Name	SecurityEventMappingCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of security event mappings within the enclosing intrusion detection management.
Multiplicity	0..1
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
SecurityEventMapping	1..*	This container represents the ability to map SecurityEventDefinition to Process on target-configuration level and to a given PPortPrototype that is typed by a IdsmTimestampProviderInterface.

⌋

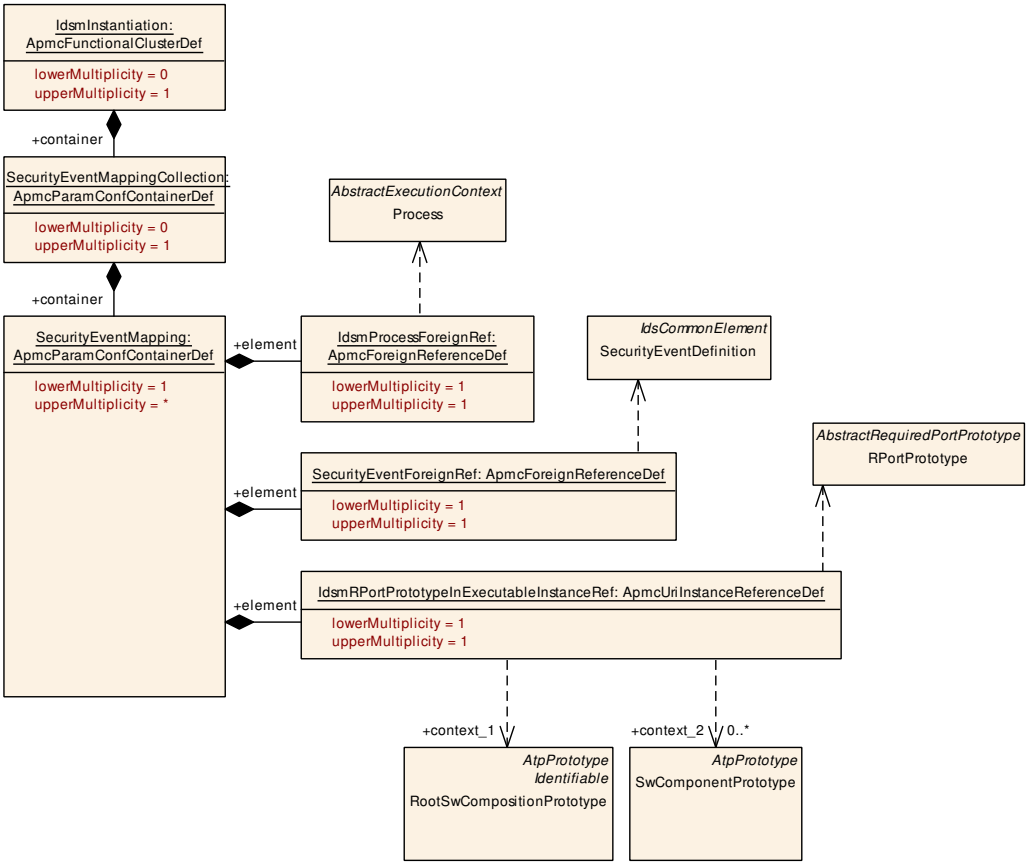


Figure 10.4: Intrusion Detection Management Security Event Mapping Configuration

10.1.3.2 IdsmSecurityEventMapping

[APMC_AIDSM_00003] Definition of ApmcParamConfContainerDef Security EventMapping ⌈

Container Name	SecurityEventMapping
Parent Container	SecurityEventMappingCollection
Upstream mapping defined	–
Description	This container represents the ability to map `SecurityEventDefinition` to `Process` on target-configuration level and to a given `PPortPrototype` that is typed by a `IdsmTimestampProviderInterface`.
Multiplicity	1..*
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmProcessForeignRef	1	[APMC_AIDSM_00013]
IdsmRPortPrototypeInExecutableInstanceRef	1	[APMC_AIDSM_00018]
SecurityEventForeignRef	1	[APMC_AIDSM_00005]

No Included Containers

⌋

For parameter table [APMC_AIDSM_00013] IdsmProcessForeignRef, see definition below container ContextProviderMapping.

For parameter table [APMC_AIDSM_00018] IdsmRPortPrototypeInExecutableInstanceRef, see definition below container QualifiedEventReceiverMapping.

[APMC_AIDSM_00005] Definition of ApmcForeignReferenceDef SecurityEventForeignRef ⌈

Parameter Name	SecurityEventForeignRef
Parent Container	SecurityEventMapping
Upstream mapping defined	–
Description	This reference identifies the corresponding `SecurityEventDefinition`.
Multiplicity	1
Type	Foreign reference to SECURITY-EVENT-DEFINITION

⌋

10.1.4 IdsM Timestamp Provider Mapping

10.1.4.1 IdsmTimestampProviderMappingCollection

[APMC_AIDSM_00007] Definition of ApmcParamConfContainerDef TimestampProviderMappingCollection ⌈

Container Name	TimestampProviderMappingCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of timestamp provider mappings in the context of the enclosing intrusion detection management.
Multiplicity	0..1
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	Multiplicity	Description
TimestampProviderMapping	1..*	This container represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `PPortPrototype` that is typed by a `IdsmTimestampProviderInterface`.

⌋

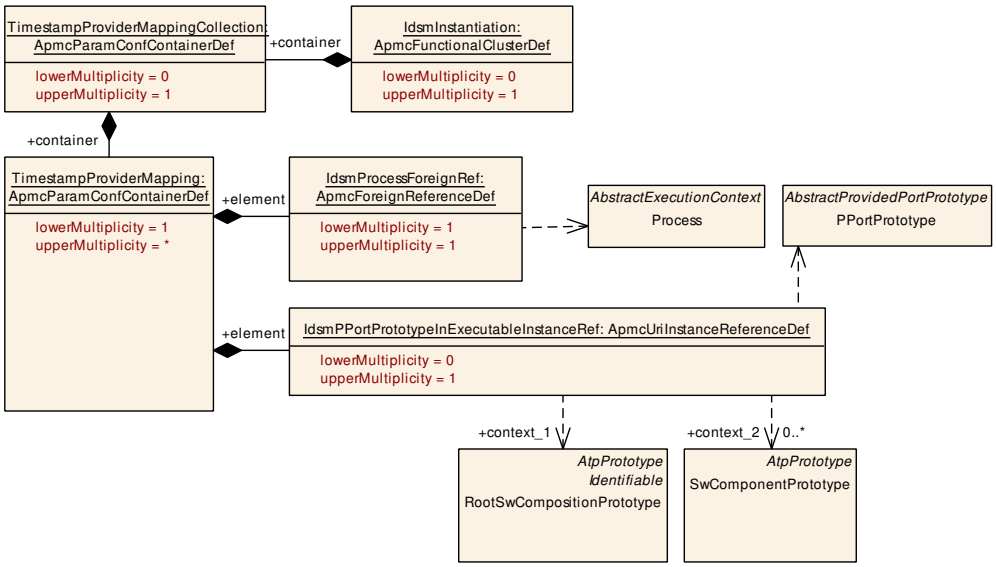


Figure 10.5: Intrusion Detection Management Timestamp Provider Mapping Configuration

10.1.4.2 IdsmTimestampProviderMapping

[APMC_AIDSM_00008] Definition of ApmcParamConfContainerDef Timestamp ProviderMapping ⌈

Container Name	TimestampProviderMapping
Parent Container	TimestampProviderMappingCollection
Upstream mapping defined	–
Description	This container represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `PPortPrototype` that is typed by a `IdsmTimestampProviderInterface`.
Multiplicity	1..*
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmPPortPrototypeInExecutableInstanceRef	0..1	[APMC_AIDSM_00014]
IdsmProcessForeignRef	1	[APMC_AIDSM_00013]

No Included Containers

⌋

For parameter table [APMC_AIDSM_00014] IdsmPPortPrototypeInExecutableInstanceRef, see definition below container ContextProviderMapping.

For parameter table [APMC_AIDSM_00013] IdsmProcessForeignRef, see definition below container ContextProviderMapping.

10.1.5 IdsM Reporting Mode Provider Mapping

10.1.5.1 IdsmReportingModeProviderMappingCollection

[APMC_AIDSM_00015] Definition of ApmcParamConfContainerDef ReportingModeProviderMappingCollection ⌈

Container Name	ReportingModeProviderMappingCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of reporting mode provider mappings in the context of the enclosing intrusion detection management.
Multiplicity	0..1
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
ReportingModeProviderMapping	1..*	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `RPortPrototype` that is typed by a `IdsmReportingModeProviderInterface`.

⌋

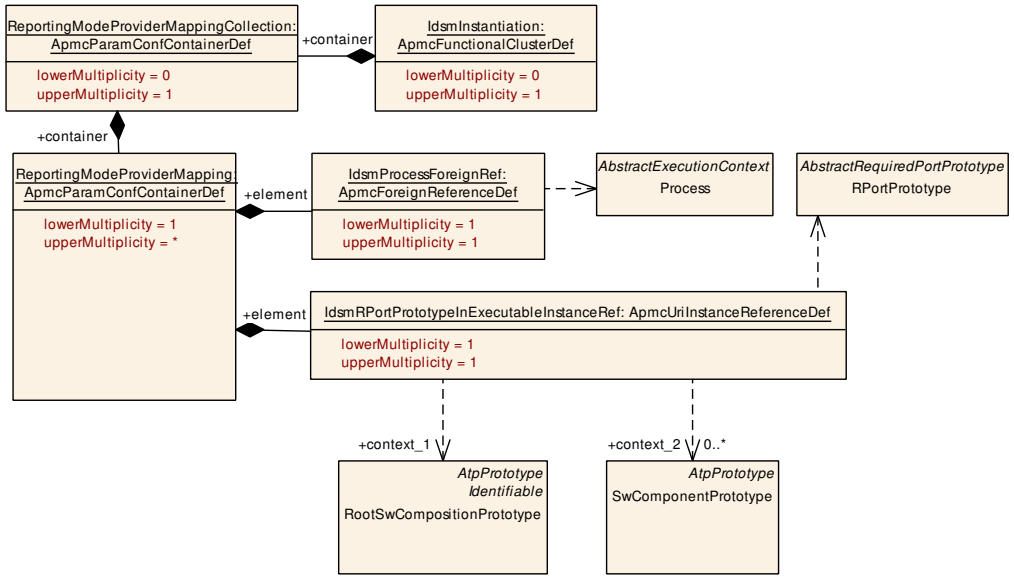


Figure 10.6: Intrusion Detection Management Reporting Mode Provider Mapping Configuration

10.1.5.2 IdsmReportingModeProviderMapping

[APMC_AIDSM_00016] Definition of ApmcParamConfContainerDef ReportingModeProviderMapping ⌈

Container Name	ReportingModeProviderMapping
Parent Container	ReportingModeProviderMappingCollection
Upstream mapping defined	–
Description	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `RPortPrototype` that is typed by a `IdsmReportingModeProviderInterface`.
Multiplicity	1..*
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmProcessForeignRef	1	[APMC_AIDSM_00013]
IdsmRPortPrototypeInExecutableInstanceRef	1	[APMC_AIDSM_00018]

No Included Containers

⌋

For parameter table [APMC_AIDSM_00013] IdsmProcessForeignRef, see definition below container ContextProviderMapping.

For parameter table [APMC_AIDSM_00018] IdsmRPortPrototypeInExecutableInstanceRef, see definition below container QualifiedEventReceiverMapping.

10.1.6 IdsM Qualified Event Receiver Mapping

10.1.6.1 IdsmQualifiedEventReceiverMappingCollection

[APMC_AIDSM_00019] Definition of ApmcParamConfContainerDef QualifiedEventReceiverMappingCollection ⌈

Container Name	QualifiedEventReceiverMappingCollection
Parent Container	IdsmInstantiation
Upstream mapping defined	–
Description	This container represents the collection of qualified event receiver mappings in the context of the enclosing intrusion detection management.
Multiplicity	0..1
Configuration Parameters	

No Included Parameters

Included Containers		
Container Name	**Multiplicity**	**Description**
QualifiedEventReceiverMapping	1..*	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `RPortPrototype` that is typed by a `IdsmQualifiedEventReceiverInterface`.

⌋

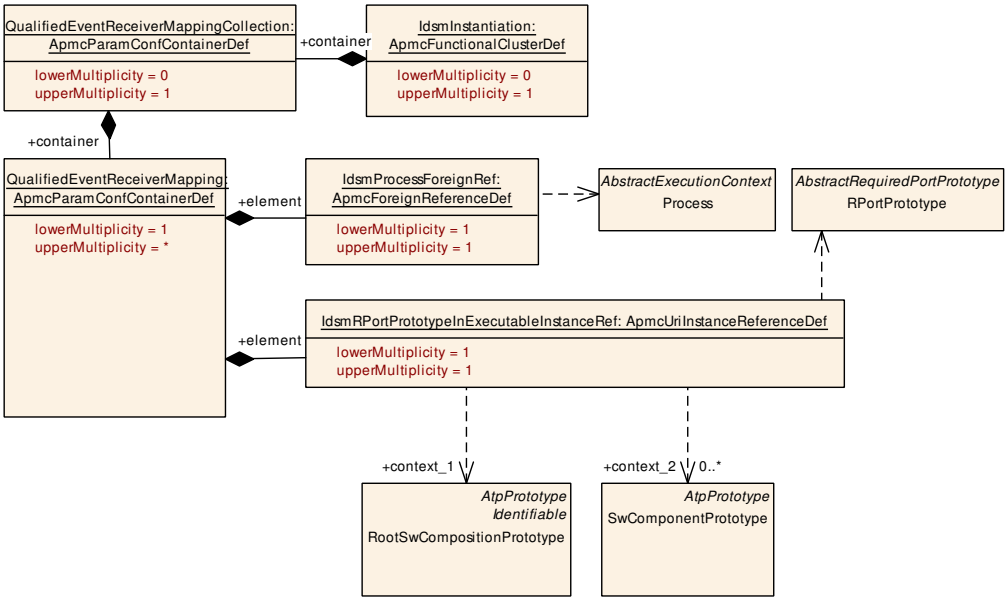


Figure 10.7: Intrusion Detection Management Qualified Event Receiver Mapping Configuration

10.1.6.2 IdsmQualifiedEventReceiverMapping

[APMC_AIDSM_00020] Definition of ApmcParamConfContainerDef QualifiedEventReceiverMapping ⌈

Container Name	QualifiedEventReceiverMapping
Parent Container	QualifiedEventReceiverMappingCollection
Upstream mapping defined	–
Description	This meta-class represents the ability to define a mapping to a `Process` on target-configuration level and to to a given `RPortPrototype` that is typed by a `IdsmQualifiedEventReceiverInterface`.
Multiplicity	1..*
Configuration Parameters	

Included Parameters		
Parameter Name	**Multiplicity**	**Apmc ID**
IdsmProcessForeignRef	1	[APMC_AIDSM_00013]
IdsmRPortPrototypeInExecutableInstanceRef	1	[APMC_AIDSM_00018]

No Included Containers

⌋

For parameter table [APMC_AIDSM_00013] IdsmProcessForeignRef, see definition below container ContextProviderMapping.

[APMC_AIDSM_00018] Definition of ApmcUriInstanceReferenceDef IdsmRPortPrototypeInExecutableInstanceRef ⌈

Parameter Name	IdsmRPortPrototypeInExecutableInstanceRef
Parent Container	QualifiedEventReceiverMapping, ReportingModeProviderMapping, SecurityEventMapping
Upstream mapping defined	–
Description	This instanceRef identifies the `RPortPrototype` that is applicable for the enclosing mapping.
Multiplicity	1
Type	Instance reference to R-PORT-PROTOTYPE context: ROOT-SW-COMPOSITION-PROTOTYPE SW-COMPONENT-PROTOTYPE*

⌋

10.1.7 Correspondence Table

The following table lists the corresponding model elements for the APMC-based `Target Configuration` and for the legacy M2-based `Target Configuration`.

In Chapter 7, only the APMC-based configuration elements are used.

Readers interested to read Chapter 7 may use the table to understand, how the M2-based legacy approach corresponds to the APMC-based configuration elements.

M1 Model Element	M2 Model Element
`IdsmInstantiation`	`IdsPlatformInstantiation`
`ContextProviderMapping`	`IdsmContextProviderMapping`
`ContextProviderMapping.IdsmPPortPrototypeInExecutableInstanceRef`	`IdsmTimestampProviderMapping.pPortPrototypeInExecutable`
`ContextProviderMapping.IdsmPPortPrototypeInExecutableInstanceRef`	`IdsmContextProviderMapping.pPortPrototypeInExecutable`
`IdsTimeBaseRefCollection`	`IdsPlatformInstantiation.idsTimeBase`
`IdsTimeBaseRefCollection.IdsTimeBaseForeignRef`	`IdsPlatformInstantiation.idsTimeBase`
`IdsmGeneral.IdsmInstanceForeignRef`	`IdsmInstance.idsmModuleInstantiation`
`IdsmAuthenticationSupport`	`IdsmModuleInstantiationToCryptoKeySlotMapping`
`IdsmAuthenticationSupport.AuthenticationFormat`	`IdsmModuleInstantiationToCryptoKeySlotMapping.authenticatorFormat`
`IdsmAuthenticationSupport.AuthenticationKeySlotRef`	`IdsmModuleInstantiationToCryptoKeySlotMapping.cryptoKeySlot`
`IdsmAuthenticationSupport.AuthenticationPrimitive`	`IdsmModuleInstantiationToCryptoKeySlotMapping.cryptoPrimitive`
`QualifiedEventReceiverMapping`	`IdsmQualifiedEventReceiverMapping`

▽

△

M1 Model Element	M2 Model Element
QualifiedEventReceiverMapping. IdsmRPortPrototypeInExecutableInstanceRef	SecurityEventMapping.reportingPortPrototype
QualifiedEventReceiverMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmQualifiedEventReceiverMapping. rPortPrototypeInExecutable
QualifiedEventReceiverMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmReportingModeProviderMapping. rPortPrototypeInExecutable
ReportingModeProviderMapping	IdsmReportingModeProviderMapping
ReportingModeProviderMapping. IdsmRPortPrototypeInExecutableInstanceRef	SecurityEventMapping.reportingPortPrototype
ReportingModeProviderMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmQualifiedEventReceiverMapping. rPortPrototypeInExecutable
ReportingModeProviderMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmReportingModeProviderMapping. rPortPrototypeInExecutable
SecurityEventMappingCollection	IdsmModuleInstantiation. reportableSecurityEvent
SecurityEventMapping	SecurityEventMapping
SecurityEventMapping. IdsmRPortPrototypeInExecutableInstanceRef	SecurityEventMapping.reportingPortPrototype
SecurityEventMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmQualifiedEventReceiverMapping. rPortPrototypeInExecutable
SecurityEventMapping. IdsmRPortPrototypeInExecutableInstanceRef	IdsmReportingModeProviderMapping. rPortPrototypeInExecutable
SecurityEventMapping.SecurityEventForeignRef	SecurityEventMapping.securityEvent
TimestampProviderMapping	IdsmTimestampProviderMapping
TimestampProviderMapping. IdsmPPortPrototypeInExecutableInstanceRef	IdsmTimestampProviderMapping. pPortPrototypeInExecutable
TimestampProviderMapping. IdsmPPortPrototypeInExecutableInstanceRef	IdsmContextProviderMapping. pPortPrototypeInExecutable

Table 10.1: Correspondence between APMC (M1) elements and MetaModel (M2) elements for FunctionalCluster IdsmInstantiation

10.2 Default Values

This functional cluster does not define any default values for attributes specified in [10].

10.3 Semantic Constraints

This section defines semantic constraints for elements specified in [10] that are part of the configuration model of this functional cluster.

[SWS_AIDSM_CONSTR_00001] Configurable Namespace for IdsmAbstractPortInterface ⌈IdsmAbstractPortInterface.namespace shall never exist.⌋

A Mentioned Manifest Elements

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

This chapter is generated.

Class	**CryptoKeySlot**			
Note	This meta-class represents the ability to define a concrete key to be used for a crypto operation. **Tags:** atp.ManifestKind=MachineManifest This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CryptoProvider.keySlot			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
algorithm Description	CryptoAlgorithm Description	*	aggr	This aggregation contains the collection of crypto algorithm descriptions that can be used in the context of the enclosing crypto key slot. **Tags:** atp.Splitkey=algorithmDescription atp.Status=candidate
cryptoKeySlot Design	CryptoKeySlotDesign	0..1	ref	This reference identifies the CryptoKeySlotDesign from which the referencing CryptoKeySlot was derived. **Tags:** atp.Splitkey=cryptoKeySlotDesign

Table A.1: CryptoKeySlot

Class	**DiagnosticEvent**			
Note	This element is used to configure DiagnosticEvents. **Tags:** atp.recommendedPackage=DiagnosticEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
associated Event Identification	PositiveInteger	0..1	attr	This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.
clearEvent Allowed Behavior	DiagnosticClearEvent AllowedBehaviorEnum	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback

△

Class	**DiagnosticEvent**			
confirmation Threshold	PositiveInteger	0..1	attr	This attribute defines the number of operation cycles with a failed result before a confirmed DTC is set to 1. The semantic of this attribute is a by "1" increased value compared to the confirmation threshold of the "trip counter" mentioned in ISO 14229-1 in figure D.4. A value of "1" defines the immediate confirmation of the DTC along with the first reported failed. This is also sometimes called "zero trip DTC". A value of "2" defines a DTC confirmation in the operation cycle after the first occurred failed. A value of "2" is typically used in the US for OBD DTC confirmation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
connected Indicator	DiagnosticConnected Indicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connectedIndicator, connected Indicator.shortName, connectedIndicator.variation Point.shortLabel vh.latestBindingTime=postBuild
prestorage FreezeFrame	Boolean	0..1	attr	This attribute describes whether the Prestorage of Freeze Frames is supported by the assigned event or not. true: Prestorage of FreezeFrames is supported fFalse: Prestorage of FreezeFrames is not supported
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm)
recoverableIn SameOperation Cycle	Boolean	0..1	attr	If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.

Table A.2: DiagnosticEvent

Class	**DiagnosticEventToSecurityEventMapping**			
Note	This meta-class represents the ability to map a security event that is defined in the context of the Security Extract to a diagnostic event defined on the context of the DiagnosticExtract. **Tags:** atp.Status=candidate atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.3: DiagnosticEventToSecurityEventMapping

Class	IdsFunctionalClusterDesign			
Note	This meta-class defines the intrusion detection system configuration settings in the MachineDesign. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *AbstractFunctionalClusterDesign*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	MachineDesign.functionalClusterDesign			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
network Interface	PlatformModule EthernetEndpoint Configuration	*	ref	This association contains the network configuration that shall be applied to an IDS entity. **Tags:** atp.Splitkey=networkInterface

Table A.4: IdsFunctionalClusterDesign

Class	***IdsPlatformInstantiation*** (abstract)			
Note	This meta-class acts as an abstract base class for platform modules that implement the intrusion detection system. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *AdaptiveModuleInstantiation*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *NonOsModuleInstantiation*, *Referrable*			
Subclasses	IdsmModuleInstantiation			
Aggregated by	*AtpClassifier*.atpFeature, Machine.moduleInstantiation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
idsDesign	IdsFunctionalCluster Design	0..1	ref	Reference to the Ids Design that this IdsPlatform Instantiation implements. **Tags:** atp.Splitkey=idsDesign
idsTimeBase	TimeBaseResource	*	ref	This reference identifies the applicable time base resource. **Tags:** atp.Splitkey=idsTimeBase

Table A.5: IdsPlatformInstantiation

Class	***IdsmAbstractPortInterface*** (abstract)			
Note	This abstract meta-class acts as a base class for all kinds of PortInterfaces related to security event handling. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Subclasses	IdsmContextProviderInterface, IdsmQualifiedEventReceiverInterface, IdsmReportingModeProvider Interface, IdsmTimestampProviderInterface, SecurityEventReportInterface			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.6: IdsmAbstractPortInterface

Class	IdsmContextProviderInterface
Note	This meta-class provides the ability to define a PortInterface for providing a Context for security events in the context of the intrusion detection system. **Tags:** atp.recommendedPackage=IdsmPortInterfaces This Class is only used by the AUTOSAR Adaptive Platform.

▽

△

Class	IdsmContextProviderInterface			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *IdsmAbstractPortInterface*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.7: IdsmContextProviderInterface

Class	IdsmInstance			
Note	This meta-class provides the ability to create a relation between an EcuInstance and a specific class of filters for security events that apply for all security events reported on the referenced EcuInstance. **Tags:** atp.recommendedPackage=IdsmInstanceToEcuInstanceMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
idsmInstanceId	PositiveInteger	0..1	attr	This attribute is used to provide a source identification in the context of reporting security events..
rateLimitation Filter	IdsmRateLimitation	0..1	ref	This reference identifies the applicable rate limitation filter for all security events on the related EcuInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=rateLimitationFilter, rateLimitationFilter.idsmRateLimitation, rateLimitationFilter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
timestamp Format	String	0..1	attr	The existence of this attribute specifies that the IdsM shall add a timestamp to the QSEv messages it sends onto the network. I.e., if this attribute does not exist, no timestamp shall be added to the QSEv messages. The content of this attribute further specifies the timestamp format as follows: - "AUTOSAR" defines AUTOSAR standardized timestamp format according to the Synchronized Time-Base Manager - Any other string defines a proprietary timestamp format. Note: A string defining a proprietary timestamp format shall be prefixed by a company-specific name fragment to avoid collisions.
trafficLimitation Filter	IdsmTrafficLimitation	0..1	ref	This reference identifies the applicable traffic limitation filter for all security events on the related EcuInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=trafficLimitationFilter, trafficLimitationFilter.idsmTrafficLimitation, trafficLimitationFilter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.8: IdsmInstance

Class	IdsmQualifiedEventReceiverInterface			
Note	This meta-class provides the ability to define a PortInterface for receiving qualified security events in the context of the intrusion detection system. **Tags:** atp.recommendedPackage=IdsmPortInterfaces This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Idsm AbstractPortInterface*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.9: IdsmQualifiedEventReceiverInterface

Class	IdsmRateLimitation			
Note	This meta-class represents the configuration of a rate limitation filter for security events. This means that security events are dropped if the number of events (of any type) processed within a configurable time window is greater than a configurable threshold.			
Base	*ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmProperties.rateLimitationFilter			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxEventsIn Interval	PositiveInteger	0..1	attr	This attribute configures the threshold for dropping security events if the number of all processed security events exceeds the threshold in the respective time interval.
timeInterval	Float	0..1	attr	This attribute configures the length of the time interval in seconds for dropping security events if the number of all processed security events exceeds the configurable threshold within the respective time interval.

Table A.10: IdsmRateLimitation

Class	IdsmReportingModeProviderInterface			
Note	This meta-class provides the ability to define a PortInterface for setting and getting the reporting mode for security events in the context of the intrusion detection system. **Tags:** atp.recommendedPackage=IdsmPortInterfaces This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Idsm AbstractPortInterface*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.11: IdsmReportingModeProviderInterface

Class	IdsmTimestampProviderInterface
Note	This meta-class provides the ability to define a PortInterface for providing a timestamp for security events in the context of the intrusion detection system. **Tags:** atp.recommendedPackage=IdsmPortInterfaces This Class is only used by the AUTOSAR Adaptive Platform.
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Idsm AbstractPortInterface*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*

▽

△

Class	**IdsmTimestampProviderInterface**			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.12: IdsmTimestampProviderInterface

Class	**IdsmTrafficLimitation**			
Note	This meta-class represents the configuration of a traffic limitation filter for Security Events. This means that security events are dropped if the size (in terms of bandwidth) of security events (of any type) processed within a configurable time window is greater than a configurable threshold.			
Base	*ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmProperties.trafficLimitationFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
maxBytesIn Interval	PositiveInteger	0..1	attr	This attribute configures the threshold for dropping security events if the size of all processed security events exceeds the threshold in the respective time interval.
timeInterval	Float	0..1	attr	This attribute configures the length of the time interval in seconds for dropping security events if the size of all processed security events exceeds the configurable threshold within the respective time interval.

Table A.13: IdsmTrafficLimitation

Class	**PPortPrototype**			
Note	Component port providing a certain port interface.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType **Tags:** atp.Splitkey=providedInterface

Table A.14: PPortPrototype

Class	**PlatformModuleEthernetEndpointConfiguration**			
Note	This meta-class defines the attributes for the configuration of a port, protocol type and IP address (local address) of the communication on a VLAN. **Tags:** atp.recommendedPackage=PlatformModuleEndpointConfigurations This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *PlatformModuleEndpointConfiguration*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
communication Connector	EthernetCommunication Connector	0..1	ref	Reference to the CommunicationConnector (VLAN) for which the network configuration is defined. **Tags:** atp.Splitkey=communicationConnector

▽

△

Class	PlatformModuleEthernetEndpointConfiguration			
remoteConfig	RemoteEndpoint Configuration	*	aggr	Defintion of remote addresses of peers. **Tags:** atp.Splitkey=remoteConfig
secureCom PropsForTcp	SecureComProps	0..1	ref	Reference to communication security configuration settings that are valid for the tcp unicast endpoint (Tcp Port + unicast IP Address) defined by the PlatformModule EthernetEndpointConfiguration. **Tags:** atp.Splitkey=secureComPropsForTcp
secureCom PropsForUdp	SecureComProps	0..1	ref	Reference to communication security configuration settings that are valid for the udp unicast endpoint (Udp Port + unicast IP Address) defined by the PlatformModule EthernetEndpointConfiguration. **Tags:** atp.Splitkey=secureComPropsForUdp
tcpPort	ApApplicationEndpoint	0..1	ref	This reference allows to configure a tcp port number. **Tags:** atp.Splitkey=tcpPort
udpPort	ApApplicationEndpoint	0..1	ref	This reference allows to configure a udp port number. **Tags:** atp.Splitkey=udpPort

Table A.15: PlatformModuleEthernetEndpointConfiguration

Class	*PortInterface* (abstract)			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractRawDataStreamInterface*, *AbstractSuspendToRamInterface*, *AbstractSynchronizedTimeBase Interface*, ClientServerInterface, *CryptoInterface*, *DataInterface*, *DiagnosticPortInterface*, FirewallState SwitchInterface, *IdsmAbstractPortInterface*, LogAndTraceInterface, ModeSwitchInterface, Network ManagementPortInterface, *PersistencyInterface*, *PlatformHealthManagementInterface*, ServiceInterface, *StateManagementPortInterface*, TriggerInterface			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
namespace (ordered)	SymbolProps	*	aggr	This represents the SymbolProps used for the definition of a hierarchical namespace applicable for the generation of code artifacts out of the definition of a ServiceInterface. This Attribute is only used by the AUTOSAR Adaptive Platform.

Table A.16: PortInterface

Class	*PortPrototype* (abstract)			
Note	Base class for the ports of an AUTOSAR software component. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this `PortPrototype` with respect to client/server communication. **Tags:** atp.Splitkey=clientServerAnnotation

▽

△

Class	***PortPrototype*** (abstract)			
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port. **Tags:** atp.Splitkey=delegatedPortAnnotation
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	aggr	Annotations on this IO Hardware Abstraction port. **Tags:** atp.Splitkey=ioHwAbstractionServerAnnotation
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port. **Tags:** atp.Splitkey=modePortAnnotation
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port. **Tags:** atp.Splitkey=nvDataPortAnnotation
parameterPort Annotation	ParameterPort Annotation	*	aggr	Annotations on this parameter port. **Tags:** atp.Splitkey=parameterPortAnnotation
portPrototype Props	PortPrototypeProps	0..1	aggr	This attribute allows for the definition of further qualification of the semantics of a PortPrototype. **Tags:** atp.Splitkey=portPrototypeProps This Attribute is only used by the AUTOSAR Adaptive Platform.
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication. **Tags:** atp.Splitkey=senderReceiverAnnotation
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port. **Tags:** atp.Splitkey=triggerPortAnnotation

Table A.17: PortPrototype

Class	**Process**			
Note	This meta-class provides information required to execute the referenced `Executable`. **Tags:** atp.recommendedPackage=Processes This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AbstractExecutionContext*, *AtpClassifier*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDeploymentElement*, *Uploadable PackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
design	ProcessDesign	0..1	ref	This reference represents the identification of the design-time representation for the Process that owns the reference. **Tags:** atp.Splitkey=design
executable	Executable	*	ref	Reference to executable that is executed in the process. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=executable
functionCluster Affiliation	String	0..1	attr	This attribute specifies which functional cluster the Process is affiliated with.
numberOf RestartAttempts	PositiveInteger	0..1	attr	This attribute defines how often a process shall be restarted if the start fails. numberOfRestartAttempts = “0” OR Attribute not existing, start once numberOfRestartAttempts = “1”, start a second time

▽

△

Class	Process			
posixUser	PosixUser	0..1	ref	This reference identifies the POSIX user ID under which the Process shall be launched. **Tags:** atp.Splitkey=posixUser
preMapping	Boolean	0..1	attr	This attribute describes whether the executable is preloaded into the memory.
primaryGroup	PosixGroup	0..1	ref	This reference identifies the collection of primary group IDs under which the Process shall be launched. **Tags:** atp.Splitkey=primaryGroup
processState Machine	ModeDeclarationGroup Prototype	0..1	aggr	Set of Process States that are defined for the process. This attribute is used to support the modeling of execution dependencies that utilize the condition of process state. Please note that the process states may not be modeled arbitrarily at any stage of the AUTOSAR workflow because the supported states are standardized in the context of the SWS Execution Management [11]. **Tags:** atp.Splitkey=processStateMachine, processState Machine.shortName
stateDependent StartupConfig	StateDependentStartup Config	*	aggr	Applicable startup configurations. **Tags:** atp.Splitkey=stateDependentStartupConfig
supplementary Group	PosixGroup	*	ref	This reference identifies the collection of supplementary group IDs under which the Process shall be launched. **Tags:** atp.Splitkey=supplementaryGroup

Table A.18: Process

Class	RPortPrototype			
Note	Component port requiring a certain port interface.			
Base	*ARObject*, *AbstractRequiredPortPrototype*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
required Interface	PortInterface	0..1	tref	The interface that this port requires. **Stereotypes:** isOfType **Tags:** atp.Splitkey=requiredInterface

Table A.19: RPortPrototype

Class	RemoteEndpointConfiguration			
Note	This meta-class is used to define the IP address and port of a peer. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*			
Aggregated by	PlatformModuleEthernetEndpointConfiguration.remoteConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ipv4Address	Ip4AddressString	0..1	attr	remote Unicast or Multicast IPv4 Address
ipv6Address	Ip6AddressString	0..1	attr	remote Unicast or Multicast IPv6 Address
tcpPort	PositiveInteger	0..1	attr	remote tcpPort
udpPort	PositiveInteger	0..1	attr	remote udpPort

Table A.20: RemoteEndpointConfiguration

Class	SecurityEventAggregationFilter			
Note	This meta-class represents the aggregation filter that aggregates all security events occurring within a configured time frame into one (i.e. the last reported) security event.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.aggregation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextData Source	SecurityEventContext DataSourceEnum	0..1	attr	This attributes defines whether the context data of the first or last time-aggregated security event shall be used for the resulting qualified security event.
minimum IntervalLength	TimeValue	0..1	attr	This attribute represents the configuration of the minimum time window in seconds for the aggregation filter.

Table A.21: SecurityEventAggregationFilter

Class	*SecurityEventContextMapping* (abstract)			
Note	This meta-class represents the ability to create an association between a collection of security events, an IdsM instance which handles the security events and the filter chains applicable to the security events.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *Uploadable PackageElement*			
Subclasses	SecurityEventContextMappingApplication, SecurityEventContextMappingCommConnector, Security EventContextMappingFunctionalCluster			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
filterChain	SecurityEventFilter Chain	0..1	ref	This reference defines the filter chain to be applied to each of the referenced security events (depending on the reporting mode). **Stereotypes:** atpVariation **Tags:** atp.Splitkey=filterChain, filterChain.securityEventFilter Chain, filterChain.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
idsmInstance	IdsmInstance	0..1	ref	This reference defines the IdsmInstance onto which the security events are mapped. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=idsmInstance, idsmInstance.idsmInstance, idsmInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
mappedSecurity Event	SecurityEventContext Props	*	aggr	This aggregation represents (through further references) the SecurityEventDefinitions to be mapped to an Idsm Instance with additional mapping-dependent properties. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=mappedSecurityEvent, mappedSecurity Event.shortName, mappedSecurityEvent.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.22: SecurityEventContextMapping

Class	SecurityEventContextProps
Note	This meta-class specifies the SecurityEventDefinition to be mapped to an IdsmInstance and adds mapping-dependent properties of this security event valid only for this specific mapping.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	**SecurityEventContextProps**			
Aggregated by	*SecurityEventContextMapping*.mappedSecurityEvent			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
default ReportingMode	SecurityEventReporting ModeEnum	0..1	attr	This attribute defines the default reporting mode for the referenced security event.
persistent Storage	Boolean	0..1	attr	This attribute controls whether qualified reportings of the referenced security event shall be stored persistently by the mapped IdsmInstance or not.
securityEvent	SecurityEventDefinition	0..1	ref	This reference defines the security event that is mapped and enriched by SecurityEventMappingProps with mapping dependent properties. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEvent, securityEvent.securityEvent Definition, securityEvent.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
sensorInstance Id	PositiveInteger	0..1	attr	This attribute defines the ID of the security sensor that detects the referenced security event.
severity	PositiveInteger	0..1	attr	This attribute defines how critical/severe the referenced security event is. Please note that currently, the severity level meanings of specific integer values is not specified by AUTOSAR but left to the party responsible for the IDS system design (e.g. the OEM).

Table A.23: SecurityEventContextProps

Class	**SecurityEventDefinition**			
Note	This meta-class defines a security-related event as part of the intrusion detection system. **Tags:** atp.recommendedPackage=SecurityEventDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
eventSymbol Name	SymbolProps	0..1	aggr	This aggregation defines optionally an alternative Event Name for the SecurityEventDefinition in case there is a collision of shortNames. **Tags:** atp.Splitkey=eventSymbolName, eventSymbol Name.shortName
id	PositiveInteger	0..1	attr	This attribute represents the numerical identification of the defined security event. The identification shall be unique within the scope of the IDS.
securityEvent ContextData Definition	SecurityEventContext DataDefinition	*	ref	Definition of additional context data that is reported with the security event in order to better support the analysis of a possible security threat. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=securityEventContextDataDefinition, security EventContextDataDefinition.securityEventContextData Definition, securityEventContextDataDefinition.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table A.24: SecurityEventDefinition

Class	**SecurityEventFilterChain**			
Note	This meta-class represents a configurable chain of filters used to qualify security events. The different filters of this filter chain are applied in the follow order: SecurityEventStateFilter, SecurityEventOneEvery NFilter, SecurityEventAggregationFilter, SecurityEventThresholdFilter. **Tags:** atp.recommendedPackage=SecurityFilterChains			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
aggregation	SecurityEvent AggregationFilter	0..1	aggr	This aggregation represents the aggregation filter in the filter chain. **Tags:** atp.Splitkey=aggregation, aggregation.shortName
oneEveryN	SecurityEventOneEvery NFilter	0..1	aggr	This aggregation represents the sampling filter in the filter chain. **Tags:** atp.Splitkey=oneEveryN, oneEveryN.shortName
state	SecurityEventStateFilter	0..1	aggr	This aggregation represents the state filter in the event chain. **Tags:** atp.Splitkey=state, state.shortName
threshold	SecurityEventThreshold Filter	0..1	aggr	This aggregation represents the threshold filter in the filter chain. **Tags:** atp.Splitkey=threshold, threshold.shortName

Table A.25: SecurityEventFilterChain

Class	**SecurityEventOneEveryNFilter**			
Note	This meta-class represents the configuration of a sampling (i.e. every n-th event is sampled) filter for security events.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.oneEveryN			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
n	PositiveInteger	0..1	attr	This attribute represents the configuration of the sampling filter, i.e. it configures the parameter "n" that controls how many events (n-1) shall be dropped after a sampled event until a new sample is created.

Table A.26: SecurityEventOneEveryNFilter

Class	**SecurityEventReportInterface**			
Note	This meta-class provides the ability to define a PortInterface for the reporting of security events in the context of the intrusion detection system. **Tags:** atp.recommendedPackage=SecurityEventReportInterfaces This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Idsm AbstractPortInterface*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table A.27: SecurityEventReportInterface

Class	**SecurityEventStateFilter**			
Note	This meta-class represents the configuration of a state filter for security events. The referenced states represent a block list, i.e. the security events are dropped if the referenced state is the active state in the relevant state machine (which depends on whether the IdsM instance runs on the Classic or the Adaptive Platform).			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.state			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
blockIfState ActiveAp	ModeDeclaration	*	iref	For the AP, this reference defines the machine states of the block list. That means, if a security event (mapped to the filter chain to which the SecurityEventStateFilter belongs to) is reported when the machine is in one of the block listed states, the IdsM shall discard the reported security event. **Tags:** atp.Splitkey=blockIfStateActiveAp, blockIfState ActiveAp.contextModeDeclarationGroupPrototype, blockIf StateActiveAp.targetModeDeclaration **InstanceRef implemented by:** FunctionGroupStateIn FunctionGroupSetInstanceRef This Attribute is only used by the AUTOSAR Adaptive Platform.

Table A.28: SecurityEventStateFilter

Class	**SecurityEventThresholdFilter**			
Note	This meta-class represents the threshold filter that drops (repeatedly at each beginning of a configurable time interval) a configurable number of security events . All subsequently arriving security events (within the configured time interval) pass the filter.			
Base	*ARObject*, *AbstractSecurityEventFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventFilterChain.threshold			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
intervalLength	TimeValue	0..1	attr	This attribute configures the time interval in seconds for one threshold filter operation.
threshold Number	PositiveInteger	0..1	attr	This attribute configures the threshold number, i.e. how many security events in the configured time frame are dropped before subsequent events start to pass the filter.

Table A.29: SecurityEventThresholdFilter

Class	***TimeBaseResource*** (abstract)			
Note	This meta-class represents the attributes of one Time Base Resource for Time Synchronization. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	SynchronizedTimeBaseConsumer, SynchronizedTimeBaseProvider			
Aggregated by	TimeSyncModuleInstantiation.timeBase			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.30: TimeBaseResource

B Mentioned APMC Elements from other Functional Clusters

For the sake of completeness, this chapter contains a set of object tables representing APMC configuration elements mentioned in the context of this document but which are not contained directly in this functional cluster.

This chapter is generated.

C Demands and constraints on Base Software (normative)

This functional cluster defines no demands or constraints for the Base Software on which the AUTOSAR Adaptive Platform is running on (usually a POSIX-compatible operating system).

D Platform Extension Interfaces (normative)

This functional cluster does not specify any Platform Extension Interface.

E Not implemented requirements

This functional cluster implements all functional requirements specified in the corresponding requirement specifications.

F History of Constraints and Specification Items

F.1 Constraint and Specification Item History of this document according to AUTOSAR Release R22-11

F.1.1 Added Specification Items in R22-11

none

F.1.2 Changed Specification Items in R22-11

[SWS_AIDSM_01401] [SWS_AIDSM_10101] [SWS_AIDSM_10201] [SWS_AIDSM_10202] [SWS_AIDSM_10203] [SWS_AIDSM_10301] [SWS_AIDSM_10302] [SWS_AIDSM_10303] [SWS_AIDSM_10304] [SWS_AIDSM_10305] [SWS_AIDSM_20101]

F.1.3 Deleted Specification Items in R22-11

[SWS_IdsM_91015]

F.2 Constraint and Specification Item History of this document according to AUTOSAR Release R23-11

F.2.1 Added Specification Items in R23-11

[SWS_AIDSM_01202] [SWS_AIDSM_01203] [SWS_AIDSM_01501] [SWS_AIDSM_01502] [SWS_AIDSM_10204] [SWS_AIDSM_10205] [SWS_AIDSM_10400] [SWS_AIDSM_10401] [SWS_AIDSM_10402] [SWS_AIDSM_10403] [SWS_AIDSM_10404] [SWS_AIDSM_10405] [SWS_AIDSM_10406] [SWS_AIDSM_10407] [SWS_AIDSM_10408] [SWS_AIDSM_10409] [SWS_AIDSM_10500] [SWS_AIDSM_10501] [SWS_AIDSM_10502] [SWS_AIDSM_10503] [SWS_AIDSM_10504] [SWS_AIDSM_10505] [SWS_AIDSM_10506] [SWS_AIDSM_10507] [SWS_AIDSM_10508] [SWS_AIDSM_10509]

F.2.2 Changed Specification Items in R23-11

[SWS_AIDSM_00101] [SWS_AIDSM_00201] [SWS_AIDSM_00202] [SWS_AIDSM_00301] [SWS_AIDSM_00302] [SWS_AIDSM_00303] [SWS_AIDSM_00304] [SWS_AIDSM_00305] [SWS_AIDSM_00306] [SWS_AIDSM_00401] [SWS_AIDSM_00501] [SWS_AIDSM_00502] [SWS_AIDSM_00600] [SWS_AIDSM_00601] [SWS_AIDSM_00602] [SWS_AIDSM_00603] [SWS_AIDSM_00604] [SWS_AIDSM_00605] [SWS_AIDSM_00606] [SWS_AIDSM_00607] [SWS_AIDSM_00701] [SWS_AIDSM_00702]

[SWS_AIDSM_00804] [SWS_AIDSM_01301] [SWS_AIDSM_10101] [SWS_AIDSM_10201] [SWS_AIDSM_10202] [SWS_AIDSM_10203]

F.2.3 Deleted Specification Items in R23-11

[SWS_AIDSM_00807] [SWS_AIDSM_20101]

F.3 Constraint and Specification Item History of this document according to AUTOSAR Release R24-11

F.3.1 Added Specification Items in R24-11

[SWS_AIDSM_00102] [SWS_AIDSM_00103] [SWS_AIDSM_00203] [SWS_AIDSM_00204] [SWS_AIDSM_00608] [SWS_AIDSM_00903] [SWS_AIDSM_00904] [SWS_AIDSM_01204] [SWS_AIDSM_01205] [SWS_AIDSM_01503] [SWS_AIDSM_01601] [SWS_AIDSM_10206] [SWS_AIDSM_10207] [SWS_AIDSM_10306] [SWS_AIDSM_10307] [SWS_AIDSM_10600] [SWS_AIDSM_10601] [SWS_AIDSM_10602] [SWS_AIDSM_10603] [SWS_AIDSM_10702] [SWS_AIDSM_10703] [SWS_AIDSM_10704] [SWS_AIDSM_10705] [SWS_AIDSM_10706] [SWS_AIDSM_10707] [SWS_AIDSM_10708] [SWS_AIDSM_10709] [SWS_AIDSM_10710] [SWS_AIDSM_10711] [SWS_AIDSM_10712] [SWS_AIDSM_10800] [SWS_AIDSM_10801] [SWS_AIDSM_10802] [SWS_AIDSM_10803] [SWS_AIDSM_10804] [SWS_AIDSM_10805] [SWS_AIDSM_10806] [SWS_AIDSM_10807] [SWS_AIDSM_10808] [SWS_AIDSM_10809] [SWS_AIDSM_10810]

F.3.2 Changed Specification Items in R24-11

[SWS_AIDSM_00600] [SWS_AIDSM_00901] [SWS_AIDSM_01201] [SWS_AIDSM_10201] [SWS_AIDSM_10204] [SWS_AIDSM_10301] [SWS_AIDSM_10302] [SWS_AIDSM_10303] [SWS_AIDSM_10304] [SWS_AIDSM_10305] [SWS_AIDSM_10401] [SWS_AIDSM_10402] [SWS_AIDSM_10403] [SWS_AIDSM_10404] [SWS_AIDSM_10405] [SWS_AIDSM_10406] [SWS_AIDSM_10407] [SWS_AIDSM_10408] [SWS_AIDSM_10409] [SWS_AIDSM_10501] [SWS_AIDSM_10502] [SWS_AIDSM_10503] [SWS_AIDSM_10504] [SWS_AIDSM_10505] [SWS_AIDSM_10506] [SWS_AIDSM_10507] [SWS_AIDSM_10508] [SWS_AIDSM_10509]

F.3.3 Deleted Specification Items in R24-11

[SWS_AIDSM_00101] [SWS_AIDSM_01302] [SWS_AIDSM_01303] [SWS_AIDSM_01502]

F.3.4 Added Constraints in R24-11

[SWS_AIDSM_CONSTR_00001]

F.3.5 Changed Constraints in R24-11

none

F.3.6 Deleted Constraints in R24-11

none

F.4 Constraint and Specification Item History of this document according to AUTOSAR Release R25-11

F.4.1 Added Specification Items in R25-11

[SWS_AIDSM_00703] [SWS_AIDSM_01003] [SWS_AIDSM_01004] [SWS_AIDSM_01206] [SWS_AIDSM_02001] [SWS_AIDSM_20000] [SWS_AIDSM_20001]

F.4.2 Changed Specification Items in R25-11

none

F.4.3 Deleted Specification Items in R25-11

none

F.4.4 Added Constraints in R25-11

none

F.4.5 Changed Constraints in R25-11

none

F.4.6 Deleted Constraints in R25-11

none

F.5 Constraint and Specification Item History of this document according to AUTOSAR Release R26-11

F.5.1 Added Specification Items in R26-11

[APMC_AIDSM_00001] [APMC_AIDSM_00002] [APMC_AIDSM_00003] [APMC_AIDSM_00005] [APMC_AIDSM_00007] [APMC_AIDSM_00008] [APMC_AIDSM_00011] [APMC_AIDSM_00012] [APMC_AIDSM_00013] [APMC_AIDSM_00014] [APMC_AIDSM_00015] [APMC_AIDSM_00016] [APMC_AIDSM_00018] [APMC_AIDSM_00019] [APMC_AIDSM_00020] [APMC_AIDSM_00023] [APMC_AIDSM_00024] [APMC_AIDSM_00025] [APMC_AIDSM_00026] [APMC_AIDSM_00027] [APMC_AIDSM_00028] [APMC_AIDSM_00029] [APMC_AIDSM_00030] [APMC_AIDSM_00031] [SWS_AIDSM_00003]

F.5.2 Changed Specification Items in R26-11

[SWS_AIDSM_00102] [SWS_AIDSM_00802] [SWS_AIDSM_00901] [SWS_AIDSM_01001] [SWS_AIDSM_01002] [SWS_AIDSM_01003] [SWS_AIDSM_01004] [SWS_AIDSM_01201] [SWS_AIDSM_01202] [SWS_AIDSM_01203] [SWS_AIDSM_01204] [SWS_AIDSM_01205] [SWS_AIDSM_01401] [SWS_AIDSM_01501] [SWS_AIDSM_01601] [SWS_AIDSM_10101] [SWS_AIDSM_10301] [SWS_AIDSM_10401] [SWS_AIDSM_10501] [SWS_AIDSM_10601] [SWS_AIDSM_10801] [SWS_AIDSM_20000] [SWS_AIDSM_20001]

F.5.3 Deleted Specification Items in R26-11

none

F.5.4 Added Constraints in R26-11

none

F.5.5 Changed Constraints in R26-11

none

F.5.6 Deleted Constraints in R26-11

none