Document Title	Security Extract Template
Document Category	Template specification (TPS)
Document UID	980
Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• Various fixes and improvements
2025-11-27	R25-11	AUTOSAR Release Management	• Support context data elements with variable length • Various fixes and improvements
2024-11-27	R24-11	AUTOSAR Release Management	• Reporting Security Events
2023-11-23	R23-11	AUTOSAR Release Management	• No content changes
2022-11-24	R22-11	AUTOSAR Release Management	• Filtering of security events
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 7
1.1 Overview . . . 7
1.2 Document Conventions . . . 8
2 Use Cases 10
2.1 SECXT as Collection and Exchange Format . . . 10
2.2 SECXT as Configuration Format for IdsM . . . 10
2.3 SECXT as Standardization Format . . . 10
3 Conceptual Background 11
3.1 Main Development Phases for an IDS . . . 11
3.1.1 Security Analysis Phase . . . 12
3.1.2 IDS Design Phase . . . 12
3.1.3 IDS Deployment Phase . . . 12
3.1.4 IDS Operational Phase . . . 13
4 Description of Security Extract Modeling 14
4.1 Overview on Main Model Elements . . . 14
4.2 IdsDesign . . . 15
4.3 Definition of Security Event . . . 17
4.3.1 Properties of a Security Event . . . 17
4.3.2 Attributes of Mapped Security Events . . . 18
4.4 Filtering of Security Events . . . 19
4.4.1 Overview on SecurityEventFilterChain . . . 19
4.4.2 SecurityEventStateFilter . . . 21
4.4.2.1 SecurityEventStateFilter for the Classic Platform . . . 23
4.4.2.2 SecurityEventStateFilter for the Adaptive Platform . . . 23
4.4.3 SecurityEventOneEveryNFilter . . . 23
4.4.4 SecurityEventAggregationFilter . . . 24
4.4.5 SecurityEventThresholdFilter . . . 25
4.4.6 Final Qualification of a reported Security Event . . . 26
4.5 Limitation Filters . . . 26
4.5.1 Rate Limitation Filter . . . 27
4.5.2 Traffic Limitation Filter . . . 28
4.6 Overview on Security Event Mappings . . . 30
4.6.1 Mapping of Security Events to an IdsM Instance . . . 31
4.6.1.1 Context Data definition . . . 34
4.6.1.2 Default Reporting Mode definition . . . 37
4.6.1.3 Persistent Storage definition . . . 38
4.6.1.4 Severity Level definition . . . 39
4.6.1.5 Sensor Instance ID definition . . . 39
4.6.2 Mapping of Security Events with BSW Module Context . . . 39
4.6.3 Mapping of Security Events with Functional Cluster Context . . . 41

4.6.4 Mapping of Security Events with Communication Connector Context 42
4.6.5 Mapping of Security Events with Application Context 44
4.7 Configuration of an IdsM Instance 46
4.7.1 Attributes of an IdsM Instance 48
4.7.1.1 Instance ID of IdsM 48
4.7.1.2 Timestamp in QSEv messages 48
4.7.1.3 Signature Support in QSEv Messages 49
4.7.2 Association of Security Events with an IdsM Instance 50
4.7.3 Network Configuration of an IdsM instance 51
4.7.3.1 IdsM Network Configuration on Classic Platform 51
4.7.3.2 IdsM Network Configuration on Adaptive Platform 52
4.7.4 Block States of an IdsM instance on CP 53
4.8 Reporting Security Events 54
4.8.1 Security Event Context Data Identification 54
A Reference Material 61
A.1 Terms and Abbreviations 61
A.2 Imposition Times of Constraints 62
B Security Events in Use 63
C Examples 69
C.1 Example for modeling of SecurityEventContextDataDefinition . 69
D Mentioned Class Tables 72
E Upstream Mapping 86
E.1 Firewall 86
E.2 IdsM 87
E.3 KeyM 102
E.4 TcpIp 102
F Non-Splitable Elements in the Scope of this Document 103
G Variation Points in the Scope of this Document 104
H Change History 105
H.1 Change History of this document according to AUTOSAR Release R20-11 105
H.1.1 Added Specification Items in R20-11 105
H.1.2 Changed Specification Items in R20-11 106
H.1.3 Deleted Specification Items in R20-11 106
H.1.4 Added Constraints in R20-11 107
H.1.5 Changed Constraints in R20-11 107
H.1.6 Deleted Constraints in R20-11 107
H.2 Change History of this document according to AUTOSAR Release R22-11 107
H.2.1 Added Specification Items in R22-11 107
H.2.2 Changed Specification Items in R22-11 108
H.2.3 Deleted Specification Items in R22-11 108

H.2.4 Added Constraints in R22-11 . . . 108
H.2.5 Changed Constraints in R22-11 . . . 108
H.2.6 Deleted Constraints in R22-11 . . . 108
H.3 Change History of this document according to AUTOSAR Release R23-11 108
H.3.1 Added Specification Items in R23-11 . . . 108
H.3.2 Changed Specification Items in R23-11 . . . 108
H.3.3 Deleted Specification Items in R23-11 . . . 108
H.3.4 Added Constraints in R23-11 . . . 108
H.3.5 Changed Constraints in R23-11 . . . 109
H.3.6 Deleted Constraints in R23-11 . . . 109
H.3.7 Added Advisories in R23-11 . . . 109
H.3.8 Changed Advisories in R23-11 . . . 109
H.3.9 Deleted Advisories in R23-11 . . . 109
H.4 Change History of this document according to AUTOSAR Release R24-11 109
H.4.1 Added Specification Items in R24-11 . . . 109
H.4.2 Changed Specification Items in R24-11 . . . 110
H.4.3 Deleted Specification Items in R24-11 . . . 110
H.4.4 Added Constraints in R24-11 . . . 110
H.4.5 Changed Constraints in R24-11 . . . 111
H.4.6 Deleted Constraints in R24-11 . . . 111
H.4.7 Added Advisories in R24-11 . . . 111
H.4.8 Changed Advisories in R24-11 . . . 111
H.4.9 Deleted Advisories in R24-11 . . . 111
H.5 Change History of this document according to AUTOSAR Release R25-11 111
H.5.1 Added Specification Items in R25-11 . . . 111
H.5.2 Changed Specification Items in R25-11 . . . 111
H.5.3 Deleted Specification Items in R25-11 . . . 111
H.5.4 Added Constraints in R25-11 . . . 112
H.5.5 Changed Constraints in R25-11 . . . 112
H.5.6 Deleted Constraints in R25-11 . . . 112
H.6 Change History of this document according to AUTOSAR Release R26-11 113
H.6.1 Added Specification Items in R26-11 . . . 113
H.6.2 Changed Specification Items in R26-11 . . . 113
H.6.3 Deleted Specification Items in R26-11 . . . 113
H.6.4 Added Constraints in R26-11 . . . 113
H.6.5 Changed Constraints in R26-11 . . . 114
H.6.6 Deleted Constraints in R26-11 . . . 114
H.6.7 Added Advisories in R26-11 . . . 114
H.6.8 Changed Advisories in R26-11 . . . 114
H.6.9 Deleted Advisories in R26-11 . . . 114

References

[1] Specification of Intrusion Detection System Manager
AUTOSAR_CP_SWS_IntrusionDetectionSystemManager

[2] Diagnostic Extract Template
AUTOSAR_CP_TPS_DiagnosticExtractTemplate

[3] System Template
AUTOSAR_CP_TPS_SystemTemplate

[4] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

[5] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[6] Standardized M1 Models used for the Definition of AUTOSAR
AUTOSAR_FO_MOD_GeneralDefinitions

[7] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate

[8] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[9] Specification of Basic Software Mode Manager
AUTOSAR_CP_SWS_BSWModeManager

[10] Specification of ECU Configuration
AUTOSAR_CP_TPS_ECUConfiguration

[11] Glossary
AUTOSAR_FO_TR_Glossary

[12] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate

1 Introduction

1.1 Overview

The Security Extract Template (SECXT) is part of the Intrusion Detection System (IDS). The elements of an IDS are described in the document SWS_IntrusionDetectionSystemManager [1]. In the context of ECU development projects, the SECXT serves multiple use cases that are described in Chapter 2.

The Intrusion Detection System Manager (IdsM) is a Basic Software module (for the AUTOSAR Classic Platform) or a Platform Service (for the AUTOSAR Adaptive Platform) that collects and centrally aggregates security incidents that possibly result from malicious attacks on the vehicle's software, communications or electronics system. In each of the security relevant ECUs or machines within the vehicle, an instance of the IdSM module or service collects and filters security events (optionally including additional data) in order to store them in a local Security Event Memory (Sem) and/or to forward them over the vehicle network to a central Intrusion Detection System Reporter (IdsR). This IdsR might be, for example, located within a telematics unit enabling it to send security reports and associated data via a cellular network to an OEM's Security Operations Center (SOC). This information is then analyzed by the Security Incident and Event Management (SIEM) and, if necessary, used to develop and decide on appropriate defense or mitigation actions to counter the attack.

The SECXT specifies the security events and their properties for a vehicle on system level. Similar to the Diagnostic Extract [2], it extends the System Template [3] and the Manifest [4] to enable a formal exchange of security event definitions among an OEM and its various suppliers. The Security Extract as a specific, "standalone" file for security event definitions is in particular useful in view of the reasonable expectation that new approaches or kinds of attacks are identified after SOP of a vehicle. The resulting new or changed security events lead to an updated SECXT file that can subsequently be deployed onto the affected ECUs or machines of a vehicle together with a software update. Additionally, the SECXT file can potentially be used by the SIEM and SOC to interpret incoming reports of the IdsR instances of the vehicles in field.

To summarize, the Security Extract Template defines a standardized AUTOSAR exchange format for defining security events and their properties. The Security Extract (SECXT) is formalized as an ARXML file and applicable for both the AUTOSAR Adaptive and AUTOSAR Classic Platforms in a way similar to a Diagnostic Extract file.

1.2 Document Conventions

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The headers in the table have the following meaning:

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

Type: The type of an attribute of the class.

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

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

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

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

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see [5, Standardization Template].

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

2 Use Cases

The Security Extract primarily serves as collection and exchange format for definition of security events and their system-related properties. Additionally, the SECXT can be used to specify instances of the IdsM module and their system-level configurations.

The Security Extract Template has been defined in a way that makes it applicable to both the Classic and the Adaptive Platform of AUTOSAR at the same time. That means, the same Security Extract file can contain definitions that can be applied to an IdsM running on Classic Platform as well as on an IdsM running on Adaptive Platform.

Furthermore, the SECXT is also used in the context of AUTOSAR standardization as collection format for the *standardized security events*.

2.1 SECXT as Collection and Exchange Format

During the development of an ECU, the security aspects have also to be taken into account due to new legislative regulations (“Cybersecurity Engineering”). This security engineering process is usually carried out in parallel to the functional development process and usually also leads to identification of possible *indicators* for specific threats that, later in the field, shall be identified, filtered and, if necessary, sent as *qualified security events* (QSEv) via the IdsR to a central SIEM for further analysis and handling.

An IdsR, a SIEM or any other entity that needs information about security events can potentially also use Security Extract files as input for configuration of the security events it needs to handle.

2.2 SECXT as Configuration Format for IdsM

A part of the Intrusion Detection System standardized by AUTOSAR, the Security Extract Template contains additional elements to specify IdsM instances and their system-level properties such as provisioning of timestamp or authentication (i.e. signature) information in the QSEv messages to be sent to the IdsR.

2.3 SECXT as Standardization Format

The standardized security events for a subset of BSW modules (Classic Platform) and Functional Clusters (Adaptive Platform) are defined within the ARXML file AUTOSAR_MOD_GeneralDefinition_SecurityEvents.arxml which is based on the Security Extract Template and distributed as part of AUTOSAR_MOD_GeneralDefinitions.zip.

3 Conceptual Background

In this chapter, further background information on the overall concept of the `Security Extract` file format is given to create a better basis for understanding the meta-model described in Chapter 4.

3.1 Main Development Phases for an IDS

Typically, an `Intrusion Detection System` (IDS) is based on the system parts IdsM, IdsR and the Security Operation Center (SOC) as exemplarily depicted in Figure 3.1.

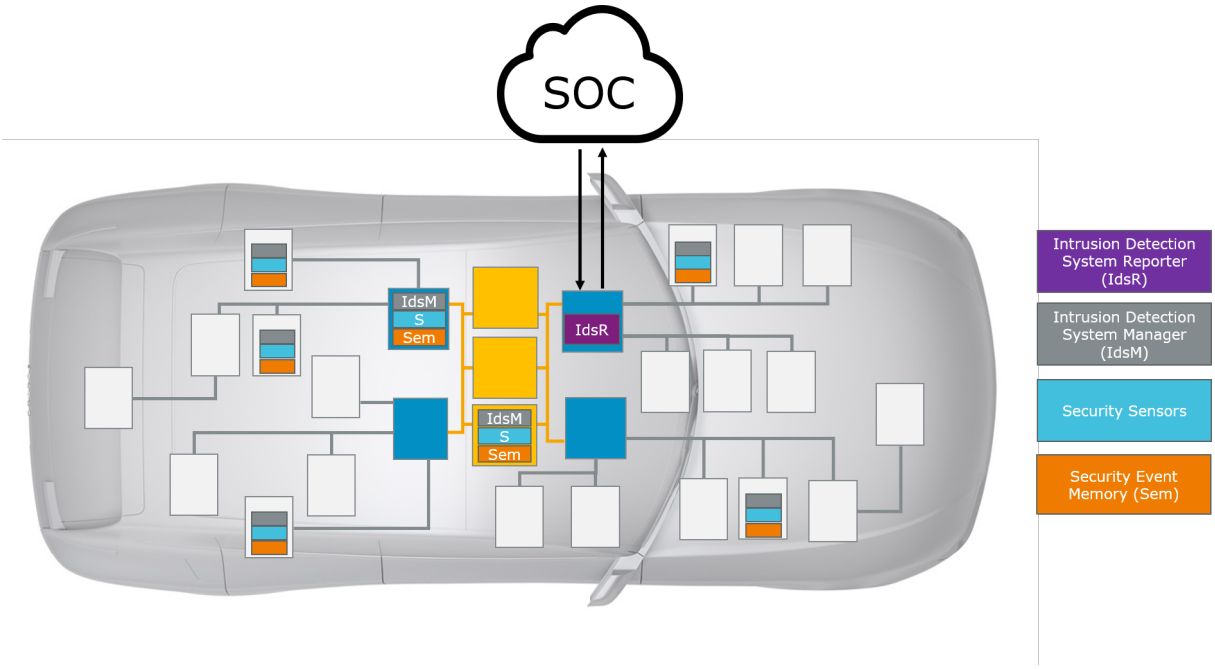


Figure 3.1: Architecture of a distributed `Intrusion Detection System`

The development of such an IDS can be divided into the following main phases:

1. Security Analysis phase
2. IDS Design phase
3. IDS Deployment phase
4. IDS Operational phase

The `Security Extract Template` supports all these four phases and can both be used for specification and exchange of IDS related definitions by and between OEMs and their suppliers. Therefore, a `Security Extract` file has potentially a high number of release cycles starting with security analysis and ending with “end of support” for a specific vehicle.

3.1.1 Security Analysis Phase

In the *Security Analysis* phase, the vehicle's electronics and software system is examined and analyzed by security experts to identify and evaluate potential approaches of attacks on the components of the system that could lead to a security breach. In a second step, based on these potential attack approaches, detectable events that deviate from the normal behavior of the system are identified and defined as `Security Events`.

One example of such a security event is the failed check of a CRC within a received End-to-End protected network message. While one occurrence of such a CRC failure would be explained by random transmission error (e.g. electromagnetic interference), a high number of reports of this security event within a short time and, in particular, only for a certain kind of network messages would arouse suspicion of a malicious attack on the network system.

The `Security Extract Template` supports this phase by formalizing the definition of these security events and their attributes (such as the ID). In addition, AUTOSAR also provides standardized security events in `Security Extract` format (as already mentioned in Ch. 2.3).

3.1.2 IDS Design Phase

The *IDS Design* phase distributes, customizes and adapts the generic IDS components towards a concrete vehicle electronics and software system taking into consideration the security events identified in the previous phase. For example, IdsM instances are defined for the relevant ECUs and the respective security events are associated with these IdsM instances together with the definition of filters to prevent, for example, reporting of single and therefore harmless security events (like in the CRC failure example above).

In this phase, the `Security Extract Template` is enriched with the design decisions such as definition of IdsM instances, the mapping of security events onto them and the configuration of filters.

3.1.3 IDS Deployment Phase

The *IDS Deployment* phase comprises the realization of the IDS Design from the previous step towards the real system in hardware and software.

This phase is supported by the `Security Extract Template` through definition of IdsM instance deployment onto specific ECU-HW and the possibility to derive ECU configuration parameters for the IdsM modules on the Classic Platform (i.e. definition of `Upstream Mapping` rules, see also Ch. E).

3.1.4 IDS Operational Phase

The *IDS Operational* phase refers to the running IDS in the field when the vehicle is used by the end customer.

This phase is still regarded as part of the development process because it typically involves an *IDS update process* to keep the IDS up to date with new versions of application and platform software as well as with newly identified attack approaches and thus new security events.

During the *IDS update process*, `Security Extract` files can be used to reconfigure the IdsM instances of the IDS and also to make these reconfigurations known to the IdsR.

This is a notable difference to other AUTOSAR (M2 level) exchange files (e.g. System Description) which usually do not evolve further after the final configuration of the ECU-HW devices of the vehicle has been specified for SOP. On the other hand, the `Security Extract` file is expected to be maintained and further extended even after SOP of the vehicle it relates to due to its involvement in the *IDS update process*.

4 Description of Security Extract Modeling

In this chapter, the meta-model of the `Security Extract Template` is described in detail.

4.1 Overview on Main Model Elements

The `Security Extract Template` comprises the main elements as shown in Figure 4.1.

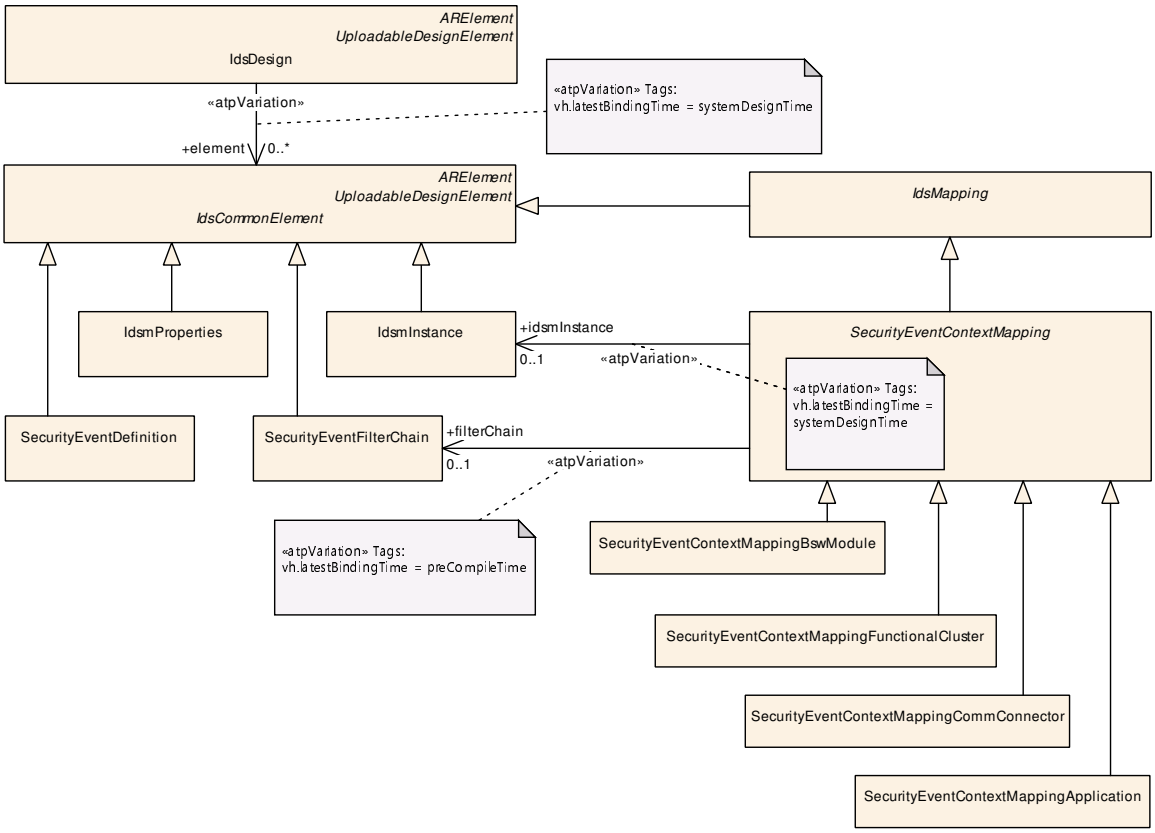


Figure 4.1: Main model elements of the `Security Extract Template`

These elements have the following purposes:

- The `IdsDesign` is the “umbrella” meta-class, i.e. the root element that links together all relevant Security Extract elements to form and define the scope of the IDS under design and to be implemented.
- The abstract meta-class `IdsCommonElement` serves as base class for the Security Extract elements. Its only purpose is to be referenced by the single role `element` of `IdsDesign`.
- The meta-class `SecurityEventDefinition` is derived from `IdsCommonElement` and defines a security event together with its general properties. The

SecurityEventDefinitions can be provided by different parties of a development project in multiple Security Extract files.

- IdsmInstance is derived from IdsCommonElement and specifies an instance of the IdsM together with its system-level configuration parameters.
- IdsmProperties is derived from IdsCommonElement and provides a container for definition of functional properties related to IdsmInstances that can be applied in a re-usable manner by respective referencing. One example is the limitation of network bandwidth created by an IdsM instance.
- SecurityEventFilterChain is derived from IdsCommonElement and defines the applicability and properties of the various kind of filters that can be applied to reported SecurityEventDefinitions. A reported SecurityEventDefinition that has successfully passed the whole filter chain becomes a *qualified security event* (but is still subject to the limitation filters of the IdsM). A specific SecurityEventFilterChain applies to a specific collection of SecurityEventDefinitions as defined by mapping (see Ch. 4.4.1).
- The abstract meta-class IdsMapping is derived from IdsCommonElement and serves as base class for SecurityEventContextMapping and possible additional mapping classes in future releases.
- The abstract meta-class SecurityEventContextMapping derived from IdsMapping serves as base class for the various context dependent mapping definition elements for security events. Its only purpose is to be included into an IdsDesign by being referenced in the role element. The following concrete meta-classes are derived from SecurityEventContextMapping:
  - SecurityEventContextMappingBswModule maps SecurityEventDefinitions to an IdsmInstance defining the executional context of their occurrence within a BSW module.
  - SecurityEventContextMappingFunctionalCluster maps SecurityEventDefinitions to an IdsmInstance defining the executional context of their occurrence within a functional cluster.
  - SecurityEventContextMappingCommConnector maps SecurityEventDefinitions to an IdsmInstance defining the executional context of their occurrence in relation to a CommunicationConnector.
  - SecurityEventContextMappingApplication maps SecurityEventDefinitions to an IdsmInstance defining the executional context of their occurrence within application software.

4.2 IdsDesign

[TPS_SECXT_01043] Semantics of IdsDesign ⌈The meta-class IdsDesign represents a structural container that defines the scope (and thus the system boundaries)

of an IDS design and implementation by linking together (through the references in the role `element` all relevant Security Extract elements.⌋

The `IdsDesign` linking together all relevant Security Extract elements is depicted in Figure 4.1.

Class	**IdsDesign**			
Note	This meta-class represents the root element of a SecurityExtract file for IDS development. It defines the scope of an IDS to be designed and implemented by referencing all SecurityExtract meta-classes that need to be included into the IDS development process. **Tags:** atp.recommendedPackage=IdsDesigns			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
element	IdsCommonElement	*	ref	This reference includes an element with IDS related definitions into the IdsDesign. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.idsCommonElement, element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime

Table 4.1: IdsDesign

Please note that the meta-classes directly referenced by `IdsDesign` also inherit from the generic abstract meta-class `ARElement` and are thus allowed to be instantiated in a self-contained way within any `ARPackage`. This modeling enables the definition and exchange of Security Extract content that is not yet associated with a concrete `IdsDesign` (e.g. `SecurityEventDefinition`s related only to a specific functionality as contribution to an IDS under development). One example of such Security Extract content not related to a concrete `IdsDesign` is the specification of the *AUTOSAR Standardized Security Events* inside the general definitions [6].

4.3 Definition of Security Event

[TPS_SECXT_01001] Semantics of `SecurityEventDefinition` ⌈A `SecurityEventDefinition` represents the atomic unit of a security-related event with predefined properties that is reported by security sensors and further processed by the IdsM.⌋

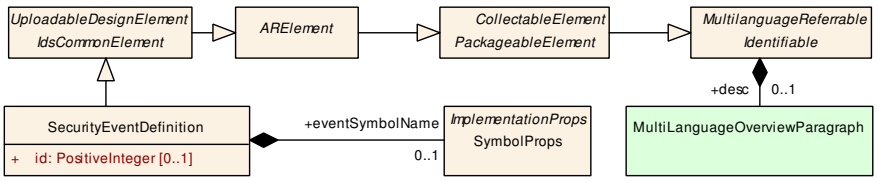


Figure 4.2: Modeling of `SecurityEventDefinition`

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 4.2: SecurityEventDefinition

4.3.1 Properties of a Security Event

[TPS_SECXT_01002] EventName of `SecurityEventDefinition` ⌈A `SecurityEventDefinition` shall be named and referred to by a symbolic EventName composed of upper-case letters and underscore characters with an abbreviated prefix indicating the source BSW module (Classic Platform) or source functional cluster (Adaptive Platform) of the security event (e.g. `KEYM_CERTIFICATE_FAILED`). In a `Security`

Extract, an instance of a SecurityEventDefinition shall use this EventName as its shortName.⌋

[TPS_SECXT_01000] Alternative EventName of SecurityEventDefinition ⌈If SecurityEventDefinitions from different sources are merged and a collision of their shortNames is detected, then the aggregated SymbolProps (in the role eventSymbolName) shall be used to define an alternative EventName for the colliding SecurityEventDefinition. The EventName defined through the role eventSymbolName takes precedence over the EventName defined by the shortName.⌋

An instance of SecurityEventDefinition needs to be uniquely identifiable (i.e. within an IDS scope) by its id:

[TPS_SECXT_01003] Semantics of attribute SecurityEventDefinition.id ⌈The attribute id shall define the numerical value of the SecurityEventDefinition for external identification (i.e. outside the IdsM instance).⌋

[constr_5600] Valid interval for attribute SecurityEventDefinition.id ⌈The valid interval for attribute SecurityEventDefinition.id is 0..65534.⌋

[constr_5601] Uniqueness of SecurityEventDefinition.id ⌈Within the scope of an IDS, i.e. for all SecurityEventDefinitions referenced by the same IdsDesign, there shall be no attribute id of any other SecurityEventDefinition that has the same value.⌋

[constr_12011] Existence of attribute SecurityEventDefinition.id ⌈For each SecurityEventDefinition, the attribute id shall exist.⌋

[TPS_SECXT_01004] Textual description of SecurityEventDefinition ⌈The MultiLanguageOverviewParagraph aggregated in the role desc by a SecurityEventDefinition shall be used for a brief textual description of the security event.⌋

These brief textual descriptions of SecurityEventDefinitions can be collected, for example, into overview tables.

4.3.2 Attributes of Mapped Security Events

Additionally to the general properties of a SecurityEventDefinition described in Ch. 4.3.1, there are additional properties of a SecurityEventDefinition that can only be defined in the concrete context of its use, i.e. in particular, when its mapping to an IdsmInstance has been defined (see Ch. 4.6). The additional properties of a SecurityEventDefinition that are dependent on its mapping are defined by the meta-class SecurityEventContextProps and described in detail in Ch. 4.6.1.

4.4 Filtering of Security Events

In general, reported security events do not immediately become qualified security events but need to pass a set of well-defined condition checks in order to become qualified.

These condition checks are performed in sequence as follows:

- Default reporting mode (see Chapter 4.6.1.2)
- Filter chain (see Chapter 4.4.1)

After qualified security events have successfully passed these security event specific filters, additional filters that are generic for an IdsM Instance are applied in order to reduce possible high rates of incoming security events:

- Limitation filters (see Chapter 4.5)

The reporting mode and filter chain checks (the first two steps documented above) are modeled around the abstract meta-class `SecurityEventContextMapping` affecting only the referenced `SecurityEventDefinition`s while the limitation filters (the third step) are modeled separately because it applies to the whole IdsM Instance with all its `SecurityEventDefinition`s.

[TPS_SECXT_02000] Definition of pre-qualified security events ⌈Pre-qualified security events - which are, for example, reported by a smart security sensor (i.e. with advanced detecting and processing capabilities) and to be immediately processed by the IdsM Instance as qualified security events - shall be defined as security events with an empty filter chain.⌋

Please note that these pre-qualified security events are still subject to limitation filters.

4.4.1 Overview on SecurityEventFilterChain

A `SecurityEventFilterChain` contains the definitions of filtering algorithms that can be applied in a standardized order towards the occurrence of a security event.

[TPS_SECXT_01006] Filtering Semantics of `SecurityEventFilterChain` ⌈A `SecurityEventFilterChain` defines for each of the contained filter algorithms whether this algorithm

- shall be applied with the specified filter algorithm parameters or
- shall not be applied.

The order of application of the contained filter algorithms is standardized.⌋

[TPS_SECXT_01007] Applicability of `SecurityEventFilterChain` towards `SecurityEventDefinition`s ⌈A specific `SecurityEventFilterChain` shall

only be applied to those SecurityEventDefinitions to which this SecurityEventFilterChain is mapped by derived meta-classes of the abstract meta-class SecurityEventContextMapping.

This mapping is described in detail in Chapter 4.6.

Figure 4.3 shows an overview on the modeling of the filter chain for security events.

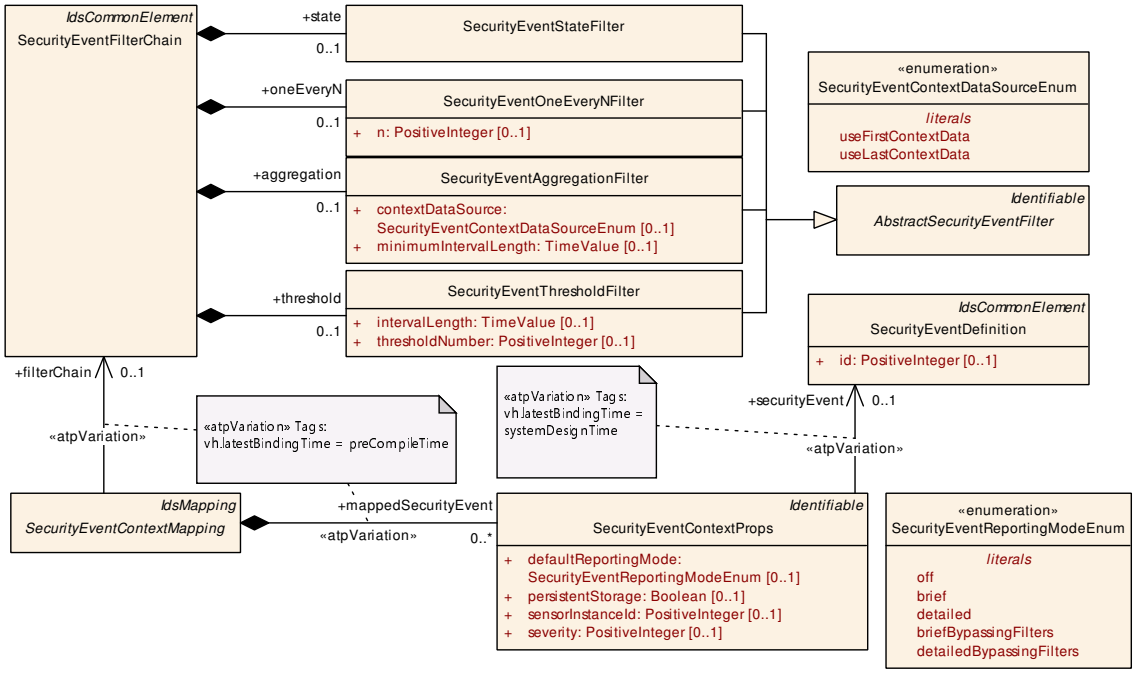


Figure 4.3: Modeling of SecurityEventFilterChain

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

△

Class	**SecurityEventFilterChain**			
threshold	SecurityEventThreshold Filter	0..1	aggr	This aggregation represents the threshold filter in the filter chain. **Tags:** atp.Splitkey=threshold, threshold.shortName

Table 4.3: SecurityEventFilterChain

Note: AbstractSecurityEventFilter serves as abstract meta-class from which concrete meta-classes that represent well-defined filter algorithms are derived. These well-defined filters contribute to the filter chain.

Class	***AbstractSecurityEventFilter*** (abstract)			
Note	This meta-class acts as a base class for security event filters.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	SecurityEventAggregationFilter, SecurityEventOneEveryNFilter, SecurityEventStateFilter, SecurityEvent ThresholdFilter			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.4: AbstractSecurityEventFilter

4.4.2 SecurityEventStateFilter

[TPS_SECXT_01008] Semantics of SecurityEventStateFilter ⌈The SecurityEventStateFilter defines a blocking filter of functionality “State Filter” and is applicable to both the Classic and Adaptive Platform. If any of the referenced states (respectively for CP and AP) is active, then the reported SecurityEvent-Definition shall be discarded by the IdsM. For the Classic Platform, the possible active states are referenced by blockIfStateActiveCp. For the Adaptive Platform, the possible active states are referenced by blockIfStateActiveAp.⌋

Please note that the state machines which indicate the currently active state are defined differently for the Classic and the Adaptive Platform.

[constr_5613] Unambiguous definition of SecurityEventStateFilter for CP or AP ⌈For SecurityEventStateFilter, either the references in the role block-IfStateActiveCp or the references in the role blockIfStateActiveAp shall be defined in order to ensure the unambiguous applicability of the SecurityEvent-StateFilter towards the Classic or the Adaptive Platform.⌋

[constr_5615] Restriction of SecurityEventStateFilter referencing BlockStates on CP ⌈For a SecurityEventStateFilter on the Classic Platform, the references in the role blockIfStateActiveCp shall only reference those BlockStates that are aggregated in the role blockState by the IdsmInstance which is mapped (by SecurityEventContextMapping) to that SecurityEventFilter-Chain of which the SecurityEventStateFilter is part of.⌋

In other words, a SecurityEventStateFilter on Classic Platform shall not reference a BlockState in the role blockIfStateActiveCp if this BlockState does not belong to the IdsmInstance to which the SecurityEventStateFilter applies to (by mapping through the enclosing SecurityEventFilterChain and SecurityEventContextMapping).

Please note that SecurityEventContextMapping additionally defines mappedSecurityEvents. That means that on a given IdsmInstance, a SecurityEventDefinition is always associated (through SecurityEventContextMapping) with none or one specific SecurityEventFilterChain. In the latter case, if SecurityEventStateFilter is part of the SecurityEventFilterChain, the SecurityEventDefinition is in the end mapped to a possibly distinct set of BlockStates with any of these BlockStates - when active - leading to the dropping of the SecurityEventDefinition during filter evaluation.

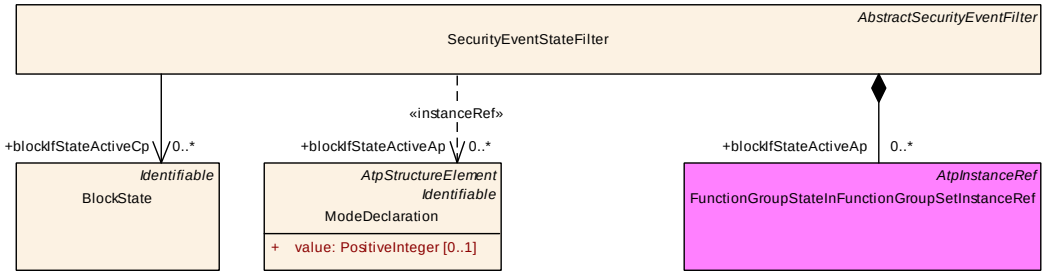


Figure 4.4: Modeling overview of the SecurityEventStateFilter

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.

△

Class	SecurityEventStateFilter			
blockIfState ActiveCp	BlockState	*	ref	For the CP, this reference defines the 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 currently active block state in the IdsM is one of the referenced block listed states, the IdsM shall discard the reported security event. **Tags:** atp.Splitkey=blockIfStateActiveCp This Attribute is only used by the AUTOSAR Classic Platform.

Table 4.5: SecurityEventStateFilter

4.4.2.1 SecurityEventStateFilter for the Classic Platform

[TPS_SECXT_01045] Semantics of `SecurityEventStateFilter` for CP ⌈For the Classic Platform, if a `SecurityEventDefinition`, that is mapped to the `SecurityEventFilterChain` to which the `SecurityEventStateFilter` belongs to, is reported to the IdsM when the currently active `BlockState` in the IdsM matches one of the `BlockState`s referenced in the role `blockIfStateActiveCp`, then the IdsM shall discard the reported `SecurityEventDefinition`.⌋

4.4.2.2 SecurityEventStateFilter for the Adaptive Platform

[TPS_SECXT_01046] Semantics of `SecurityEventStateFilter` for AP ⌈For the Adaptive Platform, if a `SecurityEventDefinition`, that is mapped to the `SecurityEventFilterChain` to which the `SecurityEventStateFilter` belongs to, is reported to the IdsM when the currently active function group state matches one of the function group states referenced in the role `blockIfStateActiveAp`, then the IdsM shall discard the reported `SecurityEventDefinition`.⌋

4.4.3 SecurityEventOneEveryNFilter

[TPS_SECXT_01009] Semantics of `SecurityEventOneEveryNFilter` ⌈`SecurityEventOneEveryNFilter` defines a sampling filter of functionality "Forward Every Nth" with N being defined by the attribute `n`. Every `n`'th security event passes this filter further down the filter chain.⌋

[constr_5602] Valid interval for attribute `SecurityEventOneEveryNFilter.n` ⌈The valid interval for attribute `SecurityEventOneEveryNFilter.n` is 2..65535.⌋

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 4.6: SecurityEventOneEveryNFilter

[constr_12014] Existence of attribute `SecurityEventOneEveryNFilter.n` ⌈For each `SecurityEventOneEveryNFilter`, the attribute `n` shall exist.⌋

4.4.4 SecurityEventAggregationFilter

[TPS_SECXT_01010] Semantics of `SecurityEventAggregationFilter` ⌈`SecurityEventAggregationFilter` defines an accumulating filter of functionality "aggregation filter". It counts for each consecutive time interval `minimumIntervalLength` the number of occurrences of the specific `SecurityEventDefinition`. If at the end of a time interval this number is greater than zero, the resulting aggregated security event containing this number and optional context data is passed further down the filter chain.⌋

[constr_5603] Valid interval for attribute `SecurityEventAggregationFilter.minimumIntervalLength` ⌈The valid interval for attribute `SecurityEventAggregationFilter.minimumIntervalLength` is]0..INF[seconds.⌋

[TPS_SECXT_01011] Semantics of attribute `SecurityEventAggregationFilter.contextDataSource` ⌈The attribute `contextDataSource` defines whether - in case the qualifying condition of the `SecurityEventAggregationFilter` is met - the context data of the first or of the last reported `SecurityEventDefinition` within that time interval shall be attached to the resulting aggregated security event.⌋

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**

▽

△

Class	**SecurityEventAggregationFilter**			
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 4.7: SecurityEventAggregationFilter

[constr_12012] Existence of attribute `SecurityEventAggregationFilter.contextDataSource` ⌈For each `SecurityEventAggregationFilter`, the attribute `contextDataSource` shall exist.⌋

[constr_12013] Existence of attribute `SecurityEventAggregationFilter.minimumIntervalLength` ⌈For each `SecurityEventAggregationFilter`, the attribute `minimumIntervalLength` shall exist.⌋

Enumeration	**SecurityEventContextDataSourceEnum**
Note	This enumeration controls the elements used to creating the resulting qualified security event
Aggregated by	SecurityEventAggregationFilter.contextDataSource
Literal	*Description*
useFirstContext Data	Context data of first received security event shall be used for resulting qualified security event. **Tags:** atp.EnumerationLiteralIndex=0
useLastContext Data	Context data of last received security event shall be used for resulting qualified security event. **Tags:** atp.EnumerationLiteralIndex=1

Table 4.8: SecurityEventContextDataSourceEnum

4.4.5 SecurityEventThresholdFilter

[TPS_SECXT_01012] Semantics of `SecurityEventThresholdFilter` ⌈`SecurityEventThresholdFilter` defines an accumulating filter of functionality “threshold filter”. It discards for each consecutive time interval `intervalLength` the first `thresholdNumber`-1 occurrences of the specific `SecurityEventDefinition`. All subsequently reported security events within the same time interval are passed further down the filter chain.⌋

[constr_5604] Valid interval for attribute `SecurityEventThresholdFilter.intervalLength` ⌈The valid interval for attribute `SecurityEventThresholdFilter.intervalLength` is]0..INF[seconds.⌋

[constr_5605] Valid interval for attribute `SecurityEventThresholdFilter.thresholdNumber` ⌈The valid interval for attribute `SecurityEventThresholdFilter.thresholdNumber` is 2..65535.⌋

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 4.9: SecurityEventThresholdFilter

[constr_12015] Existence of attribute `SecurityEventThresholdFilter.intervalLength` ⌈For each `SecurityEventThresholdFilter`, the attribute `intervalLength` shall exist.⌋

[constr_12016] Existence of attribute `SecurityEventThresholdFilter.thresholdNumber` ⌈For each `SecurityEventThresholdFilter`, the attribute `thresholdNumber` shall exist.⌋

4.4.6 Final Qualification of a reported Security Event

[TPS_SECXT_01013] Final Qualification of a `SecurityEventDefinition` ⌈A reported `SecurityEventDefinition` that is not blocked by the `defaultReportingMode` and that has successfully passed all filters of a `SecurityEventFilterChain` as configured becomes a **qualified security event (QSEv)**.⌋

Note: This QSEv is still subject to limitation filtering (if configured) before it is sent onto the network. Please refer to Chapter 4.5.

4.5 Limitation Filters

Security events might occur in high numbers within a short time. Therefore, limitation filters can be applied if the network bandwidth for sending qualified security event (QSEv) messages needs to be limited in order to not significantly affect the remaining network communication in a negative way.

Since the properties of the limitation filters usually need to be defined dependent on the network connection properties of the ECU on which the IdsM instance runs, the specifically configured limitation filters are associated with an `IdsmInstance` and not with `SecurityEventDefinition`s.

Therefore, the meta-classes representing the limitation filter, IdsmRateLimitation and IdsmTrafficLimitation, are aggregated by IdsmProperties as shown in Figure 4.5.

An IdsmInstance can use specific IdsmRateLimitation and/or IdsmTrafficLimitation filters by referencing one or both of them in the role rateLimitationFilter or trafficLimitationFilter, respectively.

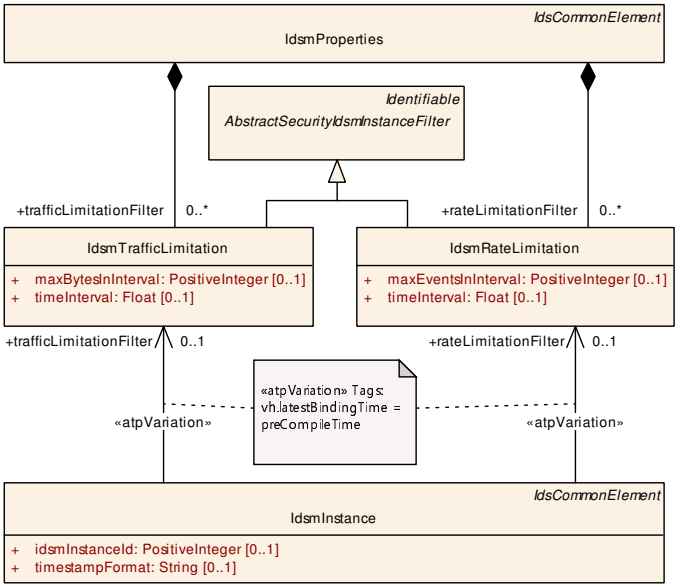


Figure 4.5: Modeling overview on IdsmProperties with filters IdsmRateLimitation and IdsmTrafficLimitation

Class	***AbstractSecurityIdsmInstanceFilter*** (abstract)			
Note	This meta-class acts as an abstract base class for security event filters that apply for an entire IdsM.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	IdsmRateLimitation, IdsmTrafficLimitation			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table 4.10: AbstractSecurityIdsmInstanceFilter

4.5.1 Rate Limitation Filter

[TPS_SECXT_01014] Semantics of IdsmRateLimitation ⌈IdsmRateLimitation defines a rate limitation filter. During each consecutive time interval timeInterval, when the accumulated number of sent QSEv messages exceeds maxEventsInInterval then all subsequent QSEv messages within the same time interval are not sent onto the network but discarded.⌋

[constr_5606] Valid interval for attribute IdsmRateLimitation.timeInterval ⌈The valid interval for attribute IdsmRateLimitation.timeInterval is 0..65535 seconds.⌋

[constr_5607] Valid interval for attribute IdsmRateLimitation.maxEventsInInterval ⌈The valid interval for attribute IdsmRateLimitation.maxEventsInInterval is 1..65535.⌋

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 4.11: IdsmRateLimitation

[constr_12022] Existence of role IdsmRateLimitation.maxEventsInInterval

Imposition time: IT_SecEx

⌈For each IdsmRateLimitation, the reference in the role maxEventsInInterval shall exist.⌋

[constr_12023] Existence of role IdsmRateLimitation.timeInterval

Imposition time: IT_SecEx

⌈For each IdsmRateLimitation, the reference in the role timeInterval shall exist.⌋

4.5.2 Traffic Limitation Filter

[TPS_SECXT_01015] Semantics of IdsmTrafficLimitation ⌈IdsmTrafficLimitation defines a traffic limitation filter. During each consecutive time interval timeInterval, when the accumulated size of sent QSEv messages exceeds maxBytesInInterval then all subsequent QSEv messages within the same time interval are not sent onto the network but discarded.⌋

[constr_5608] Valid interval for attribute IdsmTrafficLimitation.timeInterval ⌈The valid interval for attribute IdsmTrafficLimitation.timeInterval is 0..65535 seconds.⌋

[constr_5609] Valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval ⌈The valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval is 1..65535.⌋

[constr_12009] Existence of attribute IdsmTrafficLimitation.maxBytesInInterval ⌈For each IdsmTrafficLimitation, the attribute maxBytesInInterval shall exist.⌋

[constr_12010] Existence of attribute IdsmTrafficLimitation.timeInterval ⌈For each IdsmTrafficLimitation, the attribute timeInterval shall exist.⌋

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 4.12: IdsmTrafficLimitation

4.6 Overview on Security Event Mappings

The mapping of SecurityEventDefinitions serves the following three main purposes:

1. to link the SecurityEventDefinition with the IdsmInstance that shall be able to report it,
2. to associate the SecurityEventDefinition with the SecurityEventFilterChain which is applicable for it,
3. to add information on the executional context in which the SecurityEventDefinition can occur.

To meet these three purposes, the abstract meta-class SecurityEventContextMapping has the following derived concrete meta-classes (also shown in Figure 4.6):

- SecurityEventContextMappingBswModule
- SecurityEventContextMappingFunctionalCluster
- SecurityEventContextMappingCommConnector
- SecurityEventContextMappingApplication

These concrete meta-classes add their respective executional context information to the mapping of SecurityEventDefinitions to an IdsmInstance.

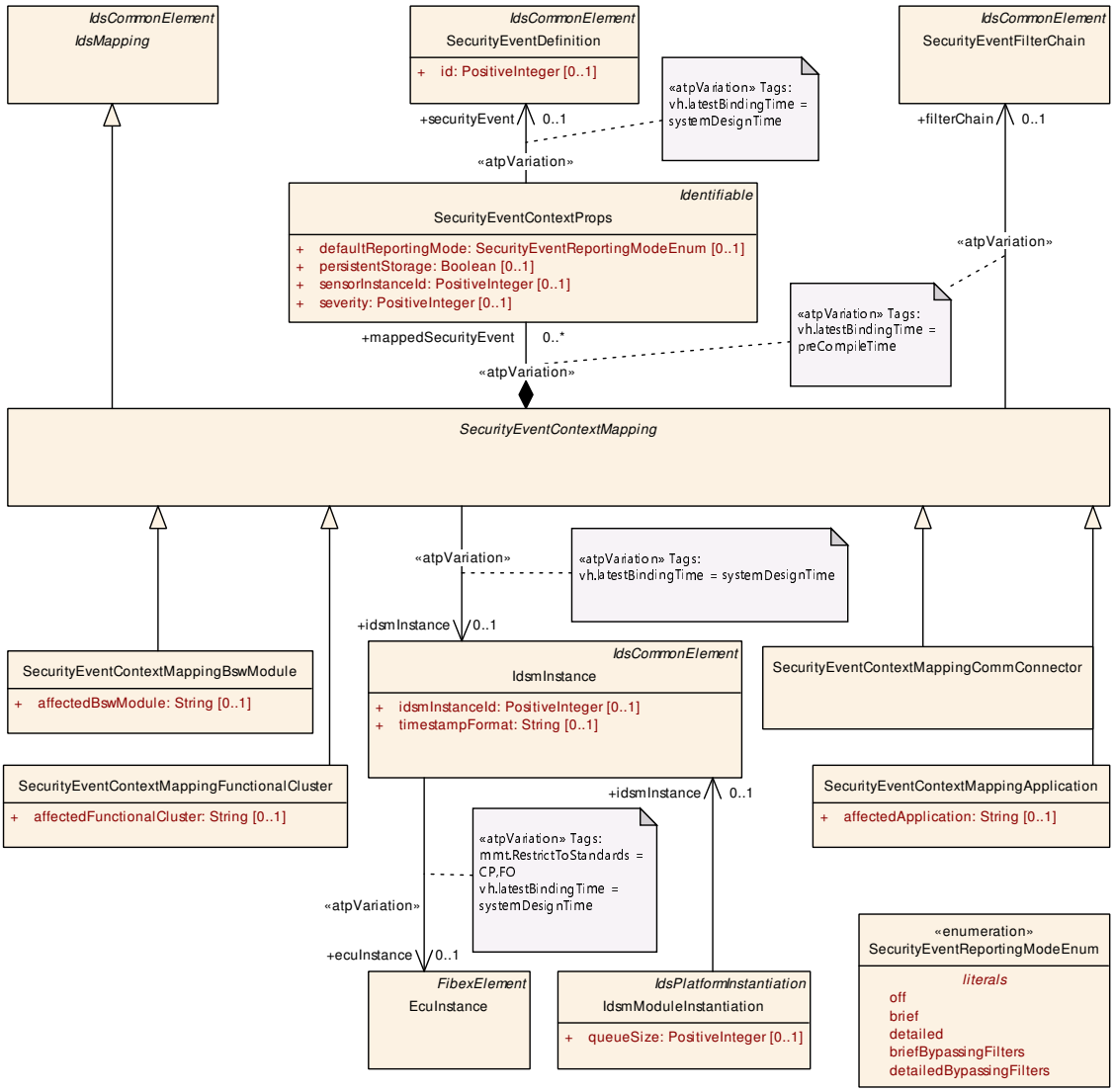


Figure 4.6: Modeling overview on mapping of security events

4.6.1 Mapping of Security Events to an IdsM Instance

[TPS_SECXT_01016] Semantics of `SecurityEventContextMapping` ⌈The abstract meta-class `SecurityEventContextMapping` maps the `SecurityEventDefinition`s respectively referenced in the role `securityEvent` by the `SecurityEventContextProps` that are aggregated in the role `mappedSecurityEvent` to the `IdsmInstance` referenced in the role `idsmInstance`.⌋

Since the `IdsmInstance` itself refers to the `EcuInstance` (for Classic Platform) or to the `IdsmModuleInstantiation` (for Adaptive Platform) which is again aggregated by `Machine`, the mapping of `SecurityEventDefinition`s to an `IdsmInstance` implicitly defines the mapping of these `SecurityEventDefinition`s to an `EcuInstance` or to a `Machine` as well (for CP and AP, respectively).

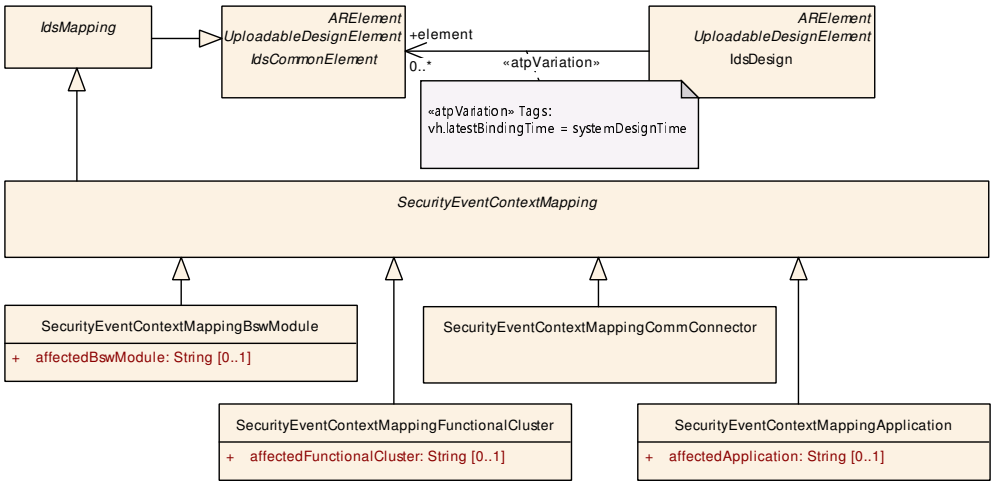


Figure 4.7: Meta-class hierarchy related to `SecurityEventContextMapping`

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, SecurityEventContextMappingBswModule, SecurityEvent ContextMappingCommConnector, SecurityEventContextMappingFunctionalCluster			
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 4.13: SecurityEventContextMapping

[constr_12017] Existence of attribute `SecurityEventContextMapping.idsm-Instance` ⌈For each `SecurityEventContextMapping`, the reference to `Idsm-Instance` in the role `idsmInstance` shall exist.⌋

[constr_12018] Existence of attribute SecurityEventContextMapping.mappedSecurityEvent ⌈For each SecurityEventContextMapping, the aggregation of SecurityEventContextProps in the role mappedSecurityEvent shall exist.⌋

[TPS_SECXT_01040] Semantics of SecurityEventContextProps ⌈The meta-class SecurityEventContextProps aggregated by SecurityEventContextMapping in the role mappedSecurityEvent contains mapping-dependent properties applicable to the SecurityEventDefinition referenced in the role securityEvent. These properties are therefore only relevant in the context of the mapping of this SecurityEventDefinition to the IdsmInstance as specified in [TPS_SECXT_01016].⌋

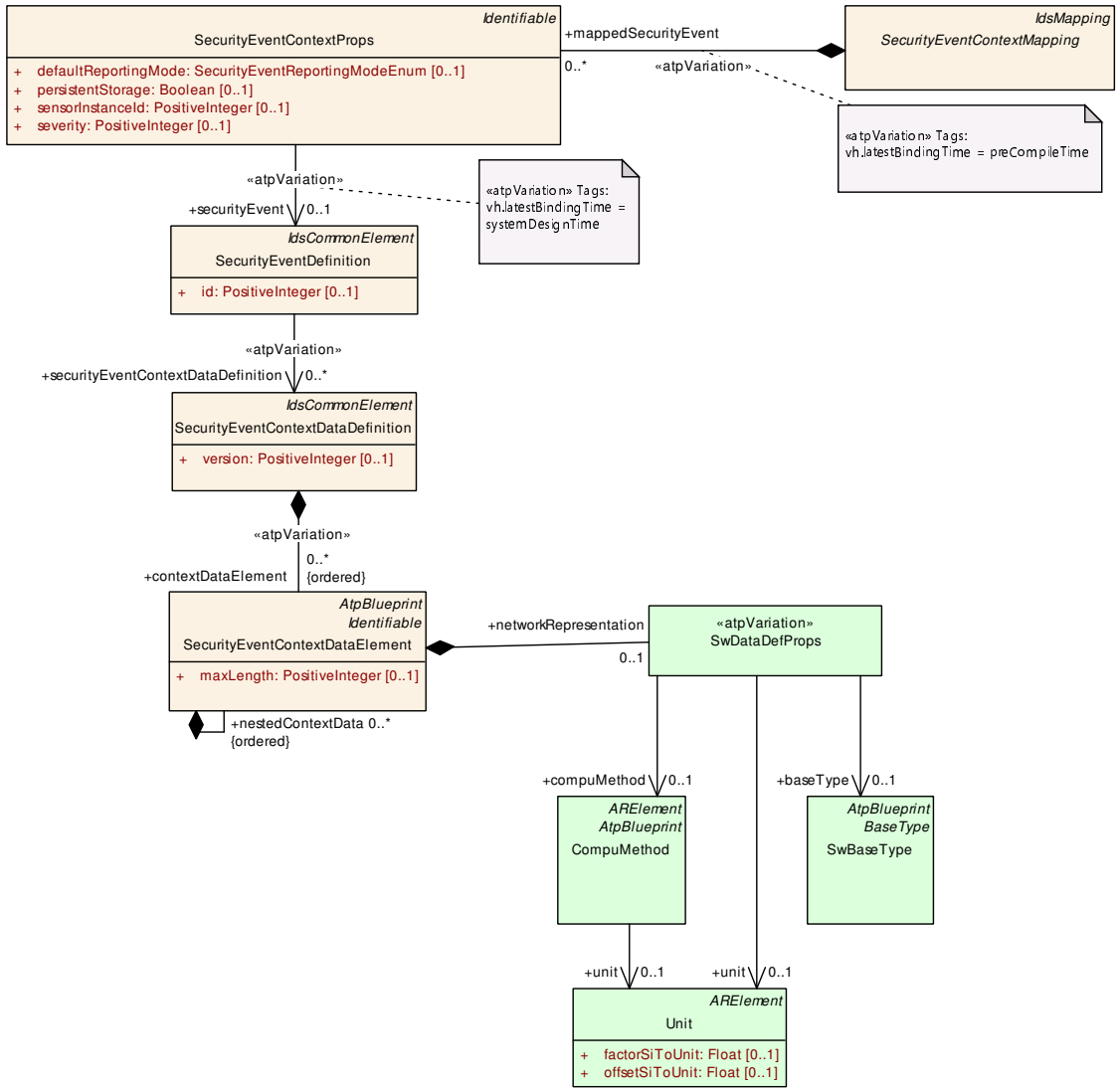


Figure 4.8: Modeling of SecurityEventContextProps

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*			
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 4.14: SecurityEventContextProps

[constr_12019] Existence of attribute `SecurityEventContextProps.defaultReportingMode` ⌈For each `SecurityEventContextProps`, the attribute `defaultReportingMode` shall exist.⌋

[constr_12020] Existence of attribute `SecurityEventContextProps.securityEvent` ⌈For each `SecurityEventContextProps`, the reference to `SecurityEventDefinition` in the role `securityEvent` shall exist.⌋

[constr_12021] Existence of attribute `SecurityEventContextProps.sensorInstanceId` ⌈For each `SecurityEventContextProps`, the attribute `sensorInstanceId` shall exist.⌋

4.6.1.1 Context Data definition

For certain security events, the security sensor can provide additional context data to be reported to the IdsM in order to better support, for example, analysis of a possible security threat.

[TPS_SECXT_01005] Semantics of `SecurityEventContextDataDefinition` ⌈If additional context data is added to a `SecurityEventDefinition` when it is reported to the IdsM, then a `SecurityEventContextDataDefinition` shall be referenced by the `SecurityEventDefinition` in the role `securityEventContextDataDefinition`.⌋

Class	SecurityEventContextDataDefinition			
Note	This meta-class represents the possibility to add context data to the referencing SecurityEventDefinition. **Tags:** atp.recommendedPackage=SecurityEventContextDataDefinitions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextData Element (ordered)	SecurityEventContext DataElement	*	aggr	Description of contained SecurityEventContextData Elements. If the SecurityEventContextDataDefintion has a primitive type then only one SecurityEventContextData Element shall be used. If the SecurityEventContextData Defintion is structured into several elements then for each one a SecurityEventContextDataElement shall be aggregated. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
version	PositiveInteger	0..1	attr	Version number of the context data. For more details see the IDSM protocol specfication.

Table 4.15: SecurityEventContextDataDefinition

Class	SecurityEventContextDataElement			
Note	This meta-class represents one ContextDataElement in the context of the aggregating SecurityEvent ContextDataDefintion.			
Base	*ARObject*, *AtpBlueprint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecurityEventContextDataDefinition.contextDataElement, SecurityEventContextDataElement.nested ContextData			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxLength	PositiveInteger	0..1	attr	Describes the maximal length of the context data in case of Arrays and Strings.
nestedContext Data (ordered)	SecurityEventContext DataElement	*	aggr	This self-aggregation supports the desription of nested context data.
network Representation	SwDataDefProps	0..1	aggr	Definition of the networkRepresentation of the context data element. **Tags:** atp.Splitkey=networkRepresentation
securityEvent ReportInstance Definition	SecurityEventReport InstanceDefinition	0..1	ref	This reference identifies the definition of the report instance. **Tags:** atp.Splitkey=securityEventReportInstance Definition

Table 4.16: SecurityEventContextDataElement

Please note that the example in Chapter C.1 shows how a SecurityEventDefinition defines two versions of SecurityEventContextDataDefinition.

[TPS_SECXT_04000] SecurityEventContextDataDefinition with aggregated SecurityEventContextDataElements ⌈SecurityEventContextDataDefinition is used to describe how context data is structured into SecurityEventContextDataElements and how the networkRepresentation for each of the SecurityEventContextDataElements is defined.⌋

[TPS_SECXT_04001] **SecurityEventContextDataElement.networkRepresentation** ⌈SecurityEventContextDataElement.networkRepresentation is used to define the network representation of one context data element.⌋

[TPS_SECXT_04002] **Description of a primitive as networkRepresentation of a SecurityEventContextDataElement** ⌈SecurityEventContextDataDefinition that has a primitive type like Boolean, Signed Integer, Unsigned Integer, and Float shall aggregate one SecurityEventContextDataElement that in turn has a networkRepresentation that references a SwBaseType in the role baseType. For primitive types the length of the context data shall be derived from the baseTypeSize.⌋

[constr_9339] **SecurityEventContextDataElement.maxLength usage restriction** ⌈The SecurityEventContextDataElement.maxLength attribute shall only be set if the SecurityEventContextDataElement.networkRepresentation defines an Array or a String.⌋

[TPS_SECXT_04003] **SecurityEventContextDataDefinition that is structured into SecurityEventContextDataElements** ⌈SecurityEventContextDataDefinition that is structured shall be described with several SecurityEventContextDataElements and each of those SecurityEventContextDataElements shall define a networkRepresentation.⌋

[TPS_SECXT_04004] **SecurityEventContextDataElement that is structured into SecurityEventContextDataElements** ⌈SecurityEventContextDataElement that is structured shall aggregate the sub SecurityEventContextDataElements in the nestedContextData role and each of those nested SecurityEventContextDataElements shall define a networkRepresentation.⌋

[TPS_SECXT_04005] **Description of a String as networkRepresentation of a SecurityEventContextDataElement** ⌈A SecurityEventContextDataElement that has a String type shall be described by a reference to a SwBaseType in the networkRepresentation in which the following elements are set to a value:

- baseTypeEncoding (according to [TPS_SWCT_01845])
- baseTypeSize

⌋

[TPS_SECXT_04006] **Description of one dimensional array as networkRepresentation of a SecurityEventContextDataElement** ⌈A SecurityEventContextDataElement that has a one-dimensional Array type shall set the maxLength attribute to a value and the data type of the array shall be described by the SwBaseType in the networkRepresentation.⌋

Please note that a SecurityEventContextDataElement of type String is described as an Array of characters.

[TPS_SECXT_04007] Description of a n-dimensional array as networkRepresentation of a SecurityEventContextDataElement ⌈A SecurityEventContextDataElement that has a n-dimensional Array type shall aggregate further SecurityEventContextDataElements in the role nestedContextData and all the aggregated nestedContextDatas shall have the maxLength attribute set to a value. Each nestedContextData defines one dimension in the Array that is represented by the aggregating SecurityEventContextDataElement.

- The maxLength attribute of the aggregating SecurityEventContextDataElement defines the number of dimensions.
- The maxLength attribute in the nestedContextData defines the maximal number of entries in the dimension.

The data type of the Array is described by the SwBaseType in the networkRepresentation of the aggregating SecurityEventContextDataElement.⌋

[constr_9340] Datatype of an Array ⌈The nestedContextData that is aggregated by a SecurityEventContextDataElement that has the maxLength attribute set to a value (represents an Array) shall not define a SwBaseType in the networkRepresentation since the data type of the Array is described by the SwBaseType in the networkRepresentation of the aggregating SecurityEventContextDataElement.⌋

[constr_9341] CompuMethod in SecurityEventContextDataElement.networkRepresentation ⌈The CompuMethod that is used in the networkRepresentation of a SecurityEventContextDataElement is limited to category TEXTTABLE.⌋

With the TEXTTABLE CompuMethod it is possible to define an enumeration as networkRepresentation of a SecurityEventContextDataElement. More details can be found in [7].

[constr_9342] Allowed range of SecurityEventContextDataDefinition.version ⌈The value given for SecurityEventContextDataDefinition.version shall be in the range from 0 to 65535.⌋

4.6.1.2 Default Reporting Mode definition

[TPS_SECXT_01017] Semantics of attribute SecurityEventContextProps.defaultReportingMode ⌈The attribute defaultReportingMode of SecurityEventContextProps defines the default *reporting mode* applicable to the SecurityEventDefinition referenced in the role securityEvent as follows:

off: The reported security event is not processed further by the IdsM and therefore discarded.

brief: Only the main security event properties such as its ID are processed. Any additional context data (if existing) is discarded.

detailed: The main properties and the context data (if existing) of the reported security event are processed further.

briefBypassingFilters: The reported security event without its context data (if existing) is processed further but the SecurityEventFilterChain is bypassed.

detailedBypassingFilter: The reported security event including its context data (if existing) is processed further but the SecurityEventFilterChain is bypassed.

⌋

Please note that during runtime of the IdsM, the reporting mode of a specific SecurityEventDefinition can be changed through diagnostic services.

Enumeration	**SecurityEventReportingModeEnum**
Note	This enumeration controls the reporting mode of a security event.
Aggregated by	SecurityEventContextProps.defaultReportingMode
Literal	***Description***
brief	Only the main security event properties such as its ID are processed. Any additional context data (if existing) is discarded. **Tags:** atp.EnumerationLiteralIndex=1
briefBypassing Filters	The reported security event without its context data (if existing) is processed further but the filter chain is bypassed. **Tags:** atp.EnumerationLiteralIndex=3
detailed	The main properties and the context data (if existing) of the reported security event are processed further. **Tags:** atp.EnumerationLiteralIndex=2
detailedBypassing Filters	The reported security event including its context data (if existing) is processed further but the filter chain is bypassed. **Tags:** atp.EnumerationLiteralIndex=4
off	The reported security event is not further processed by the IdsM and therefore discarded. **Tags:** atp.EnumerationLiteralIndex=0

Table 4.17: SecurityEventReportingModeEnum

4.6.1.3 Persistent Storage definition

[TPS_SECXT_01041] Semantics of attribute SecurityEventContextProps.persistentStorage ⌈The attribute persistentStorage of SecurityEventContextProps defines whether a qualified reporting event of the SecurityEventDefinition referenced in the role securityEvent shall be stored persistently by the IdsmInstance on which the referenced SecurityEventDefinition is mapped:

false: The mapped IdsmInstance *shall not* persistently store qualified reporting events of the SecurityEventDefinition referenced in the role securityEvent.

true: The mapped IdsmInstance *shall* persistently store qualified reporting events of the SecurityEventDefinition referenced in the role securityEvent.

⌋

4.6.1.4 Severity Level definition

[TPS_SECXT_01042] Semantics of attribute SecurityEventContextProps.severity ⌈The attribute severity of SecurityEventContextProps defines the severity level to be applied to the SecurityEventDefinition referenced in the role securityEvent. The specified severity level shall only be relevant for the mapping of this SecurityEventDefinition onto the IdsmInstance as specified in [TPS_SECXT_01016].⌋

Please note that the severity level meanings associated with specific positive integer values of the attribute severity is currently not specified by AUTOSAR but has to be defined by the party responsible for the IDS system design (e.g. an OEM).

4.6.1.5 Sensor Instance ID definition

[TPS_SECXT_01047] Semantics of attribute SecurityEventContextProps.sensorInstanceId ⌈The attribute sensorInstanceId of SecurityEventContextProps defines numerical identifier of the security sensor that detects the SecurityEventDefinition referenced in the role securityEvent. The specified sensorInstanceId shall only be relevant for the mapping of this SecurityEventDefinition onto the IdsmInstance as specified in [TPS_SECXT_01016].⌋

[constr_12004] Valid interval for attribute SecurityEventContextProps.sensorInstanceId ⌈The valid interval for attribute SecurityEventContextProps.sensorInstanceId is 0..63.⌋

4.6.2 Mapping of Security Events with BSW Module Context

[TPS_SECXT_01018] Semantics of SecurityEventContextMappingBswModule ⌈For the Classic Platform, SecurityEventContextMappingBswModule defines that the mapped SecurityEventDefinitions can occur in the executional context of the BSW module defined as name by attribute affectedBswModule on the mapped IdsmInstance.⌋

[TPS_SECXT_02001] Content of `SecurityEventContextMappingBswModule.affectedBswModule` ⌈The value of `SecurityEventContextMappingBswModule.affectedBswModule` shall be taken from the column "Module abbreviation (API service prefix)" of the table [SWS_BSW_91001] defined in the [8, CP SWS BSWGeneral] Chapter 11.2.⌋

[TPS_SECXT_01019] Mapping of Security Events to Filter Chain by `SecurityEventContextMappingBswModule` ⌈Each individual `SecurityEventDefinition` mapped through the `SecurityEventContextProps`s aggregated by `SecurityEventContextMappingBswModule` shall be input to the `SecurityEventFilterChain` referenced in the role `filterChain` by the same `SecurityEventContextMappingBswModule`.⌋

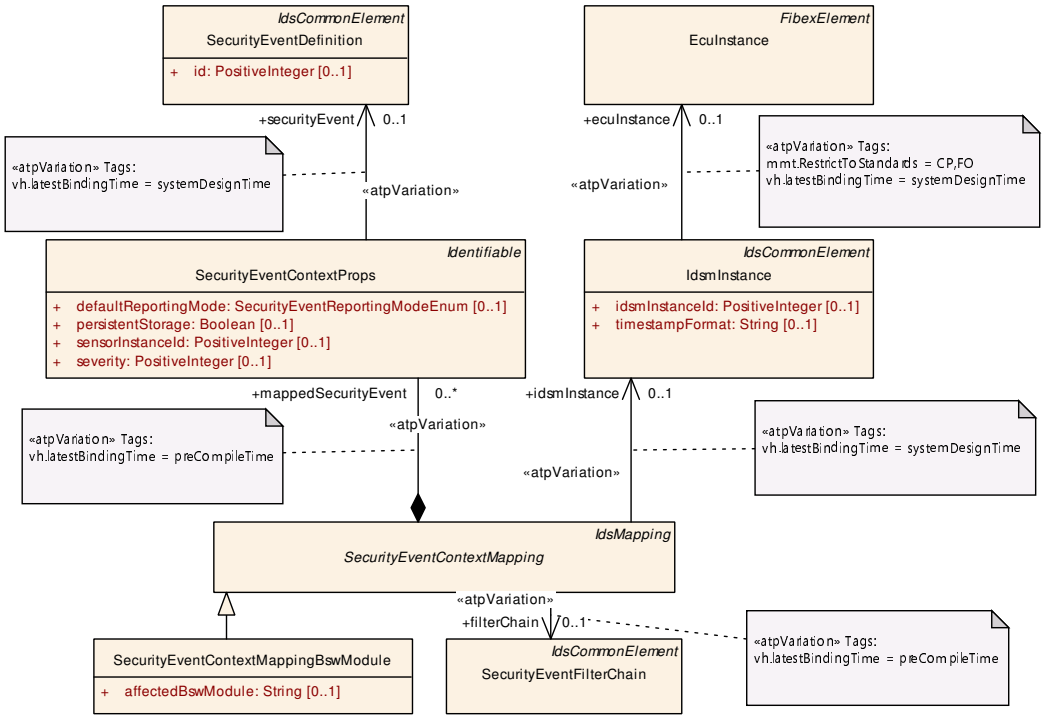


Figure 4.9: Modeling of `SecurityEventContextMappingBswModule`

Class	**SecurityEventContextMappingBswModule**			
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context of a BSW module in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingBswModules This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	**SecurityEventContextMappingBswModule**			
affectedBsw Module	String	0..1	attr	This attribute is used to identify the name of the BSW module in whose executional context a security event can occur. The set of BSW module names is standardized by AUTOSAR.

Table 4.18: SecurityEventContextMappingBswModule

4.6.3 Mapping of Security Events with Functional Cluster Context

[TPS_SECXT_01020] Semantics of `SecurityEventContextMappingFunctionalCluster` ⌈For the Adaptive Platform, `SecurityEventContextMappingFunctionalCluster` defines that the mapped `SecurityEventDefinition`s can occur in the executional context of the functional cluster defined as name by attribute `affectedFunctionalCluster` on the mapped `IdsmInstance`.⌋

[TPS_SECXT_02002] Content of `SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster` ⌈The value of `SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster` shall be taken from the column “Name” of the table defined in [AP_SWS_GENERAL_90025].⌋

[TPS_SECXT_01021] Mapping of Security Events to Filter Chain by `SecurityEventContextMappingFunctionalCluster` ⌈Each individual `SecurityEventDefinition` mapped through the `SecurityEventContextProps`s aggregated by `SecurityEventContextMappingFunctionalCluster` shall be input to the `SecurityEventFilterChain` referenced in the role `filterChain` by the same `SecurityEventContextMappingFunctionalCluster`.⌋

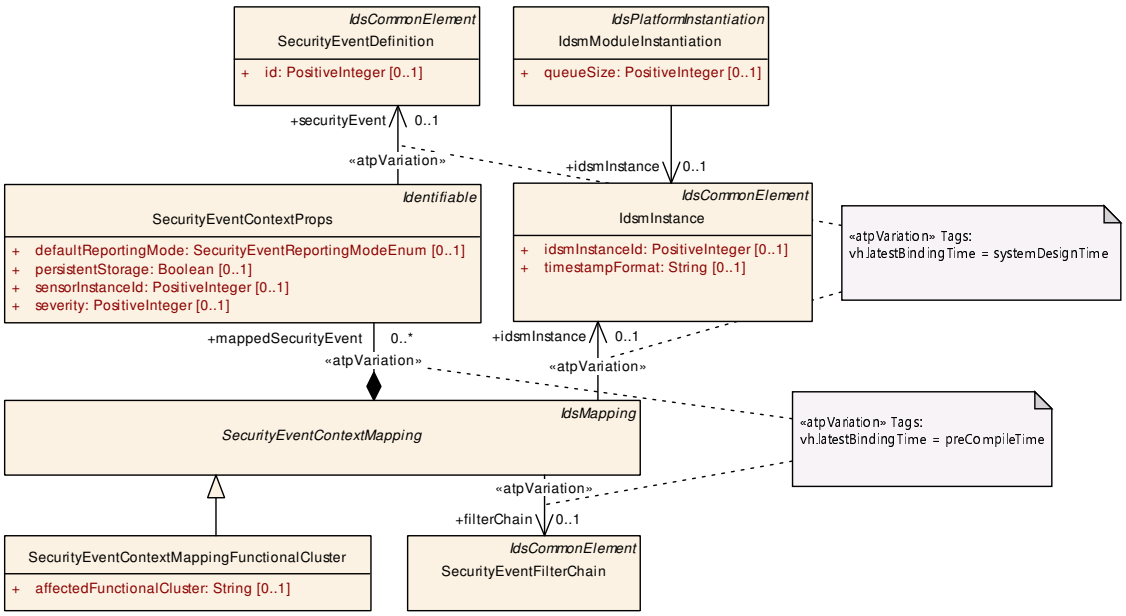


Figure 4.10: Modeling of `SecurityEventContextMappingFunctionalCluster`

Class	SecurityEventContextMappingFunctionalCluster			
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context of a functional cluster in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingFunctionalClusters This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
affected Functional Cluster	String	0..1	attr	This attribute is used to identify the name of the functional cluster in whose executional context a security event can occur. The set of functional cluster names is standardized by AUTOSAR.

Table 4.19: SecurityEventContextMappingFunctionalCluster

[constr_12024] Existence of role `SecurityEventContextMapping-FunctionalCluster.affectedFunctionalCluster`

Imposition time: `IT_SecEx`

⌈For each `SecurityEventContextMappingFunctionalCluster`, the reference in the role `affectedFunctionalCluster` shall exist.⌋

4.6.4 Mapping of Security Events with Communication Connector Context

[TPS_SECXT_01022] Semantics of `SecurityEventContextMappingComm-Connector` ⌈`SecurityEventContextMappingCommConnector` defines that the mapped `SecurityEventDefinition`s can occur in the executional context related to the referenced `CommunicationConnector` in the role `commConnector` on the mapped `IdsmInstance`.⌋

[TPS_SECXT_01023] Mapping of Security Events to Filter Chain by `SecurityEventContextMappingCommConnector` ⌈Each individual `Security-EventDefinition` mapped through the `SecurityEventContextProps`s aggregated by `SecurityEventContextMappingCommConnector` shall be input to the `SecurityEventFilterChain` referenced in the role `filterChain` by the same `SecurityEventContextMappingCommConnector`.⌋

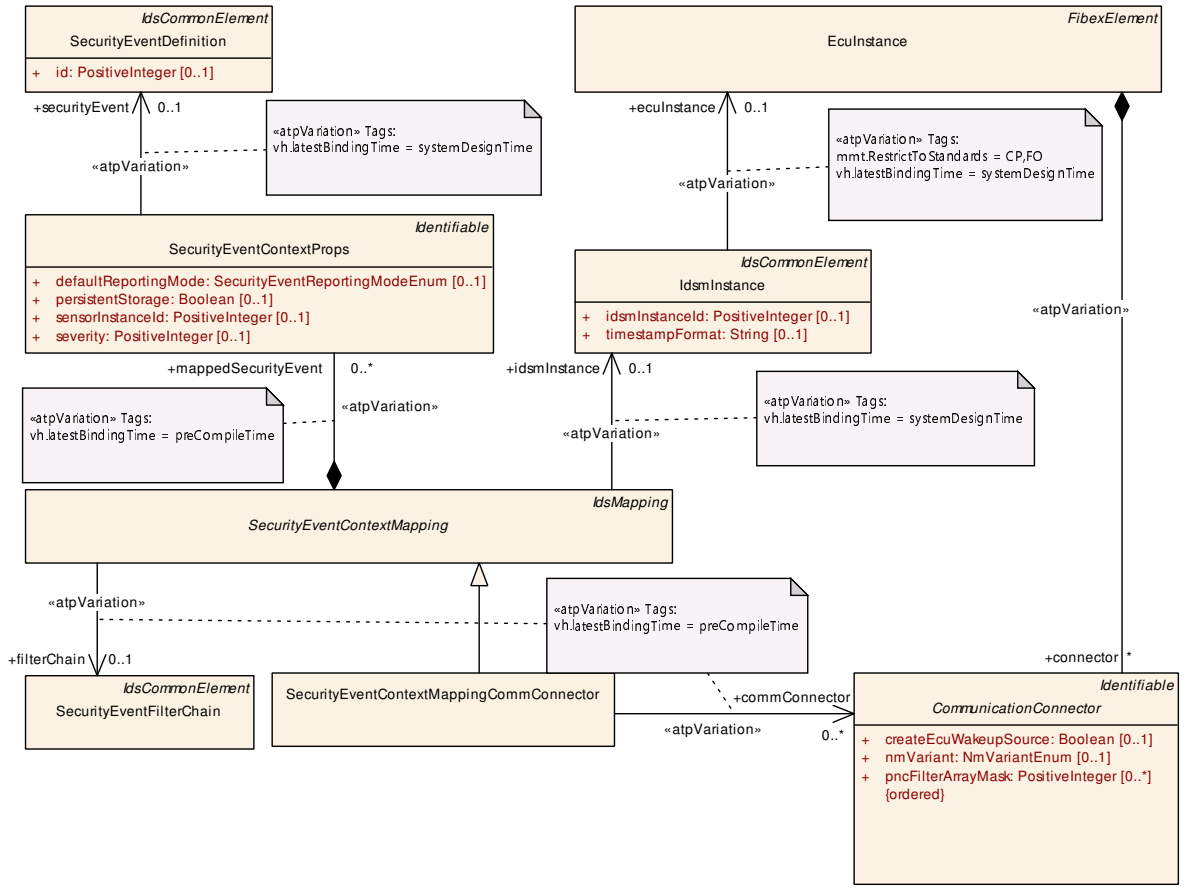


Figure 4.11: Modeling of `SecurityEventContextMappingCommConnector`

Class	**SecurityEventContextMappingCommConnector**			
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context related to a CommunicationConnector in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingCommConnectors			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
comm Connector	Communication Connector	*	ref	This reference identifies the respective Communication Connector for which the collection of security events can be reported. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commConnector, comm Connector.communicationConnector, comm Connector.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table 4.20: SecurityEventContextMappingCommConnector

4.6.5 Mapping of Security Events with Application Context

[TPS_SECXT_01024] **Semantics of SecurityEventContextMappingApplication** ⌈SecurityEventContextMappingApplication defines that the mapped SecurityEventDefinitions can occur in the executional context of the application defined as name by attribute affectedApplication on the mapped IdsmInstance.⌋

[TPS_SECXT_01025] **Mapping of Security Events to Filter Chain by SecurityEventContextMappingApplication** ⌈Each individual SecurityEventDefinition mapped through the SecurityEventContextProps aggregated by SecurityEventContextMappingApplication shall be input to the SecurityEventFilterChain referenced in the role filterChain by the same SecurityEventContextMappingApplication.⌋

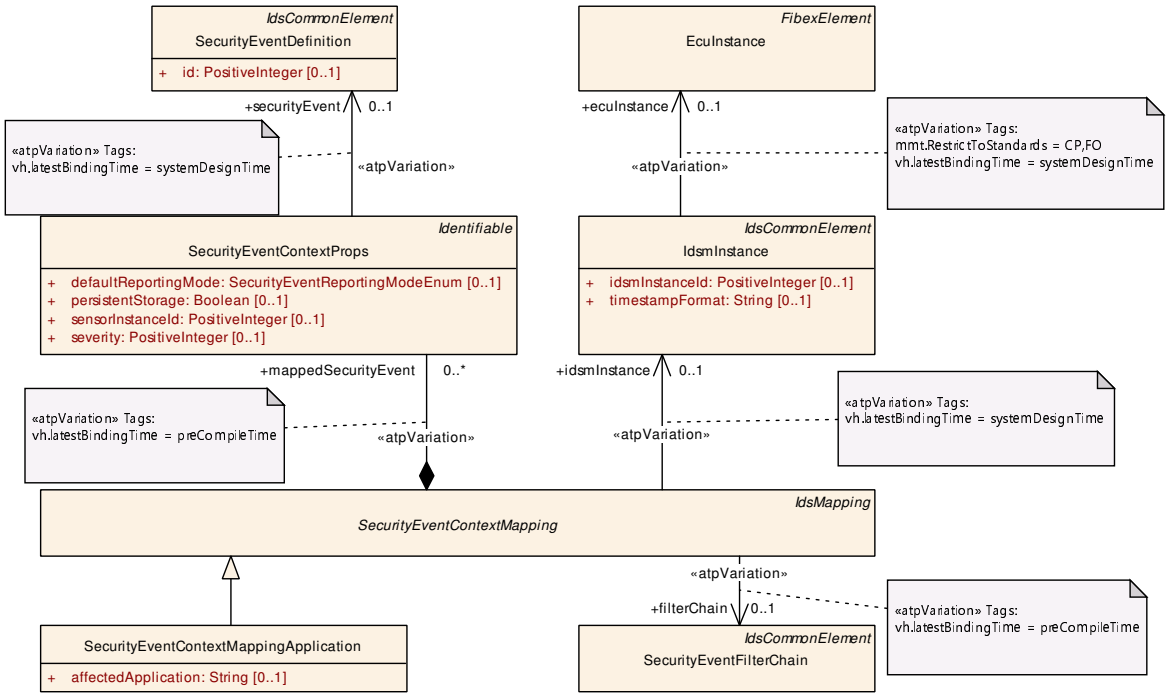


Figure 4.12: Modeling of SecurityEventContextMappingApplication

Class	**SecurityEventContextMappingApplication**			
Note	This meta-class represents the ability to associate a collection of security events with an IdsM instance and with the executional context of an application (e.g. name of SWC on CP or name of SWCL on AP) in which this IdsM instance can receive reports for these security events. **Tags:** atp.recommendedPackage=SecurityEventContextMappingApplications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping*, *Uploadable DesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***

△

Class	**SecurityEventContextMappingApplication**			
affected Application	String	0..1	attr	This attribute is used to identify the name of the application in whose executional context a security event can occur. This application can be, for example, a name of a Software Component (for CP) or a Software Cluster name (for AP).

Table 4.21: SecurityEventContextMappingApplication

[constr_12025] Existence of role `SecurityEventContextMapping-Application.affectedApplication`

Imposition time: `IT_SecEx`

⌈For each `SecurityEventContextMappingApplication`, the reference in the role `affectedApplication` shall exist.⌋

4.7 Configuration of an IdsM Instance

The Security Extract Template allows for definition of IdsM instances that can be individually deployed on an ECU instance (Classic Platform) or a machine (Adaptive Platform). An IdsmInstance can be further attributed with system-level functional properties and put into relation with the SecurityEventDefinitions relevant to the IdsM instance.

The network configuration for an IdsM instance is handled differently on the Classic and on the Adaptive Platform (see 4.7.3).

[TPS_SECXT_01026] Semantics of IdsmInstance on CP ⌈On the Classic Platform, the IdsmInstance represents an instance of the IdsM that runs on the EcuInstance which is referenced in the role ecuInstance.⌋

Please note that certain ECU configuration parameters for an IdsmInstance on the Classic Platform can be derived from definitions of a Security Extract. These ECU configuration parameters and their upstream mappings to specific Security Extract definitions are stated in Appendix E.

[TPS_SECXT_01027] Semantics of IdsmInstance on AP ⌈On the Adaptive Platform, the IdsmInstance represents an instance of the IdsM as defined by IdsmModuleInstantiation which references IdsmInstance in the role idsmInstance.⌋

[constr_5610] Unambiguous definition of execution platform for an IdsmInstance ⌈For the meta-class IdsmInstance, either the reference in the role ecuInstance or the reference in the role idsmInstance shall be defined in order to ensure that the platform (Classic or Adaptive) on which an IdsmInstance is targeted to run is unambiguously defined.⌋

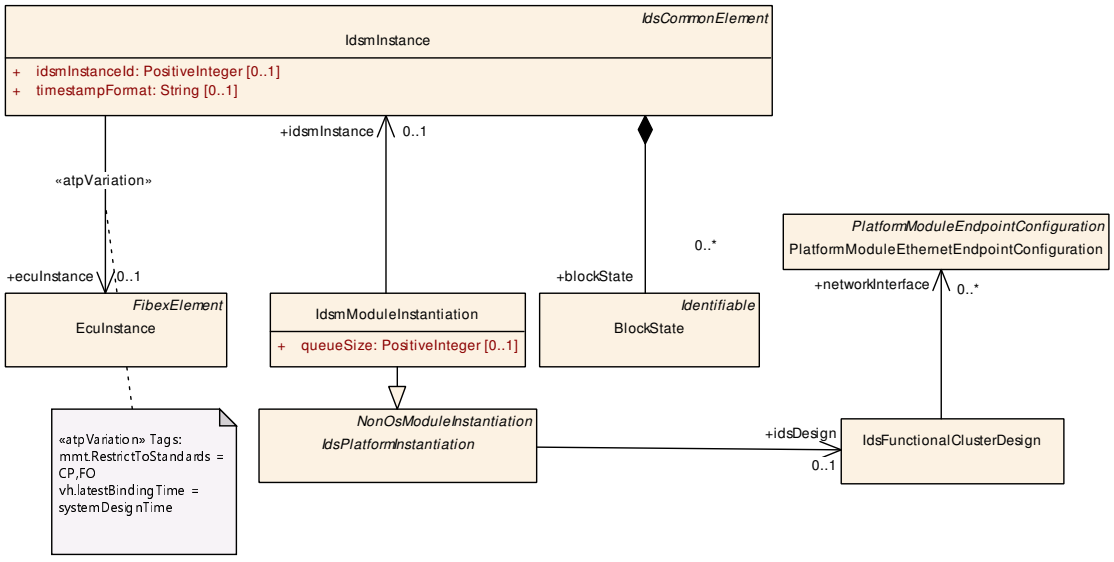


Figure 4.13: Modeling overview of IdsmInstance

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**
blockState	BlockState	*	aggr	This reference defines the BlockState in the collection BlockStateSet. **Tags:** atp.Splitkey=blockState, blockState.shortName This Attribute is only used by the AUTOSAR Classic Platform.
ecuInstance	EcuInstance	0..1	ref	This reference identifies the EcuInstance whose security events (of any type) shall be limited by the specific class of filters. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=ecuInstance, ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime This Attribute is only used by the AUTOSAR Classic Platform.
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
signature SupportCp	IdsmSignatureSupportCp	0..1	aggr	The existence of this aggregation specifies that the IdsM shall add a signature to the QSEv messages it sends onto the network. The cryptographic algorithm and key to be used for this signature is further specified by the aggregated meta-class specifically for the Classic Platform. **Tags:** atp.Splitkey=signatureSupportCp This Attribute is only used by the AUTOSAR Classic Platform.
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.



△

Class	IdsmInstance			
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, trafficLimitation Filter.idsmTrafficLimitation, trafficLimitationFilter.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table 4.22: IdsmInstance

4.7.1 Attributes of an IdsM Instance

For both platforms, the attributes of `IdsmInstance` further defines system-level functional properties.

4.7.1.1 Instance ID of IdsM

[TPS_SECXT_01028] Semantics of attribute `IdsmInstance.idsmInstanceId` ⌈The attribute `idsmInstanceId` of `IdsmInstance` defines the assigned identifier for the IdsM instance.⌋

[constr_12005] Valid interval for attribute `IdsmInstance.idsmInstanceId` ⌈The valid interval for attribute `IdsmInstance.idsmInstanceId` is 0..1023.⌋

[constr_12006] Existence of attribute `IdsmInstance.idsmInstanceId` ⌈For each `IdsmInstance`, the attribute `idsmInstanceId` shall exist.⌋

[constr_12007] Existence of attribute `IdsmInstance.rateLimitationFilter` ⌈For each `IdsmInstance`, the reference to `IdsmRateLimitation` in the role `rateLimitationFilter` shall exist.⌋

[constr_12008] Existence of attribute `IdsmInstance.trafficLimitationFilter` ⌈For each `IdsmInstance`, the reference to `IdsmTrafficLimitation` in the role `trafficLimitationFilter` shall exist.⌋

4.7.1.2 Timestamp in QSEv messages

[TPS_SECXT_01029] Definition of timestamp support for an `IdsmInstance` ⌈The existence of the attribute `timestampFormat` of `IdsmInstance` defines that the `IdsmInstance` shall add timestamp data to the QSEv messages it sends onto the network. That means, if no attribute `timestampFormat` is defined, then the `IdsmInstance` shall add no timestamp to the QSEv messages.⌋

[TPS_SECXT_01030] Semantics of attribute IdsmInstance.timestampFormat ⌈The content of the attribute timestampFormat of IdsmInstance defines the format of the timestamp data that the IdsmInstance shall add to the QSEv messages it sends onto the network:

- The string AUTOSAR specifies that the AUTOSAR standardized timestamp format shall be used (based on the AUTOSAR 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.

4.7.1.3 Signature Support in QSEv Messages

[TPS_SECXT_01031] Definition of signature support for an IdsmInstance on the AUTOSAR classic platform ⌈For an IdsmInstance, the existence of the reference in the role signatureSupportCp defines that the IdsmInstance shall add signature information (i.e. cryptographic authentication) to the QSEv messages it sends onto the network. That means, if neither of these two reference roles exists, then the IdsmInstance shall add no signature information to the QSEv messages.⌋

The configuration of signature information on the *AUTOSAR adaptive platform* is described in the document [4, Specification of Manifest].

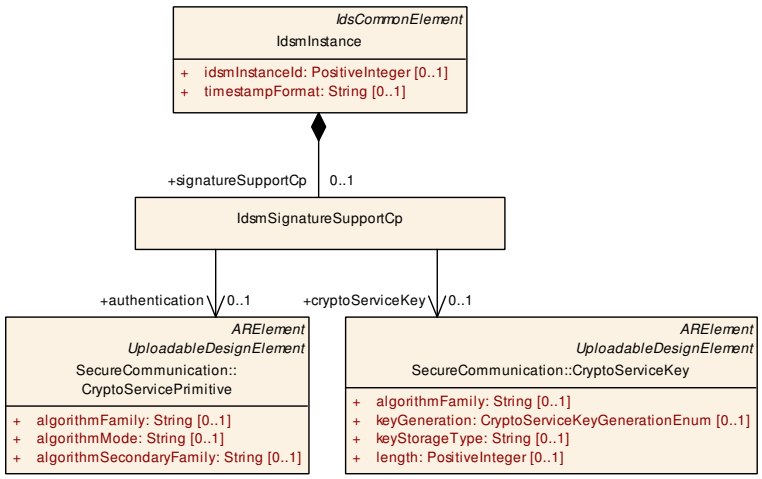


Figure 4.14: Modeling overview on signature support for an IdsmInstance

[TPS_SECXT_01032] Semantics of IdsmSignatureSupportCp ⌈For the Classic Platform, IdsmSignatureSupportCp represents the configuration of signature support for the aggregating IdsmInstance:

- The reference in the role authentication to CryptoServicePrimitive defines the cryptographic algorithm to be used.

- The reference in the role cryptoServiceKey to CryptoServiceKey defines the cryptographic key to be used.

⌋

4.7.2 Association of Security Events with an IdsM Instance

An IdsM instance needs to be configured regarding the security events it shall handle. The Security Extract Template supports this configuration by enabling the identification of all SecurityEventDefinitions that are applicable to an IdsmInstance.

All SecurityEventDefinitions that need to be configured for a specific IdsmInstance shall be identified by the relations of an IdsmInstance to the following derived concrete meta-classes of SecurityEventContextMapping:

- SecurityEventContextMappingBswModule for Classic Platform
- SecurityEventContextMappingFunctionalCluster for Adaptive Platform
- SecurityEventContextMappingCommConnector for both Classic and Adaptive Platforms
- SecurityEventContextMappingApplication for both Classic and Adaptive Platforms

[TPS_SECXT_01034] Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingBswModule on CP ⌈For all SecurityEventContextMappingBswModule on the Classic Platform referencing in the role idsmInstance the same IdsmInstance, the collection of all SecurityEventDefinitions referenced by their respective SecurityEventContextProps aggregated in the role mappedSecurityEvent shall be configured in and thus handled by this IdsmInstance.⌋

[TPS_SECXT_01035] Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingFunctionalCluster on AP ⌈For all SecurityEventContextMappingFunctionalCluster on the Adaptive Platform referencing in the role idsmInstance the same IdsmInstance, the collection of all SecurityEventDefinitions referenced by their respective SecurityEventContextProps aggregated in the role mappedSecurityEvent shall be configured in and thus handled by this IdsmInstance.⌋

[TPS_SECXT_01036] Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingCommConnector ⌈For all SecurityEventContextMappingCommConnector referencing in the role

idsmInstance the same IdsmInstance, the collection of all SecurityEvent-Definitions referenced by their respective SecurityEventContextProps aggregated in the role mappedSecurityEvent shall be configured in and thus handled by this IdsmInstance.⌋

[TPS_SECXT_01037] Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingApplication ⌈For all SecurityEventContextMappingApplication referencing in the role idsm-Instance the same IdsmInstance, the collection of all SecurityEvent-Definitions referenced by their respective SecurityEventContextProps aggregated in the role mappedSecurityEvent shall be configured in and thus handled by this IdsmInstance.⌋

4.7.3 Network Configuration of an IdsM instance

The network configuration of an IdsM instance defines how the IdsM communicates with the AUTOSAR communication stack in order to send QSEv messages onto the network addressed to the correct receiver entity.

Due to the different nature of Classic and Adaptive Platform, the network configuration of an IdsM instance is handled differently in both platforms.

[constr_5612] Unambiguous definition of platform-dependent network configuration for an IdsmInstance ⌈For the meta-class IdsmInstance, either the configuration of one GeneralPurposeIPdu with category="IDS" (for the Classic Platform as specified in [TPS_SECXT_01038]) or the network configuration through the reference idsmInstance (for the Adaptive Platform as specified in [TPS_SECXT_01039]) shall be defined in order to ensure that the platform-dependent network configuration is unambiguously defined.⌋

4.7.3.1 IdsM Network Configuration on Classic Platform

An IdsmInstance deployed on a specific EcuInstance uses a GeneralPurpose-IPdu to communicate with the PduR and thus send QSEv messages onto the network.

[TPS_SECXT_01038] Network configuration of an IdsmInstance on CP ⌈On the Classic Platform, the network configuration of an IdsmInstance is defined implicitly by two GeneralPurposeIPdus with category="IDS" on the same EcuInstance on which the IdsmInstance is deployed. One of these two GeneralPurposeIPdu with category="IDS" shall also be configured for use by a transport protocol while the other one shall be not.⌋

Please refer to the System Template [3] for more information and constraints on these GeneralPurposeIPdus with category="IDS".

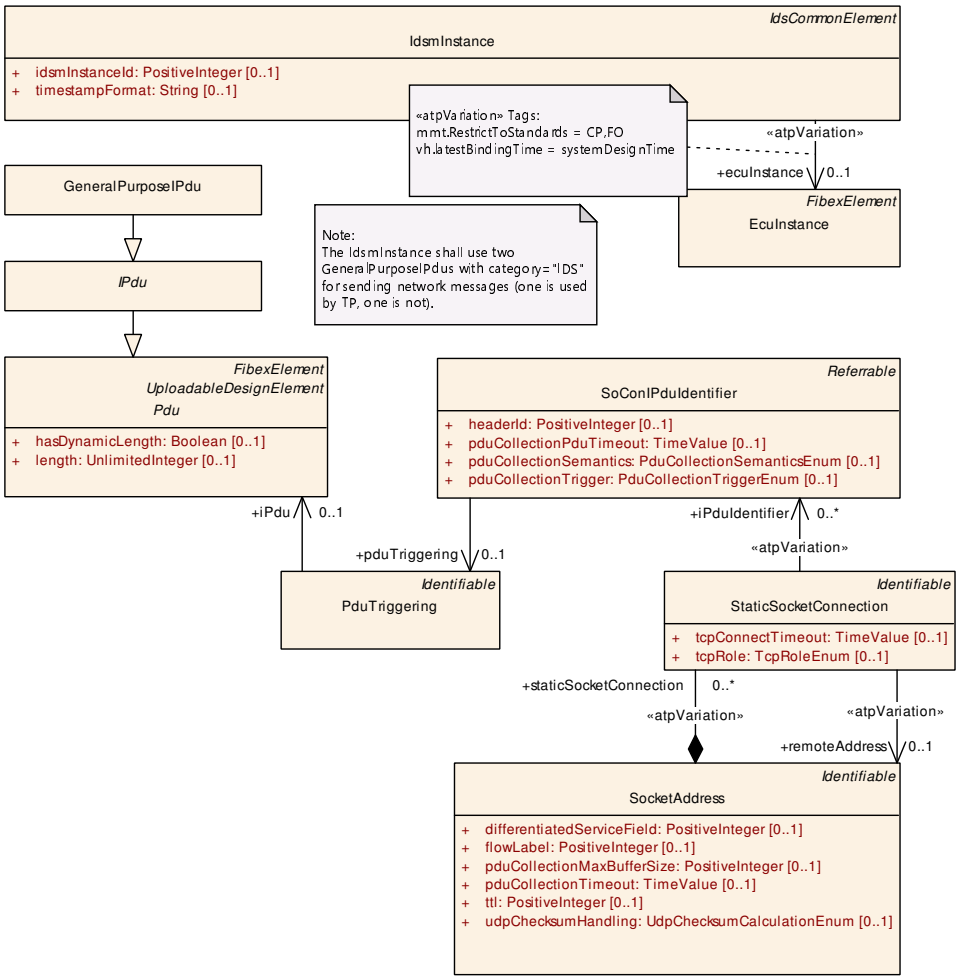


Figure 4.15: Modeling overview of the network configuration of an `IdsmInstance` on Classic Platform

4.7.3.2 IdsM Network Configuration on Adaptive Platform

For the Adaptive Platform, the deployment of an `IdsmInstance` on a specific `Machine` is defined by `IdsmModuleInstantiation` as part of the deployment section of the `Manifest` [4].

[TPS_SECXT_01039] Network configuration of an `IdsmInstance` on AP ⌈On the Adaptive Platform, the network configuration of an `IdsmInstance` shall be defined through the reference of `PlatformModuleEthernetEndpointConfiguration` in the role `networkInterface` by the `IdsmModuleInstantiation` which in turn is references the `IdsmInstance` in the role `idsmInstance`.⌋

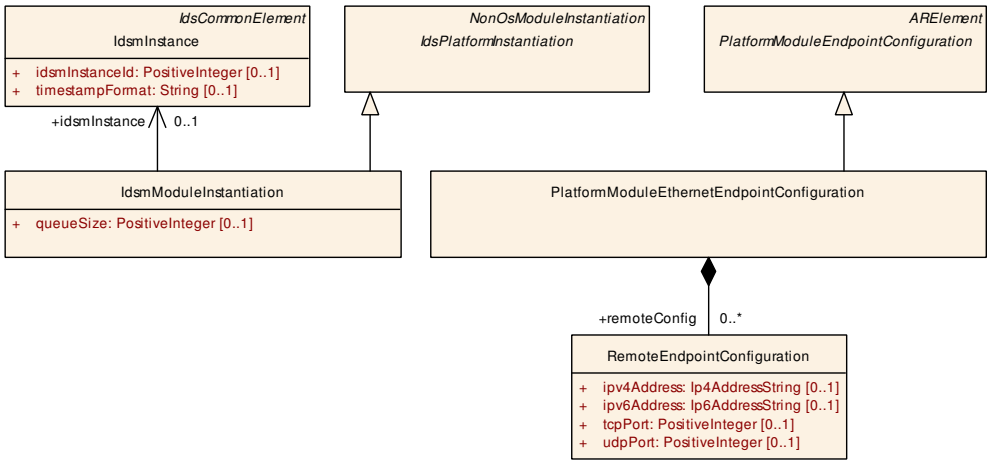


Figure 4.16: Modeling overview of the network configuration of an `IdsmInstance` on Adaptive Platform

4.7.4 Block States of an IdsM instance on CP

[TPS_SECXT_01048] Definition of `BlockState`s on CP ⌈On the Classic Platform, when a `SecurityEventStateFilter` is configured as part of a `SecurityEventFilterChain`, then the `BlockState`s that are required to represent the state machine that controls the `SecurityEventStateFilter` shall be defined and aggregated by the `IdsmInstance` which is mapped to the `SecurityEventFilterChain`. The `BlockState` shall be identified by its name defined as its `shortName`.⌋

Note: Since the `BlockState`s are named and identified using their respective `shortName`s, the uniqueness of their naming within an `IdsmInstance` is inherently given.

[TPS_SECXT_01044] Semantics of `BlockState` on CP ⌈On the Classic Platform, a `BlockState` referenced in the role `blockIfStateActiveCp` by a `SecurityEventStateFilter` indicates to this `SecurityEventStateFilter` to discard the reported `SecurityEventDefinition` when `BlockState` is currently active.⌋

[constr_5614] Upper bound for multiplicity of `BlockState`s aggregated by `IdsmInstance` ⌈For the meta-class `IdsmInstance`, the maximum number of aggregated `BlockState`s in the role `blockState` shall be 16.⌋

Note: The `BlockState` that is currently active within an `IdsmInstance` controls whether a `SecurityEventStateFilter` passes or blocks a reported security event. The logic of the state machine that indicates the `IdsmInstance`'s active block state needs to be implemented by the `Basic Software Mode Manager` (BswM) as arbitration rules according to [9].

Please also refer to Ch. 4.4.2.1.

Class	**BlockState**			
Note	This meta-class defines a block state that is part of the collection of block states belonging to a specific IdsmInstance. The IdsM shall discard any reported security event that is mapped to a filter chain containing a SecurityEventStateFilter that references the block state which is currently active in the IdsM. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IdsmInstance.blockState			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table 4.23: BlockState

4.8 Reporting Security Events

4.8.1 Security Event Context Data Identification

Some security event context data transport the identification of modeled artifacts using an numerical encoding.

One example is the identification of a specific `CanCommunicationConnector` representing one (of potentially many) CAN networks connected to an ECU. If it is important for the security event to identify which `CanCommunicationConnector` caused the security event to happen, then one `SecurityEventContextDataElement` will be defined to transport this id information.

At the point in time when a specific security event is reported to the Security Operation Center (SOC) the analysts will need to understand the relation between the numerical encoding (`id`) and the actual model element which corresponds to that `id`. In the example of the `CanCommunicationConnector` it can be determined on which physical CAN network the cause of the security event happened.

[TPS_SECXT_03000] Identification of security event context data ⌈The usage of context data for security events, formalized by meta-class `SecurityEventContextDataElement` allows an identification of some context data elements in terms of its origin in the AUTOSAR model.

The identification of the context data element can be explicitly modeled by means of the reference in the role `SecurityEventReportInstanceValue.contextDataElementIdentification` to `SecurityEventContextDataElement` that in turn refers to its corresponding `SecurityEventReportInstanceDefinition` in the role `securityEventReportInstanceDefinition`.⌋

The model structure described in [TPS_SECXT_03000] is depicted in Figure 4.17. Please note that the diagram has intentionally been organized to show the distinction between

- model elements that could be used to create standardized content on M1 level and

- model elements that are project-specific on M1 level.

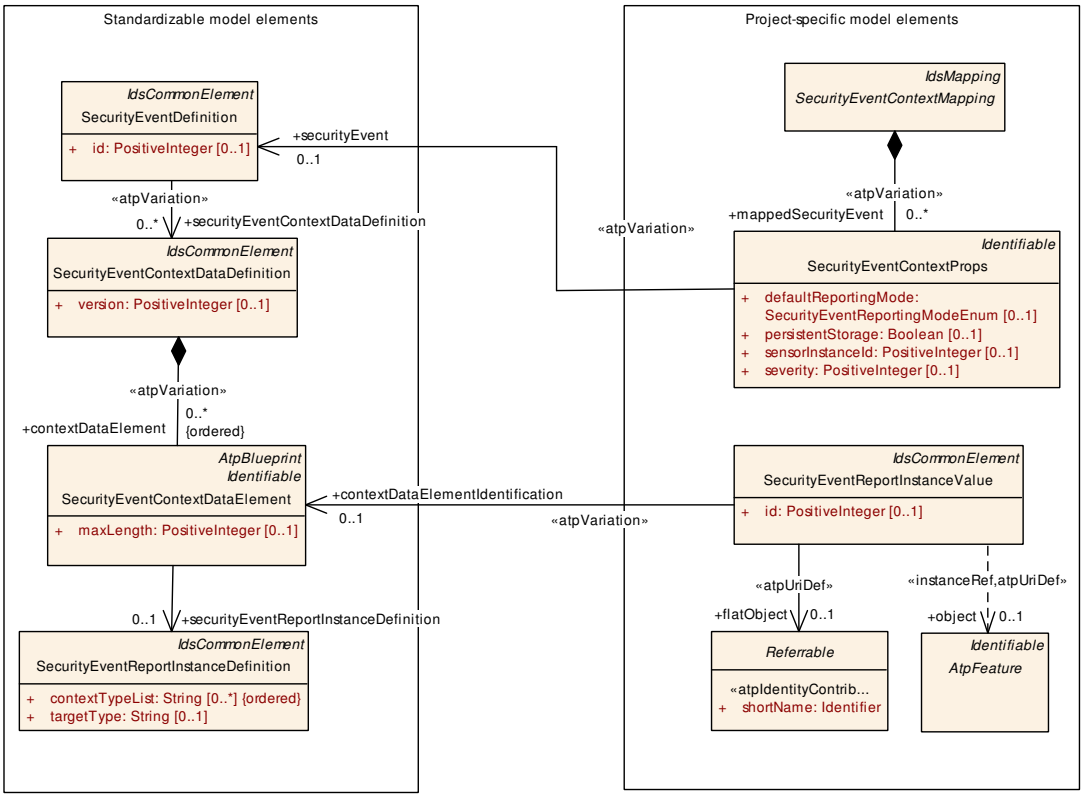


Figure 4.17: Specification of context data identification

The order of the collection `SecurityEventContextDataDefinition.contextDataElement` is significant, hence the qualification as `ordered` in the modeling of the aggregation.

[TPS_SECXT_03001] Semantics of meta-class `SecurityEventReportInstanceValue` ⌈Meta-class `SecurityEventReportInstanceValue` contributes an identification to the context data element by means of the attribute `id`. In addition, two alternative ways of referencing the **actual** model element that is represented by the context data element value are defined:

- The reference in the role `flatObject` can be used to refer to model elements for which a *flat reference* is applicable. An example for this kind of referenced object could be an `CanCommunicationConnector`.
- The instanceRef in the role `object` can be used to refer to model elements for which the knowledge of the context is required. An example for this kind of referenced object could be a `PortPrototype`.

⌋

The usage of the references `SecurityEventReportInstanceValue.flatObject` and `SecurityEventReportInstanceValue.object` is mutually exclusive, as explained in [constr_12000].

[constr_12000] Usage of references in the context of SecurityEventReportInstanceValue ⌈Within each SecurityEventReportInstanceValue, exactly one of the references in the role

- flatObject
- object

shall exist.⌋

[TPS_SECXT_03002] Semantics of the reference in the role SecurityEventReportInstanceValue.contextDataElementIdentification ⌈The existence of the reference SecurityEventReportInstanceValue.contextDataElementIdentification implies that the affected SecurityEventReportInstanceValue.id can be used as the concrete identification of the referenced SecurityEventContextDataElement.

This obviously means that the SecurityEventContextDataElement has no directly accessible information about the possible identification it can take. This information can only be retrieved by searching the model for SecurityEventReportInstanceValue that reference the respective SecurityEventContextDataElement.

The direction of the reference is also motivated by the fact that it reaches from a project-specific model element to a model element that can be standardized on the M1 level. A reference in the other direction would significantly counter the chance to standardize a SecurityEventContextDataElement.⌋

It is of course possible that several SecurityEventReportInstanceValues refer to the same SecurityEventContextDataElement, which means that the value transmitted for this SecurityEventContextDataElement can take one of the ids of the referencing SecurityEventReportInstanceValues at run-time.

Class	**SecurityEventReportInstanceValue**			
Note	This class shall be used to provide information for the identification of security-event context data on the value side. **Tags:** atp.recommendedPackage=SecurityEventReportInstanceValues			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
contextData Element Identification	SecurityEventContext DataElement	0..1	ref	This reference contributes to the identification of the context data element. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=contextDataElementIdentification, context DataElementIdentification.securityEventContextData Element, contextDataElementIdentification.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

▽

△

Class	**SecurityEventReportInstanceValue**			
flatObject	Referrable	0..1	ref	This reference shall be used if the referenced object can directly be referenced using a flat reference. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=flatObject
id	PositiveInteger	0..1	attr	This attribute represents the numerical value used for the identification of the context data element.
object	AtpFeature	0..1	iref	This reference shall be used of the target of the reference can only be identified by an instanceRef. **Stereotypes:** atpUriDef **Tags:** atp.Splitkey=object, object.contextElement, object.target **InstanceRef implemented by:** AnyInstanceRef

Table 4.24: SecurityEventReportInstanceValue

[constr_12001] Existence of attribute `SecurityEventReportInstanceValue.id` ⌈For each `SecurityEventReportInstanceValue`, attribute `id` shall exist.⌋

In "normal" cases, instanceRefs are modeled for a specific purpose and therefore only work if the intended purpose applies. But in the case of the instanceRef in the role `SecurityEventReportInstanceValue.object`, the creation of a **strict** modeling for a specific purpose is not possible because there is **only one instanceRef for each possible case**.

As a consequence, the role `object` needs to be very flexible and this requirement can only be fulfilled by one specific kind of instanceRef, the `AnyInstanceRef`.

Class	**AnyInstanceRef**			
Note	Describes a reference to any instance in an AUTOSAR model. This is the most generic form of an instance ref. Refer to the superclass notes for more details.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ApmcInstanceReferenceValue.value, ApmcUpstreamDocInstanceReferenceValue.value, ApmcUriInstanceReferenceValue.value, Collection.collectedInstance, Collection.sourceInstance, DocumentationContext.feature, EcucInstanceReferenceValue.value, FlatInstanceDescriptor.ecuExtractReference, FlatInstanceDescriptor.upstreamReference, RptContainer.byPassPoint, RptHook.rptArHook, SecurityEventReportInstanceValue.object, ViewMap.firstElementInstance, ViewMap.secondElementInstance			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
base	AtpClassifier	0..1	ref	This is the base from which navigation path begins. **Stereotypes:** atpDerived
contextElement (ordered)	AtpPrototype	*	ref	This is one step in the navigation path specified by the instance ref.
target	AtpFeature	0..1	ref	This is the target of the instance ref.

Table 4.25: AnyInstanceRef

Deployments of the `AnyInstanceRef` in the AUTOSAR meta-model are typically accompanied by a semi-formal description of the concrete instanceRef that is intended for the specific purpose.

[TPS_SECXT_03003] Specifying the sequence of context elements in attribute `SecurityEventReportInstanceDefinition.contextTypeList` ⌈A convention

for the semi-formal description of the concrete structural requirements to the usage of the AnyInstanceRef has already been established in [TPS_ECUC_02083] [10]. This convention will be taken over in a modified way for the usage within the attribute contextTypeList.

Instead of just one string that describes the whole collection of context elements, the contextTypeList holds an ordered collection of strings where each element of the collection represents one context element.

The first element of the collection represents the first context element, and so on. Indications for the multiple occurrence of a context element is indicated by adding a '*' to the name of the context element, e.g. SW-COMPONENT-PROTOTYPE*.⌋

[TPS_SECXT_03004] Semantics of meta-class SecurityEventReportInstanceDefinition ⌈Meta-class SecurityEventReportInstanceDefinition represents the opportunity to specify

- the type of model element that shall become a target of a reference within the attribute targetType and
- the structure of the application of an AnyInstanceRef according to [TPS_SECXT_03003] in the attribute contextTypeList.

In other words, the existence of SecurityEventReportInstanceDefinition represents a contract for the creation of a SecurityEventReportInstanceValue, i.e. the types of the model elements referenced from the SecurityEventReportInstanceValue shall match the definition done in the context of the SecurityEventReportInstanceDefinition.

The contract is applied to a SecurityEventContextDataElement by referencing a SecurityEventReportInstanceDefinition in the role SecurityEventContextDataElement.securityEventReportInstanceDefinition.⌋

For example, if a given SecurityEventReportInstanceDefinition sets the attribute targetType to the value CAN-COMMUNICATION-CONNECTOR, the expectation is that a SecurityEventReportInstanceValue that references the SecurityEventReportInstanceDefinition in the role contextDataElementIdentification will refer to a CanCommunicationConnector in the role flatObject.

An AUTOSAR tool can check whether this expectation is fulfilled. But since the contract is based on filling in the value of a string attribute, there is no way to enforce this contract on M2 level by means of a constraint.

Class	**SecurityEventReportInstanceDefinition**
Note	This class shall be used to provide definition-level information for the identification of security-event context data. **Tags:** atp.recommendedPackage=SecurityEventReportInstanceDefinitions

▽

△

Class	SecurityEventReportInstanceDefinition			
Base	ARElement, ARObject, CollectableElement, Identifiable, IdsCommonElement, MultilanguageReferrable, PackageableElement, Referrable, UploadableDesignElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextTypeList (ordered)	String	*	attr	This attribute shall only be used for the case that an instanceRef is required. The attribute identifies the ordered collection of the type of context elements within the instanceRef.
targetType	String	0..1	attr	This attribute shall only be used for the case that an instanceRef is required. The attribute identifies the type of the target element within the instanceRef.

Table 4.26: SecurityEventReportInstanceDefinition

[constr_12002] Existence of attribute `SecurityEventReportInstanceDefinition.targetType` ⌈For each `SecurityEventReportInstanceDefinition`, the attribute `targetType` shall exist.⌋

Obviously, the attribute `SecurityEventReportInstanceDefinition.contextTypeList` is only needed if an instanceRef is involved.

Please find the sketch of a small example for how to use `SecurityEventReportInstanceDefinition` and `SecurityEventReportInstanceValue` in Figure 4.18.

In this example, a security event shall be reported that carries the modelled identification of a `CanCommunicationConnector`. Possible values are '1' and '2'.

A `SecurityEventContextDataElement` may be associated with a run-time information that is not affiliated with an existing model element. In this case the reference in the role `SecurityEventContextDataElement.securityEventReportInstanceDefinition` does not exist and it is consequently not allowed that the `SecurityEventContextDataElement` is referenced from a `SecurityEventReportInstanceValue`. This restriction is formalized in [constr_12003].

[constr_12003] Existence of reference `SecurityEventReportInstanceValue.contextDataElementIdentification` ⌈In the context of a `SecurityEventContextDataElement`, if the reference in the role `SecurityEventContextDataElement.securityEventReportInstanceDefinition` does not exist, then the `SecurityEventContextDataElement` shall not be referenced in the role `SecurityEventReportInstanceValue.contextDataElementIdentification`.⌋

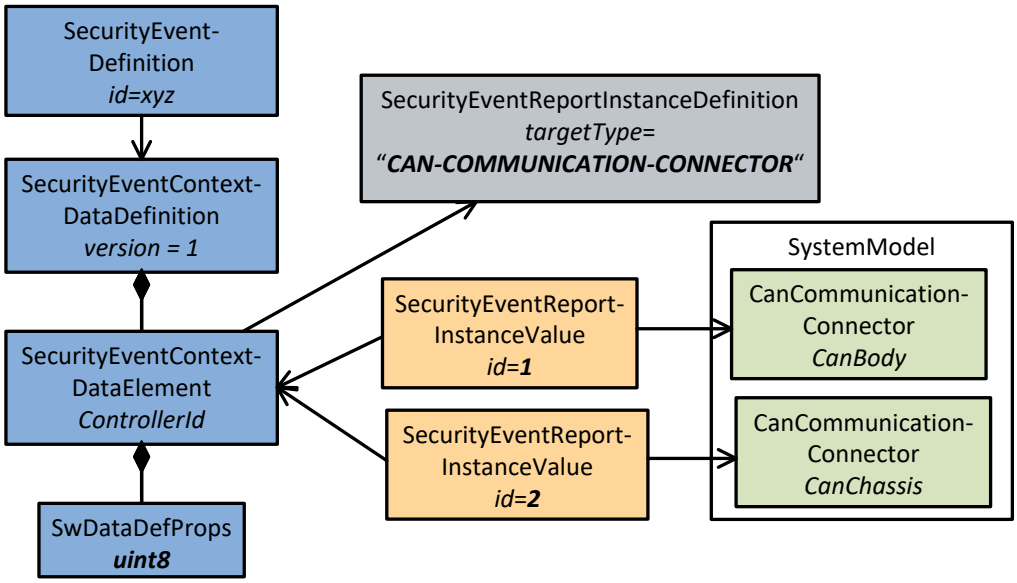


Figure 4.18: Example usage of `SecurityEventReportInstanceDefinition` and `SecurityEventReportInstanceValue`

A Reference Material

This chapter contains some relevant reference material for this specification.

A.1 Terms and Abbreviations

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This chapter includes terms and abbreviations locally relevant within this document. The main list of terms and abbreviations is available in [11]

Term	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 of 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.

Table A.1: Terms

Acronym	Description
IDS	Intrusion Detection System
IdsM	Intrusion Detection System Manager
IdsR	Intrusion Detection System Reporter
SEv	Security Event
QSEv	Qualified Security Event
Sem	Security Event Memory
SIEM	Security Incident and Event Management
SOC	Security Operation Centre
SOP	Start Of Production
SWCL	Software Cluster

Acronym	Description

Table A.2: Acronyms

A.2 Imposition Times of Constraints

The constraints formulated in this document have different *actual* imposition times which denote the steps in the workflow when the respective constraint has to be imposed.

The imposition times that are considered applicable in the scope of this document[1] are listed in Table A.3.

Please note that the imposition times are intentionally rendered as technical terms such that it is possible to link back from each constraint to the definition of the affected imposition time in Table A.3.

Imposition Time	Description	Motivation
IT_SecEx	Security Extract is completed	This imposition time indicates when the Security Extract is complete.

Table A.3: Imposition Times of constraints in this document

[1] Different imposition times may be defined in the context of other AUTOSAR standard documents

B Security Events in Use

The following Table B.1 lists all `SecurityEventDefinition` used in AUTOSAR.

SEv Id	*Name*	*Description*	*Raised by the following modules*	*Context Data Version*
Communication SEvs				
6	SEV_DROP_PDU_RX_UDP	SoAd dropped a PDU. The PDU violates stack configuration and was received via a UDP socket.	CP: SoAd AP: n/a	n/a
7	SEV_DROP_MSG_RX_UDP_LENGTH	SoAd dropped a message. The message contains at least one PDU which violates stack configuration and was received via a UDP socket. The violation relates to the length of the PDUs compared to the overall length of the message.	CP: SoAd AP: n/a	n/a
8	SEV_DROP_MSG_RX_UDP_SOCKET	SoAd received a UDP message which violates stack configuration and was dropped. No suitable socket connection matching to configuration was found.	CP: SoAd AP: n/a	n/a
9	SEV_REJECTED_TCP_CONNECTION	SoAd rejected a TCP connection. The connection request violates stack configuration.	CP: SoAd AP: n/a	n/a
10	SEV_ARP_IP_ADDR_CONFLICT	Received local IP address in ARP reply for different MAC.	CP: TcpIp AP: core	1
11	SEV_TCP_DROP_INV_DEST_PORT	Dropped TCP packet because of invalid destination TCP-Port.	CP: TcpIp AP: core	1
12	SEV_UDP_DROP_INV_DEST_PORT	Dropped UDP packet because of invalid destination UDP-Port.	CP: TcpIp AP: core	1
13	SEV_IPV4_DROP_INV_DEST_ADDR	Dropped datagram because of invalid destination IPV4 address.	CP: TcpIp AP: core	1
14	SEV_IPV6_DROP_INV_DEST_ADDR	Dropped datagram because of invalid destination IPV6 address.	CP: TcpIp AP: core	1
15	SEV_ETH_DROP_UNKNOWN_ETHERTYPE	An ethernet datagram was dropped due the Ethertype is not known.	CP: EthIf AP: core	1
16	SEV_ETH_DROP_VLAN_DOUBLE_TAG	An ethernet datagram was dropped due to double VLAN tag.	CP: EthIf AP: core	1
17	SEV_ETH_DROP_INV_VLAN	An ethernet datagram was dropped due to an invalid CrtlIdx/VLAN.	CP: EthIf AP: core	1
18	SEV_ETH_DROP_MAC_COLLISION	Ethernet datagram was dropped because local MAC was same as source MAC in an incoming frame.	CP: EthIf AP: core	1
19	SEV_CAN_TX_ERROR_DETECTED	A transmission related error was detected. Depending on the context data this could indicate suspicious CAN activity.	CP: CanIf AP: n/a	1
20	SEV_CAN_RX_ERROR_DETECTED	A reception related error was detected. Depending on the context data this could indicate suspicious CAN activity.	CP: CanIf AP: n/a	1
21	SEV_CAN_ERRORSTATE_PASSIVE	The CAN controller transitioned to state passive.	CP: CanIf AP: n/a	1
22	SEV_CAN_ERRORSTATE_BUSOFF	The CAN controller transitioned to state busoff.	CP: CanIf AP: n/a	1

▽

△

SEv Id	Name	Description	Raised by the following modules	Context Data Version
44	SEV_SECOC_MAC_VERIFICATION_FAILED	MAC verification of a received PDU failed.	CP: SecOC AP: com	1
45	SEV_SECOC_FRESHNESS_NOT_OK	Failed to get freshness value from Fv M upon reception of a SecOC secured PDU. Depending on the freshness value management, this can be an indicator of a replay attack.	CP: SecOC AP: com	1
50	SEV_DROP_PDU_RX_TCP	SoAd dropped a PDU. The PDU violates stack configuration and was received via a TCP socket.	CP: SoAd AP: n/a	n/a
51	SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH	A network packet was blocked due to a rule mismatch on IPv4 layer.	CP: Fw AP: fw	2
52	SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH	A network packet was blocked due to a rule mismatch on IPv6 layer.	CP: Fw AP: fw	2
53	SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH	A network packet was blocked due to a rule mismatch within the ICMP protocol.	CP: Fw AP: fw	2
54	SEV_FW_PACKET_BLOCKED_TCP_MISMATCH	A network packet was blocked due to a rule mismatch on TCP layer.	CP: Fw AP: fw	2
55	SEV_FW_PACKET_BLOCKED_UDP_MISMATCH	A network packet was blocked due to a rule mismatch on UDP layer.	CP: Fw AP: fw	2
56	SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH	A network packet was blocked due to a rule mismatch in the SOME/IP protocol.	CP: Fw AP: fw	2
57	SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH	A network packet was blocked due to a rule mismatch in the SOME/IP SD protocol.	CP: Fw AP: fw	2
58	SEV_FW_PACKET_BLOCKED_DDS_MISMATCH	A network packet was blocked due to a rule mismatch in the DDS-RTPS protocol.	CP: Fw AP: fw	2
59	SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH	A network packet was blocked due to a rule mismatch in the DoIP protocol.	CP: Fw AP: fw	2
60	SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH	A network packet was blocked due to a rule mismatch on generic inspection level.	CP: Fw AP: fw	2
61	SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS	A network packet was blocked due to the maximal number of open TCP connections was reached.	CP: Fw AP: fw	2
62	SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT	A network packet was blocked due to TCP timeout.	CP: Fw AP: fw	2
63	SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION	A network packet was blocked due to an invalid TCP state transition.	CP: Fw AP: fw	2
64	SEV_FW_PACKET_BLOCKED_RATELIMIT	A network packet was blocked due to the rate limit was reached.	CP: Fw AP: fw	2
66	SEV_TSYN_CAN_ICV_GENERATION_FAILED	ICV generation for a FUP message has failed.	CP: CanTSyn AP: n/a	1
67	SEV_TSYN_CAN_ICV_VERIFICATION_FAILED	ICV verification of a FUP message has failed.	CP: CanTSyn AP: n/a	1
68	SEV_TSYN_CAN_FRESHNESS_NOT_AVAILABLE	Failed to get freshness value from Fv M.	CP: CanTSyn AP: n/a	1

▽

△

SEv Id	Name	Description	Raised by the following modules	Context Data Version
69	SEV_TSYN_CAN_MSG_SEQUENCE_ERROR	Failed to receive correct sequence of SYNC and FUP from the TimeMaster within (CanTSynGlobalTimeFollowUp Timeout).	CP: CanTSyn AP: n/a	1
70	SEV_TSYN_FR_ICV_GENERATION_FAILED	ICV generation for a Sync message has failed.	CP: FrTSyn AP: n/a	1
71	SEV_TSYN_FR_ICV_VERIFICATION_FAILED	ICV verification of a received Sync message has failed.	CP: FrTSyn AP: n/a	1
72	SEV_TSYN_FR_FRESHNESS_NOT_AVAILABLE	Failed to get freshness value from Fv M.	CP: FrTSyn AP: n/a	1
73	SEV_TSYN_ETH_ICV_GENERATION_FAILED	ICV generation for a Follow_Up message failed.	CP: EthTSyn AP: tsync	1
74	SEV_TSYN_ETH_ICV_VERIFICATION_FAILED	ICV verification of a received Follow_Up message failed.	CP: EthTSyn AP: tsync	1
75	SEV_TSYN_ETH_FRESHNESS_NOT_AVAILABLE	Failed to get freshness value from Fv M.	CP: EthTSyn AP: tsync	1
76	SEV_TSYN_ETH_MSG_SEQUENCE_ERROR	Failed to receive correct sequence of SYNC and FUP from the TimeMaster within (EthTSynGlobalTimeFollowUp Timeout).	CP: EthTSyn AP: tsync	1
77	SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH	A network packet was blocked due to a rule mismatch on data link layer.	CP: Fw AP: fw	2
78	SEV_MKA_AUTHENTICATION_FAILURE	Event triggered when the authentication during the MKA communication has failed (wrong CKN/CAK).	CP: Mka AP: core	1
79	SEV_MKA_TIMEOUT	Event triggered when the timeout for the MKA communication has expired.	CP: Mka AP: core	1
80	SEV_MKA_PORT_NOT_ENABLED	Event triggered when the indicated port for the MKA communication is not enable.	CP: Mka AP: core	1
81	SEV_MKA_CIPHER_SUITE_NOT_SUPPORTED	Event triggered when there is no Cipher Suite supported.	CP: Mka AP: core	1
82	SEV_MKA_PORT_NUMBER_CHANGE	Event triggered when during the MKA communication the port number has changed.	CP: Mka AP: core	1
83	SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING	A network packet was blocked due to per-stream filtering in the switch.	CP: Fw AP: n/a	1
89	SEV_COM_RX_SIGNAL_VALUE_UNEXPECTED	Signal or group signal is received with unexpected value.	CP: Com AP: n/a	1
90	SEV_TLS_ERROR	An alert message (warning or fatal) was detected (either received or generated) by TLS.	CP: TcpIp AP: com	2
91	SEV_TLS_CONNECTION_ESTABLISHED	A TLS connection was successfully established.	CP: TcpIp AP: com	2
92	SEV_TLS_CONNECTION_CLOSED	A TLS connection was closed normally.	CP: TcpIp AP: com	2
Cryptography SEvs				
5	SEV_CERT_CHAIN_VERIFICATION_FAILED	The verification of a certificate against a certificate chain was not successful.	CP: KeyM AP: crypto	2
95	SEV_CERT_INSTALL	Attempt to install a new certificate.	CP: KeyM AP: crypto	1

▽

△

SEv Id	Name	Description	Raised by the following modules	Context Data Version
96	SEV_CERT_UPDATE	Attempt to update a certificate.	CP: KeyM AP: crypto	1
97	SEV_CERT_DELETE	Attempt to delete a certificate.	CP: KeyM AP: crypto	1
98	SEV_CERT_INSTALLED_BUT_INVALID	An already installed certificate is invalid.	CP: KeyM AP: n/a	1
Diagnostic SEvs				
100	SEV_UDS_SECURITY_ACCESS_NEEDED	Tester has sent a diagnostic request without meeting the server's security level requirements for that service. NRC 0x33 (securityAccessDenied) was returned.	CP: Dcm AP: diag	1
101	SEV_UDS_AUTHENTICATION_NEEDED	A diagnostic request was received while the required authentication to execute this service is not given. NRC 0x34 (authenticationRequired) was returned.	CP: Dcm AP: diag	1
102	SEV_UDS_SECURITY_ACCESS_SUCCESSFUL	Successful unlocked the ECU (via Security Access SID 0x27)	CP: Dcm AP: diag	1
103	SEV_UDS_SECURITY_ACCESS_FAILED	Unlocking of the ECU (via Security Access SID 0x27) failed	CP: Dcm AP: diag	1
104	SEV_UDS_AUTHENTICATION_SUCCESSFUL	Succesfully authenticated (via Authentication SID 0x29)	CP: Dcm AP: diag	1
105	SEV_UDS_AUTHENTICATION_FAILED	Authentication (via Authentication SID 0x29) failed	CP: Dcm AP: diag	1
106	SEV_UDS_WRITE_DATA_SUCCESSFUL	Diagnostic data identifier has been written by SID 0x2E WriteDataBy Identifier	CP: Dcm AP: diag	1
107	SEV_UDS_WRITE_DATA_FAILED	Change of Diagnostic data identifier has been requested by SID 0x2E WriteDataByIdentifier, but failed	CP: Dcm AP: diag	1
108	SEV_UDS_WRITE_MEMORY_SUCCESSFUL	Data has been written into memory by SID 0x3D WriteMemoryByAddress	CP: Dcm AP: n/a	2
109	SEV_UDS_WRITE_MEMORY_FAILED	Writting of Data into memory has been requested by SID 0x3D WriteMemory ByAddress, but failed	CP: Dcm AP: n/a	2
110	SEV_UDS_REQUEST_UP_DOWNLOAD_SUCCESSFUL	An upload / download sequence has been requested successfully with SID 0x34 or SID 0x35	CP: Dcm AP: diag	2
111	SEV_UDS_REQUEST_UP_DOWNLOAD_FAILED	An upload / download sequence has been requested with SID 0x34 or SID 0x35, but failed	CP: Dcm AP: diag	2
112	SEV_UDS_REQUEST_FILE_TRANSFER_SUCCESSFUL	A file transfer sequence has been requested successfully with SID 0x38.	CP: Dcm AP: diag	1
113	SEV_UDS_REQUEST_FILE_TRANSFER_FAILED	A file transfer sequence has been requested with SID 0x38, but failed	CP: Dcm AP: diag	1
114	SEV_UDS_COMMUNICATION_CONTROL_SUCCESSFUL	The control of a communication has been requested by service SID 0x28 CommunicationControl successfully.	CP: Dcm AP: diag	1
115	SEV_UDS_COMMUNICATION_CONTROL_FAILED	The control of a communication has been requested by service SID 0x28 CommunicationControl, but failed.	CP: Dcm AP: diag	1

▽

△

SEv Id	Name	Description	Raised by the following modules	Context Data Version
116	SEV_UDS_CLEAR_DTC_SUCCESSFUL	DTC information has been cleared by SID 0x14 ClearDiagnosticInformation.	CP: Dcm AP: diag	1
117	SEV_UDS_CLEAR_DTC_FAILED	Clearing DTC information has been requested by SID 0x14 Clear DiagnosticInformation, but failed.	CP: Dcm AP: diag	1
118	SEV_UDS_CONTROL_DTC_SETTING_SUCCESSFUL	The control of a DTC setting has been requested by service SID 0x85 ControlDTCSetting successfully.	CP: Dcm AP: diag	1
119	SEV_UDS_CONTROL_DTC_SETTING_FAILED	Control of DTC setting has been requested by service SID 0x85 ControlDTCSetting, but failed.	CP: Dcm AP: diag	1
120	SEV_UDS_ECU_RESET_SUCCESSFUL	ECU has been reset by SID 0x11 ECUReset.	CP: Dcm AP: diag	1
121	SEV_UDS_ECU_RESET_FAILED	ECU Reset has been requested by SID 0x11 ECUReset, but failed.	CP: Dcm AP: diag	1
122	SEV_UDS_ROUTINE_CONTROL_SUCCESSFUL	The control of a routine has been requested by service SID 0x31 RoutineControl successfully.	CP: Dcm AP: diag	1
123	SEV_UDS_ROUTINE_CONTROL_FAILED	The control of a routine has been requested by service SID 0x31 RoutineControl, but failed.	CP: Dcm AP: diag	1
124	SEV_UDS_IO_CONTROL_SUCCESSFUL	IOControl operation has been requested by service SID 0x2F Input OutputControlByIdentifier successfully.	CP: Dcm AP: n/a	1
125	SEV_UDS_IO_CONTROL_FAILED	IOControl operation has been requested by service SID 0x2F Input OutputControlByIdentifier, but failed.	CP: Dcm AP: n/a	1
127	SEV_DOIP_HEADER_CHECK_FAILED	The DoIP Header Handler rejected a request (routing or diagnostic message).	CP: DoIP AP: diag	1
128	SEV_DOIP_ROUTING_ACTIVATION_CHECK_FAILED	A routing request was rejected by the routing handler.	CP: DoIP AP: diag	1
129	SEV_DOIP_ROUTING_ACTIVATION_SUCCESS	A routing request was successful.	CP: DoIP AP: diag	1
130	SEV_DOIP_DIAG_MESSAGE_CHECK_FAILED	A diagnostic message request was rejected by the diagnostic message handler.	CP: DoIP AP: diag	1
Identity and Access Management SEvs				
65	SEV_ACCESS_CONTROL_PHM_IAM_ACCESS_DENIED	Access of an application to a resource provided by Platform Health Management was denied.	CP: n/a AP: phm	1
131	SEV_ACCESS_CONTROL_FIREWALL_IAM_ACCESS_DENIED	Access of an application to a resource provided by the firewall was denied.	CP: n/a AP: fw	1
133	SEV_ACCESS_CONTROL_DM_IAM_ACCESS_DENIED	Access of an application to a resource provided by DM was denied.	CP: n/a AP: diag	1
134	SEV_ACCESS_CONTROL_CRYPTO_IAM_ACCESS_DENIED	Access of an application to a resource provided by Cryptography was denied.	CP: n/a AP: crypto	1
135	SEV_ACCESS_CONTROL_COM_IAM_ACCESS_DENIED	Access of an application to a resource provided by Communication Management was denied.	CP: n/a AP: com	1

▽

△

SEv Id	Name	Description	Raised by the following modules	Context Data Version
136	SEV_ACCESS_CONTROL_IDSM_IAM_ACCESS_DENIED	Access of an application to a resource provided by Intrusion Detection System Management was denied.	CP: n/a AP: idsm	1
137	SEV_ACCESS_CONTROL_SM_IAM_ACCESS_DENIED	Access of an application to a resource provided by State Management was denied.	CP: n/a AP: sm	1
IdsM SEvs				
46	SEV_IDSM_NO_EVENT_BUFFER_AVAILABLE	A SEv cannot be handled because there are no more event buffers available to process the event.	CP: IdsM AP: idsm	n/a
47	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.	CP: IdsM AP: idsm	n/a
49	SEV_IDSM_COMMUNICATION_ERROR	An error occurred when sending a QSEv via PDU.	CP: IdsM AP: idsm	n/a
87	SEV_IDSM_NO_QUALIFIED_EVENT_BUFFER_AVAILABLE	A security event raised when a QSEv has to be dropped due to insufficient QSEv buffers available.	CP: IdsM AP: idsm	n/a
126	SEV_IDSM_STATISTICS	A SEV reported by application to snapshot the IdsM statistics data.	CP: IdsM AP: n/a	1
SOMEIP SEvs				
138	SEV_SOMEIP_ERROR	The processing of a SOME/IP message resulted in a return not equal to E_OK.	CP: SomeIpXf AP: com	1
139	SEV_SOMEIP_UNAUTH_SERVICE_OFFER	A service offer was received from a server that is not authorized to offer it, or a service offer was received for a service that has already been offered by a different server.	CP: Sd AP: com	1
140	SEV_SOMEIP_SERVICE_NOT_AVAILABLE	A client invoked a method or attempted to subscribe to a eventgroup that does not belong to a currently offered service.	CP: Sd AP: com	1
141	SEV_SOMEIPSD_PDU_MALFORMED	The SOME/IP-SD length information of a received PDU violates the protocol definition.	CP: Sd AP: com	1
142	SEV_SOMEIP_UNAUTH_SUBSCRIPTION_REQ	A client attempted to subscribe to an eventgroup for which it is not authorized.	CP: Sd AP: com	1
143	SEV_SOMEIP_UNAUTH_METHOD_INVOCATION	A client attempted to invoke a method for which it is not authorized.	CP: Sd AP: com	1
SW Update SEvs				
93	SEV_SW_UPDATE_FAILED	A SW update operation was requested, but it was not successful.	CP: n/a AP: ucm	1
94	SEV_SW_UPDATE_SUCCESS	A SW update operation was executed successfully.	CP: n/a AP: ucm	1
Secure Boot SEvs				
99	SEV_EXEC_SW_COMPONENT_INTEGRITY_CHECK_FAILED	The integrity check of a SW component has failed.	CP: Crypto AP: core	1

Table B.1: Security Events

C Examples

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This chapter contains a collection of selected examples that reflect concepts described in different chapters of this document.

C.1 Example for modeling of SecurityEventContextDataDefinition

The following two examples are showing the modeling of a SecurityEventDefinition named CANIF_SEV_RX_ERROR_DETECTED that references two SecurityEventContextDataDefinitions:

- version = 1 that is showed in Figure C.1 and
- version = 2 that is showed in Figure C.2

Both versions define two SecurityEventContextDataElements named *ControllerId* and *CanError*. The networkRepresentations of *ControllerId* is the same in both versions. The networkRepresentations of *CanError* defines an enumeration (modeled as a CompuMethod with category TEXTTABLE). In version = 1 the enumeration contains three literals: CAN_ERROR_CHECK_CRC_FAILED, CAN_ERROR_BUS_LOCK, CAN_ERROR_CHECK_FORM_FAILED. In version = 2 the enumeration contains only two of those literals.

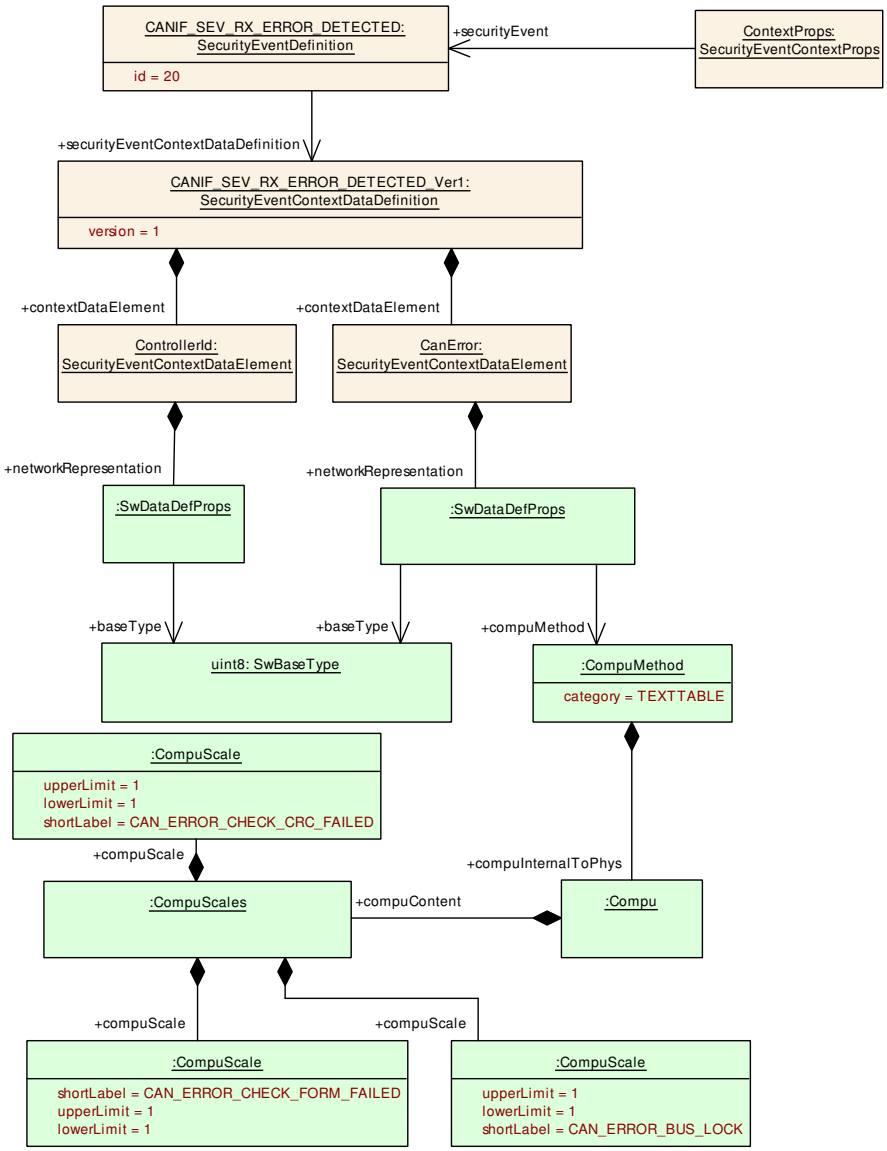


Figure C.1: Modeling of `SecurityEventContextDataDefinition` with version = 1

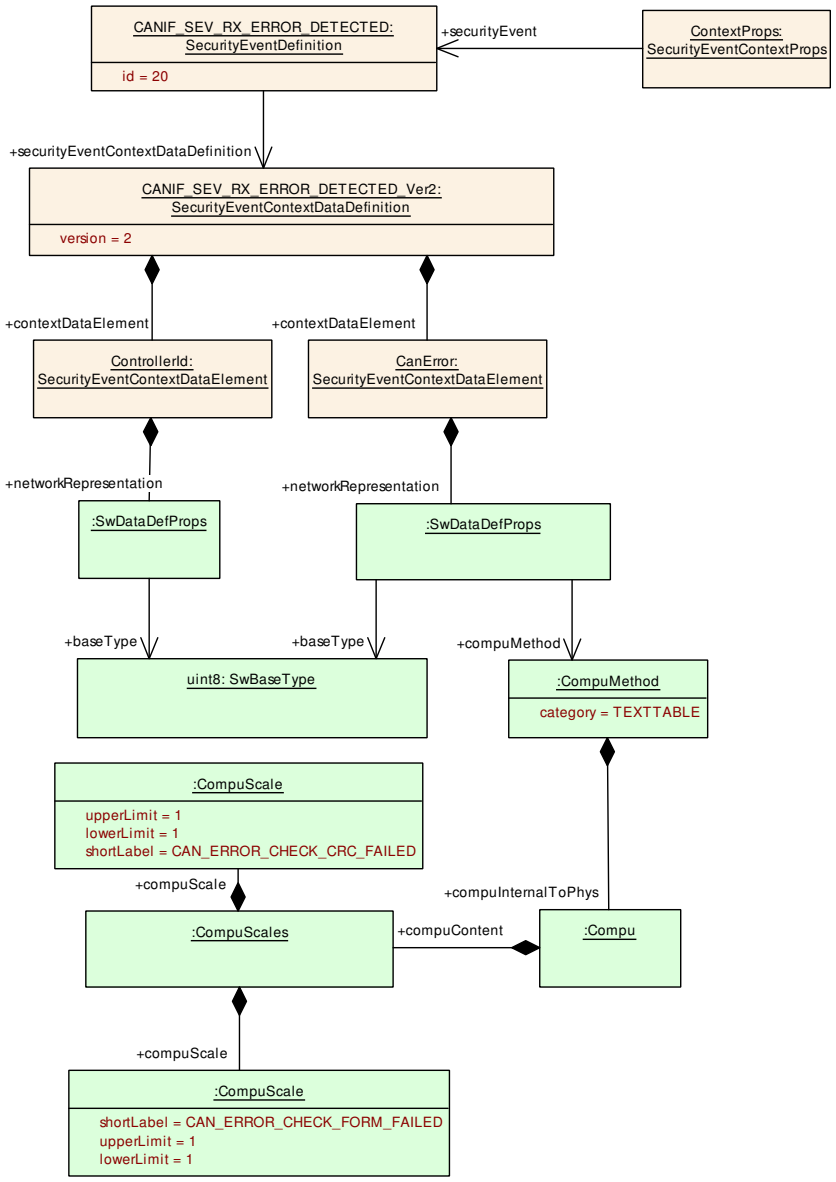


Figure C.2: Modeling of `SecurityEventContextDataDefinition` with version = 2

D Mentioned Class Tables

This chapter contains the remaining set of meta-class tables which are not shown directly in the main body of this document.

Class	***AUTOSAR*** (abstract)			
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractMessageToSecureChannelMapping*, AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicabilityInfoSet, *AutosarDataType*, *BaseType*, BlueprintMappingSet, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, DataConstr, DataTransformationSet, DataTypeMappingSet, DdsCpConfig, *DiagnosticCommonElement*, DiagnosticConnection, DiagnosticContributionSet, DltArgumentPropsSet, DltContext, DltEcu, Documentation, E2EProfileCompatibilityProps, EthIpProps, EthTcpIpIcmpProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FirewallRule, GeneralPurposeConnection, HwCategory, HwElement, HwType, *IEEE1722TpConnection*, IPSecConfigProps, *IdsCommonElement*, IdsDesign, ImpositionTimeDefinitionGroup, InterpolationRoutineMappingSet, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKayProps, MacSecParticipantSet, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, PhysicalDimension, PhysicalDimensionMappingSet, *PlatformModuleEndpointConfiguration*, *PortInterface*, PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, *SecureComProps*, *SecurityConnectivityAssociation*, SignalServiceTranslationPropsSet, SomeipSdClientEventGroupTimingConfig, SomeipSdClientServiceInstanceConfig, SomeipSdServerEventGroupTimingConfig, SomeipSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, *SwComponentType*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, System, SystemComSpecDefinitionSet, SystemSignal, SystemSignalGroup, *TimingExtension*, TlvDataIdDefinitionSet, TransformationPropsSet, Unit, UnitGroup, *UploadablePackageElement*, ViewMapSet			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table D.1: AUTOSAR

Class	**ARPackage**			
Note	AUTOSAR package, allowing to create top level packages to structure the contained AUTOSARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM.			
Base	*ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ARPackage.arPackage, AUTOSAR.arPackage			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, arPackage.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30

▽

△

Class	**ARPackage**			
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpVariation **Tags:** atp.Splitkey=element, element.shortName, element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	aggr	This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. **Tags:** atp.Splitkey=referenceBase, referenceBase.shortLabel xml.sequenceOffset=10

Table D.2: ARPackage

Class	**AUTOSAR**			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:** xml.globalElement=true			
Base	*ARObject*			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=arPackage, arPackage.shortName, arPackage.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Stereotypes:** atpStructuredComment **Tags:** atp.Splitkey=fileInfoComment xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=20

Table D.3: AUTOSAR

Class	**BaseTypeDirectDefinition**			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Aggregated by	*BaseType*.baseTypeDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
baseType Encoding	BaseTypeEncoding String	0..1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **Tags:** xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. **Tags:** xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. **Tags:** xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". **Tags:** xml.sequenceOffset=100
native Declaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE. If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size. This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems. **Tags:** xml.sequenceOffset=120

Table D.4: BaseTypeDirectDefinition

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

▽

△

Class	**CanCommunicationConnector**			
pncWakeup DataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDlc	PositiveInteger	0..1	attr	Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.

Table D.5: CanCommunicationConnector

Class	***CommunicationConnector*** (abstract)			
Note	The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunication Connector, UserDefinedCommunicationConnector			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
commController	Communication Controller	0..1	ref	Reference to the communication controller. The CommunicationConnector and referenced CommunicationController shall be aggregated by the same ECUInstance. The communicationController can be referenced by several CommunicationConnector elements. This is important for the FlexRay Bus. FlexRay communicates via two physical channels. But only one controller in an ECU is responsible for both channels. Thus, two connectors (for channel A and for channel B) shall reference to the same controller. **Tags:** atp.Splitkey=commController
createEcu WakeupSource	Boolean	0..1	attr	If this parameter is available and set to true then a channel wakeup source shall be created for the Physical Channel referencing this CommunicationConnector.
explicitWakeup Channel	PhysicalChannel	*	ref	Defines a restriction which PhysicalChannels shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupChannel, explicitWakeup Channel.physicalChannel, explicitWakeup Channel.variationPoint.shortLabel vh.latestBindingTime=postBuild
explicitWakeup Pnc	PncMappingIdent	*	ref	Defines a restriction which PNCs shall be woken up if this CommunicationConnector is the wakeup source. If not defined, then no restriction applies. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=explicitWakeupPnc, explicitWakeupPnc.pnc MappingIdent, explicitWakeupPnc.variationPoint.short Label vh.latestBindingTime=postBuild
nmVariant	NmVariantEnum	0..1	attr	Defines the functionality of Network Management.
pncFilterArray Mask (ordered)	PositiveInteger	*	attr	Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.

Table D.6: CommunicationConnector

Class	CompuMethod			
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant. **Tags:** atp.recommendedPackage=CompuMethods			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuInternalToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** atp.Splitkey=compuInternalToPhys xml.sequenceOffset=80
compuPhysToInternal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:** atp.Splitkey=compuPhysToInternal xml.sequenceOffset=90
displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. **Tags:** xml.sequenceOffset=20
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. **Tags:** atp.Splitkey=unit xml.sequenceOffset=30

Table D.7: CompuMethod

Class	CryptoServiceKey			
Note	This meta-class has the ability to represent a crypto key. **Tags:** atp.recommendedPackage=CryptoDevelopmentKeys			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
algorithmFamily	String	0..1	attr	This attribute represent the description of the family of the applicable crypto algorithm.
developmentValue	ValueSpecification	0..1	aggr	This aggregation represents the ability to assign a specific value to the crypto key as part of the system description. This value can then be taken for the development of the respective ECU. **Tags:** atp.Splitkey=developmentValue
keyGeneration	CryptoServiceKeyGenerationEnum	0..1	attr	This attribute describes how a the specific cryptographic key is created.
keyStorageType	String	0..1	attr	This attribute describes where the enclosing cryptographic key shall be stored. AUTOSAR reserves specific values for this attributes but it is possible to insert custom values as well.
length	PositiveInteger	0..1	attr	This attribute describes the length of the cryptographic key in bits.

Table D.8: CryptoServiceKey

Class	CryptoServicePrimitive			
Note	This meta-class has the ability to represent a crypto primitive. **Tags:** atp.recommendedPackage=CryptoPrimitives			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
algorithmFamily	String	0..1	attr	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.
algorithmMode	String	0..1	attr	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.
algorithm SecondaryFamily	String	0..1	attr	This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive. The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.

Table D.9: CryptoServicePrimitive

Class	EcuInstance			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description. **Tags:** atp.recommendedPackage=EcuInstances			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
associatedConsumedProvidedServiceInstanceGroup	ConsumedProvidedServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=associatedConsumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstanceGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU. **Tags:** atp.Splitkey=associatedPdurIPduGroup
channelSynchronousWakeup	Boolean	0..1	attr	If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication. **Tags:** atp.Splitkey=clientIdRange

△

Class	**EcuInstance**			
commController	Communication Controller	*	aggr	CommunicationControllers of the ECU. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=commController, commController.shortName, commController.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. **Stereotypes:** atpVariation **Tags:** atp.Splitkey=connector, connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of an EcuInstance. **Tags:** atp.Splitkey=firewallRule
pncNmRequest	Boolean	0..1	attr	Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.

Table D.10: EcuInstance

Class	**GeneralPurposeIPdu**			
Note	This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **Tags:** atp.recommendedPackage=Pdus			
Base	*ARElement*, *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Pdu*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table D.11: GeneralPurposeIPdu

Class	***Identifiable*** (abstract)
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	***Identifiable*** (abstract)			
Subclasses	ARPackage, *AbstractDoIpLogicAddressProps*, *AbstractEvent*, *AbstractFunctionalClusterDesign*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstance Filter*, *AbstractServiceInstance*, ApplicationEndpoint, ApplicationError, AppliedStandard, Artifact Checksum, *AtpBlueprint*, *AtpClassifier*, *AtpFeature*, AutosarOperationArgumentInstance, Autosar VariableInstance, BlockState, *BuildActionEntity*, BuildActionEnvironment, CanSecSecureChannel, Chapter, ClientIdDefinition, ClientServerOperation, Code, *CollectableElement*, ComManagement Mapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, *CouplingPortAbstractShaper*, *CouplingPort StructuralElement*, CryptoKeySlot, *CryptoServiceMapping*, DataPrototypeGroup, DataPrototype TransformationPropsIdent, DataTransformation, *DdsAbstractServiceInstanceElementCp*, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, DiagnosticExtendedDataRecordElement, DiagnosticFunctionInhibit Source, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, Dlt ArgumentProps, DltMessage, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, Ethernet WakeupSleepOnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeature Relation, FMFeatureRestriction, FMFeatureSelection, FlexrayTpPduPool, *FrameTriggering*, GlobalTime Gateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttributeDef, HwAttributeLiteralDef, Hw Pin, HwPinGroup, *IEEE1722TpAcfBus*, *IEEE1722TpAcfBusPart*, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignalTriggering, *IdentCaption*, ImpositionTime, InternalTriggeringPoint, Keyword, LifeCycleState, Linker, MacAddressVlanMembership, MacMulticastGroup, MacSecKay Participant, McDataInstance, MemorySection, ModeDeclaration, ModeDeclarationMapping, ModeSwitch Point, ModeSwitchSenderComSpecProps, NetworkEndpoint, *NmCluster*, *NmNode*, *Packageable Element*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, Per InstanceMemory, *PhysicalChannel*, PortGroup, *PortInterfaceMapping*, QueuedReceiverComSpecProps, ResourceConsumption, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutable Entity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntity Group, *SdgAttribute*, SdgClass, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, ServerComSpecProps, *ServiceNeeds*, SignalServiceTranslationEventProps, SignalServiceTranslation Props, SocketAddress, SomeipTpChannel, *StackUsage*, StaticSocketConnection, StructuredReq, Sw GenericAxisParamType, SwServiceArg, SwcServiceDependency, SystemMapping, *TimeBaseResource*, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, Timing ExtensionResource, TimingModeInstance, Topic1, TpAddress, TraceableTable, TraceableText, *Traced Failure*, TransformationISignalPropsIdent, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, WaitPoint			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-40
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. **Tags:** atp.Splitkey=annotation xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. **Tags:** xml.sequenceOffset=-50

▽

△

Class	***Identifiable*** (abstract)			
desc	MultiLanguageOverview Paragraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". **Tags:** atp.Splitkey=desc xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:** atp.Splitkey=introduction xml.sequenceOffset=-30
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp. **Tags:** xml.attribute=true

Table D.12: Identifiable

Class	***IdsCommonElement*** (abstract)			
Note	This meta-class represents a common base class for IDS related elements of the Security Extract. It does not contribute any specific functionality other than the ability to become the target of a reference.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	*IdsMapping*, IdsmInstance, IdsmProperties, SecurityEventContextDataDefinition, SecurityEvent Definition, SecurityEventFilterChain, SecurityEventReportInstanceDefinition, SecurityEventReport InstanceValue			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table D.13: IdsCommonElement

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 D.14: IdsFunctionalClusterDesign

Class	*IdsMapping* (abstract)			
Note	This meta-class serves as abstract base class for mappings related to an IDS design.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Subclasses	*SecurityEventContextMapping*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table D.15: IdsMapping

Class	IdsmModuleInstantiation			
Note	This meta-class defines the attributes for the IdsM configuration on a specific machine. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *AdaptiveModuleInstantiation*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *IdsPlatformInstantiation*, *MultilanguageReferrable*, *NonOsModuleInstantiation*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, Machine.moduleInstantiation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
idsmInstance	IdsmInstance	0..1	ref	This reference identifies the IdsmInstance that is associated with the enclosing IdsmModuleInstantiation. **Tags:** atp.Splitkey=idsmInstance
queueSize	PositiveInteger	0..1	attr	Length of the queue in which IDS events can be stored before processing, if immediate processing is not possible.

Table D.16: IdsmModuleInstantiation

Class	IdsmProperties			
Note	This meta-class provides the ability to aggregate filters for security events. **Tags:** atp.recommendedPackage=IdsMPropertiess			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *UploadableDesignElement*, *UploadablePackageElement*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	IdsmProperties			
rateLimitation Filter	IdsmRateLimitation	*	aggr	This aggregation represents the collection of rate limitation filters for security events in the enclosing SecurityFilterSet. **Tags:** atp.Splitkey=rateLimitationFilter, rateLimitation Filter.shortName
trafficLimitation Filter	IdsmTrafficLimitation	*	aggr	This aggregation represents the collection of traffic limitation filters for security events in the enclosing SecurityFilterSet. **Tags:** atp.Splitkey=trafficLimitationFilter, trafficLimitation Filter.shortName

Table D.17: IdsmProperties

Class	IdsmSignatureSupportCp			
Note	This meta-class defines, for the Classic Platform, the cryptographic algorithm and key to be used by the IdsM instance for providing signature information in QSEv messages. This Class is only used by the AUTOSAR Classic Platform.			
Base	*ARObject*			
Aggregated by	IdsmInstance.signatureSupportCp			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
authentication	CryptoServicePrimitive	0..1	ref	This reference dennotes the cryptographic primitives for providing authentication information in QSEv messages. **Tags:** atp.Splitkey=authentication
cryptoService Key	CryptoServiceKey	0..1	ref	This reference denotes the cryptographic key to be used by the cryptographic algorithm for providing authentication information in QSEv messages. **Tags:** atp.Splitkey=cryptoServiceKey

Table D.18: IdsmSignatureSupportCp

Class	MultiLanguageOverviewParagraph			
Note	This is the content of a multilingual paragraph in an overview item.			
Base	*ARObject*			
Aggregated by	Caption.desc, CompuScale.desc, *Describable*.desc, *Identifiable*.desc, LabeledItem.itemLabel, Modification.change, Modification.reason, SdgCaption.desc, SwRecordLayoutGroup.desc, SwRecord LayoutV.desc, VariationPoint.desc			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
l2	LOverviewParagraph	1..*	aggr	This represents the text in one particular language. **Tags:** atp.Splitkey=l2 xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table D.19: MultiLanguageOverviewParagraph

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
remoteConfig	RemoteEndpoint Configuration	*	aggr	Defintion of remote addresses of peers. **Tags:** atp.Splitkey=remoteConfig

Table D.20: PlatformModuleEthernetEndpointConfiguration

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**
–	–	–	–	–

Table D.21: PortPrototype

Class	*Referrable* (abstract)			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	*ARObject*			
Subclasses	*AtpDefinition*, BswDistinguishedPartition, *BswModuleCallPoint*, BswModuleClientServerEntry, BswVariableAccess, CouplingPortTrafficClassAssignment, *DiagnosticEnvModeElement*, EthernetPriorityRegeneration, ExclusiveAreaNestingOrder, *HwDescriptionEntity*, *ImplementationProps*, ModeTransition, *MultilanguageReferrable*, PncMappingIdent, *SingleLanguageReferrable*, SoConIPduIdentifier, TpConnectionIdent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. **Stereotypes:** atpIdentityContributor **Tags:** xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:** atp.Splitkey=shortNameFragment xml.sequenceOffset=-90

Table D.22: Referrable

Class	SwBaseType			
Note	This meta-class represents a base type used within ECU software. **Tags:** atp.recommendedPackage=BaseTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *BaseType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table D.23: SwBaseType

Class	«atpVariation» **SwDataDefProps**			
Note	This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated. **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	*ARObject*			
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, CppImplementationDataTypeElement.swDataDefProps, *DataPrototype*.swDataDefProps, DataPrototypeTransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, DiagnosticEnvDataElementCondition.swDataDefProps, DiagnosticExtendedDataRecordElement.swDataDefProps, DiagnosticSovdPrimitiveContentElement.swDataDefProps, DltArgumentProps.networkRep, FlatInstanceDescriptor.swDataDefProps, ImplementationDataTypeElement.swDataDefProps, InstantiationDataDefProps.swDataDefProps, ISignal.networkRepresentationProps, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.swDataDefProps, *ReceiverComSpec*.networkRepresentation, SecurityEventContextDataElement.networkRepresentation, *SenderComSpec*.networkRepresentation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTargetProps.swDataDefProps, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, SystemSignal.physicalProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** atp.Splitkey=annotation xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. **Tags:** atp.Splitkey=baseType xml.sequenceOffset=50
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:** atp.Splitkey=compuMethod xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:** atp.Splitkey=dataConstr xml.sequenceOffset=190



△

Class	«atpVariation» **SwDataDefProps**			
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. **Tags:** xml.sequenceOffset=210
display Presentation	DisplayPresentation Enum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** atp.Splitkey=invalidValue xml.sequenceOffset=255
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** atp.Splitkey=swComparisonVariable xml.sequenceOffset=170 xml.typeElement=false
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. **Tags:** atp.Splitkey=swHostVariable xml.sequenceOffset=220 xml.typeElement=false
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** atp.Splitkey=swTextProps xml.sequenceOffset=120
unit	Unit	0..1	ref	Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible. **Tags:** atp.Splitkey=unit xml.sequenceOffset=350

Table D.24: SwDataDefProps

Class	**SymbolProps**			
Note	If applied to Classic Platform: This meta-class represents the ability to attach with the symbol attribute a symbolic name that is conform to C language requirements to another meta-class, e.g. `AtomicSwComponentType`, that is a potential subject to a name clash on the level of RTE source code. If applied to Adaptive Platform: This meta-class represents the ability to contribute a part of a namespace.			
Base	*ARObject*, *ImplementationProps*, *Referrable*			
Aggregated by	Allocator.namespace, ApApplicationErrorDomain.namespace, *AtomicSwComponentType*.symbolProps, *CppImplementationDataType*.namespace, ImplementationDataType.symbolProps, *Port Interface*.namespace, SecurityEventDefinition.eventSymbolName, TraceSwitchConfig.namespace			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
–	–	–	–	–

Table D.25: SymbolProps

E Upstream Mapping

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This chapter describes the derived mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the upstream (M2 model) template (this document). The mappings between these models are elaborated in order to answer typical questions such as:

- How shall a supplier use the information in a `ECU_SYSTEM_DESCRIPTION` in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an "ECU Configuration Description" out of an `ECU_EXTRACT` of a `ECU_SYSTEM_DESCRIPTION`?

Parameter name	Parameter meaning	Reference
BSW Module	Name of BSW module	
BSW Context	Reference to parameter container	
BSW Parameter	Name of BSW parameter	
BSW Type	Type of BSW parameter	
BSW Description	Description from the configuration document	
Template Description	Class note or attribute note of the M2 model element	
M2 Parameter	Name of the upstream template model element	[TPS_GST_00302]
Mapping Rule	Textual description on how to transform between M2 and BSW domains	[TPS_GST_00303]
Mapping Type	• **local:** no mapping needed since parameter is local to BSW • **partial:** some data can be automatically mapped but not all • **full:** all data can be automatically mapped	[TPS_GST_00304]
ECUC Parameter ID	ID of the parameter in the respective SWS document (may be empty if the mapping is owned by an enumeration literal)	

E.1 Firewall

BSW Module	**BSW Context**
Firewall	Firewall/FirewallGeneral/FirewallSecurityEventRefs
BSW Parameter	**BSW Type**
FirewallSEvEthernetFrameMaxLength	ECUC-INTEGER-PARAM-DEF
BSW Description	
Specifies the maximum length in bytes of the Ethernet frame contained in the context data of a reported security event.	

▽

△

Template Description	
Describes the maximal length of the context data in case of Arrays and Strings.	
M2 Parameter	
SecurityEventContextDataElement.maxLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_Fw_00153]

E.2 IdsM

BSW Module	BSW Context	
IdsM		
BSW Parameter		**BSW Type**
IdsM		ECUC-MODULE-DEF
BSW Description		
Configuration of the IdsM module.		
Template Description		
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.		
M2 Parameter		
IdsmInstance		
Mapping Rule		**Mapping Type**
If "IdsmInstance" shall run on CP, then an "IdsM" EcuC module configuration needs to be created.		partial
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00001]

BSW Module	BSW Context	
IdsM	IdsM/IdsMConfiguration	
BSW Parameter		**BSW Type**
IdsMBlockState		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of an IdsM blocking state used in the IdsMStateBlockFilter to suspend the collection of security events. The active state is reported by the BswM via IdsM_BswM_StateChanged().		
Template Description		
This meta-class defines a block state that is part of the collection of block states belonging to a specific IdsmInstance. The IdsM shall discard any reported security event that is mapped to a filter chain containing a SecurityEventStateFilter that references the block state which is currently active in the IdsM.		
M2 Parameter		
BlockState		
Mapping Rule		**Mapping Type**
The (M2) BlockState is identified by its EventName (shortName or eventSymbolName) which is unique within the enclosing (M2) IdsmInstance and shall be directly mapped to an IdsMBlockState identified by its IdsMBlockStateID.		full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_IdsM_00020]

BSW Module	BSW Context	
IdsM	IdsM/IdsMConfiguration	
BSW Parameter		**BSW Type**
IdsMEvent		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Configuration of the IdsM Event unit which is reported by a sensor and its parameters.		
Template Description		
This meta-class defines a security-related event as part of the intrusion detection system.		
M2 Parameter		
SecurityEventDefinition		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00017]

BSW Module	BSW Context	
IdsM	IdsM/IdsMConfiguration/IdsMEvent	
BSW Parameter		**BSW Type**
IdsMEventSeverity		ECUC-INTEGER-PARAM-DEF
BSW Description		
The severity level of a security event. The severity of the event is used for determining the event to be dropped when buffers are full. Severity 0 is interpreted as the lowest and 255 as the highest severity.		
Template Description		
This attribute defines the ID of the security sensor that detects the referenced security event.		
M2 Parameter		
SecurityEventContextProps.sensorInstanceId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00082]

BSW Module	BSW Context	
IdsM	IdsM/IdsMConfiguration/IdsMEvent	
BSW Parameter		**BSW Type**
IdsMExternalEventId		ECUC-INTEGER-PARAM-DEF
BSW Description		

▽

△

The external security event ID which is reported to the sink.There are two different value ranges depending on the referencing module: Standardized SEv ID is defined by the AUTOSAR specification.This ID is usually derived from the SecXT. Standard ID range: 0x0000 - 0x7FFF Generic User Event ID is defined by the user. Used when the SEv is not defined by the AUTOSAR specification, for example from a SW-C or a CDD. Generic ID range: 0x8000 - 0xFFFE. 0xFFFF is considered an invalid ID
Template Description
This attribute represents the numerical identification of the defined security event. The identification shall be unique within the scope of the IDS.
M2 Parameter
SecurityEventDefinition.id

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00032]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMEvent

BSW Parameter	**BSW Type**
IdsMFilterChainRef	ECUC-REFERENCE-DEF
BSW Description	
Reference to a configured IdsM filter chain.	
Template Description	
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.	
M2 Parameter	
SecurityEventContextMapping	

Mapping Rule	**Mapping Type**
The (M2) SecurityEventDefinition (corresponding to the IdsMEvent enclosing this reference) that is referenced by (M2) SecurityEventContextProps which in turn is aggregated by (abstract M2) SecurityEventContextMapping references the (M2) SecurityEventFilterChain whose corresponding IdsMFilterChain shall be the target of this reference.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00030]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMEvent

BSW Parameter	**BSW Type**
IdsMReportingModeFilter	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The reporting mode filter defines the level of detail of the reporting. Whether SEv should be dropped, forwarded with context data or forwarded without context data. The parameter determines if the SEv is either: - dropped (OFF) - sent without context data (BRIEF) - sent with context data (DETAILED) - sent without context data, ignoring the rest of the filter chain (BRIEF_BYPASSING_FILTERS) - sent with context data ignoring the rest of the filter chain (DETAILED_BYPASSING_FILTERS)	
Template Description	
This attribute defines the default reporting mode for the referenced security event.	

▽

△

M2 Parameter	
SecurityEventContextProps.defaultReportingMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00036]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMEvent/IdsMReportingModeFilter
BSW Parameter	**BSW Type**
BRIEF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SEv is sent without context data (BRIEF)	
Template Description	
Only the main security event properties such as its ID are processed. Any additional context data (if existing) is discarded.	
M2 Parameter	
SecurityEventReportingModeEnum.brief	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMEvent/IdsMReportingModeFilter
BSW Parameter	**BSW Type**
BRIEF_BYPASSING_FILTERS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SEv is sent without context data, ignoring the rest of the filter chain (BRIEF_BYPASSING_FILTERS)	
Template Description	
The reported security event without its context data (if existing) is processed further but the filter chain is bypassed.	
M2 Parameter	
SecurityEventReportingModeEnum.briefBypassingFilters	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMEvent/IdsMReportingModeFilter
BSW Parameter	**BSW Type**
DETAILED	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SEv is sent with context data (DETAILED)	
Template Description	

▽

△

The main properties and the context data (if existing) of the reported security event are processed further.	
M2 Parameter	
SecurityEventReportingModeEnum.detailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMEvent/IdsMReportingModeFilter
BSW Parameter	**BSW Type**
DETAILED_BYPASSING_FILTERS	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SEv is sent with context data ignoring the rest of the filter chain (DETAILED_BYPASSING_FILTERS)	
Template Description	
The reported security event including its context data (if existing) is processed further but the filter chain is bypassed.	
M2 Parameter	
SecurityEventReportingModeEnum.detailedBypassingFilters	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMEvent/IdsMReportingModeFilter
BSW Parameter	**BSW Type**
OFF	ECUC-ENUMERATION-LITERAL-DEF
BSW Description	
SEv is dropped (OFF)	
Template Description	
The reported security event is not further processed by the IdsM and therefore discarded.	
M2 Parameter	
SecurityEventReportingModeEnum.off	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMEvent
BSW Parameter	**BSW Type**
IdsMSensorInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	

▽

△

The instance ID of the sensor which reports security events to the IdsM. If there is only one instance of a sensor, the default ID is 0.	
Template Description	
This attribute defines the ID of the security sensor that detects the referenced security event.	
M2 Parameter	
SecurityEventContextProps.sensorInstanceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00031]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration
BSW Parameter	**BSW Type**
IdsMFilterChain	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
A filter chain is a combination of filters that affects one or more SEvs. A filter receives a SEv, checks condition(s) and, e.g. - forwards SEv immediately/later - drops SEv - stores SEv - modifies SEv Consider that the filter order is defined as follows: - Reporting Mode Level (per SEv ID) - Block State (per SEv ID) - Forward Every nth (per SEv ID) - Event Aggregation (per SEv ID) - Event Threshold (per SEv ID) - Event Rate Limitation (per IdsM Instance) - Traffic Limitation (per IdsM Instance)	
Template Description	
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, SecurityEventOneEveryNFilter, SecurityEventAggregationFilter, SecurityEventThresholdFilter.	
M2 Parameter	
SecurityEventFilterChain	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00016]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain
BSW Parameter	**BSW Type**
IdsMBlockStateFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
This state filter drops SEvs if the current State reported by the BswM is in this state filter list.	
Template Description	
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).	
M2 Parameter	
SecurityEventStateFilter	
Mapping Rule	**Mapping Type**
1:1 mapping	full

▽

△

Mapping Status	ECUC Parameter ID
valid	[ECUC_IdsM_00021]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMBlockStateFilter
BSW Parameter	**BSW Type**
IdsMBlockStateReference	ECUC-REFERENCE-DEF
BSW Description	
The collection of SEvs during this state will be suspended.	
Template Description	
For the CP, this reference defines the 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 currently active block state in the IdsM is one of the referenced block listed states, the IdsM shall discard the reported security event.	
M2 Parameter	
SecurityEventStateFilter.blockIfStateActiveCp	

Mapping Rule	Mapping Type
The (M2) reference blockIfStateActiveCp referencing a (M2) BlockState shall be mapped to an IdsMBlockStateReference that references the IdsMBlockState which corresponds to the (M2) BlockState,.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00051]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain
BSW Parameter	**BSW Type**
IdsMEventAggregationFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
All received events of a certain event ID that are received by this filter during a single aggregation time interval are not forwarded immediately. Instead, only the last or the first received SEv is stored in an aggregation buffer, depending on the configuration of "IdsMContextDataSourceSelector". The counter field of the SEv is modified so that it contains the sum of the counter fields of all incoming SEvs during the current aggregation time interval. At the end of the aggregation time interval, the buffered SEv is sent out and the aggregation buffer is cleared. If there was no incoming SEv until the end of the aggregation time interval, no message will be sent.	
Template Description	
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.	
M2 Parameter	
SecurityEventAggregationFilter	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00024]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMEventAggregationFilter
BSW Parameter	**BSW Type**
IdsMContextDataSourceSelector	ECUC-ENUMERATION-PARAM-DEF
BSW Description	
The resulting SEv from the aggregation filter contains the context data from one of the following two sources: IDSM_FILTERS_CTX_USE_FIRST = ContextData of first received SEv is used for resulting QSEv. IDSM_FILTERS_CTX_USE_LAST = ContextData of last received SEv is used for resulting QSEv.	
Template Description	
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.	
M2 Parameter	
SecurityEventAggregationFilter.contextDataSource	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00026]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMEventAggregationFilter
BSW Parameter	**BSW Type**
IdsMEventAggregationTimeInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Length of the aggregation time interval (as float in seconds). Note: Shall be configured as a multiple of the IdsM main function period.	
Template Description	
This attribute represents the configuration of the minimum time window in seconds for the aggregation filter.	
M2 Parameter	
SecurityEventAggregationFilter.minimumIntervalLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00025]

BSW Module	BSW Context
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain
BSW Parameter	**BSW Type**
IdsMEventThresholdFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
During each time interval "IdsMEventThresholdTimeInterval", the filter drops the first "IdsMEventThresholdNumber - 1" SEvs and forwards all other incoming SEvs immediately until the end of the time interval.	
Template Description	
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.	
M2 Parameter	



△

SecurityEventThresholdFilter	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00027]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMEventThresholdFilter
BSW Parameter	**BSW Type**
IdsMEventThresholdNumber	ECUC-INTEGER-PARAM-DEF
BSW Description	
This parameter assigns the threshold 'p' for each SEv ID affected by this threshold filter. All SEvs ' p-1' are dropped, SEvs equal or greater than 'p' are forwarded.	
Template Description	
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.	
M2 Parameter	
SecurityEventThresholdFilter.thresholdNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00029]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMEventThresholdFilter
BSW Parameter	**BSW Type**
IdsMEventThresholdTimeInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Length of the threshold time interval (as float in seconds). Note: Shall be configured as a multiple of the IdsM main function period.	
Template Description	
This attribute configures the time interval in seconds for one threshold filter operation.	
M2 Parameter	
SecurityEventThresholdFilter.intervalLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00028]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain
BSW Parameter	**BSW Type**
IdsMForwardEveryNthFilter	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	

▽

△

Out of all incoming SEVs, drop all but every nth. Those will be forwarded without modification.	
Template Description	
This meta-class represents the configuration of a sampling (i.e. every n-th event is sampled) filter for security events.	
M2 Parameter	
SecurityEventOneEveryNFilter	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00022]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMConfiguration/IdsMFilterChain/IdsMForwardEveryNthFilter
BSW Parameter	**BSW Type**
IdsMNthParameter	ECUC-INTEGER-PARAM-DEF
BSW Description	
For each SEv ID for which this filter is configured, this parameter assigns the appropriate n. Only 1 from n SEvs will be forwarded.	
Template Description	
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.	
M2 Parameter	
SecurityEventOneEveryNFilter.n	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00023]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMGeneral
BSW Parameter	**BSW Type**
IdsMAuthenticator	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
If this container exists all qualified security events are authenticated by the crypto service.	
Template Description	
This meta-class defines, for the Classic Platform, the cryptographic algorithm and key to be used by the IdsM instance for providing signature information in QSEv messages.	
M2 Parameter	
IdsmSignatureSupportCp	
Mapping Rule	**Mapping Type**
If the aggregation in the role (M2) signatureSupportCp exists, then the (M1) subcontainer Ids MSignature shall be instantiated to hold the respective signature configuration.	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00059]

BSW Module	BSW Context	
IdsM	IdsM/IdsMGeneral/IdsMAuthenticator	
BSW Parameter		**BSW Type**
IdsMAuthenticatorLength		ECUC-INTEGER-PARAM-DEF
BSW Description		
This parameter defines the length of the authenticator in bytes calculated by the crypto service.		
Template Description		
CryptoServicePrimitive: This meta-class has the ability to represent a crypto primitive. **CryptoServiceKey:** This meta-class has the ability to represent a crypto key.		
M2 Parameter		
`CryptoServicePrimitive`, `CryptoServiceKey`		
Mapping Rule		**Mapping Type**
The (M1) IdsMSignatureLength needs to be deduced from the configured (M2) CryptoService Primitive and the configured (M2) CryptoServiceKey (if existing).		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00011]

BSW Module	BSW Context	
IdsM	IdsM/IdsMGeneral	
BSW Parameter		**BSW Type**
IdsMAuthenticatorSupport		ECUC-BOOLEAN-PARAM-DEF
BSW Description		
This parameter enables/disables the functionality of sending messages to the network with an authenticator calculated by the crypto services.		
Template Description		
The existence of this aggregation specifies that the IdsM shall add a signature to the QSEv messages it sends onto the network. The cryptographic algorithm and key to be used for this signature is further specified by the aggregated meta-class specifically for the Classic Platform.		
M2 Parameter		
`IdsmInstance.signatureSupportCp`		
Mapping Rule		**Mapping Type**
If the aggregation in the role (M2) signatureSupportCp exists, then IdsMSignatureSupport = TRUE. Otherwise, IdsMSignatureSupport = FALSE.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00009]

BSW Module	BSW Context	
IdsM	IdsM/IdsMGeneral	
BSW Parameter		**BSW Type**
IdsMGlobalRateLimitationFilters		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
Global rate limitation filters for all SEvs.		
Template Description		
This meta-class provides the ability to aggregate filters for security events.		
M2 Parameter		

▽

△

IdsmProperties	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00008]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters
BSW Parameter	**BSW Type**
IdsMFilterEventRateLimitation	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
For configurable time intervals of length "IdsMRateLimitationTimeInterval" this filter forwards all the SEvs until reaching the limit "IdsMRateLimitationMaximumEvents". The limit is measured in number of incoming SEvs. Until the end of the time interval, all subsequent SEvs are dropped. This is helpful to cap the load that the IdsM generates unto information sinks like the IdsR. This filter is not specific to a single SEv but it applies to all SEvs handled by the current IdsM instance. Note: Each possible SEv counts as a single one, regardless of its counter value.	
Template Description	
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.	
M2 Parameter	
IdsmRateLimitation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00053]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters/IdsMFilterEventRateLimitation
BSW Parameter	**BSW Type**
IdsMRateLimitationMaximumEvents	ECUC-INTEGER-PARAM-DEF
BSW Description	
The maximum number of SEvs which are passed on by this filter in a single rate limitation time interval.	
Template Description	
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.	
M2 Parameter	
IdsmRateLimitation.maxEventsInInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00055]

BSW Module	BSW Context
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters/IdsMFilterEventRateLimitation
BSW Parameter	**BSW Type**
IdsMRateLimitationTimeInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Time interval length of the event rate limitation filter (as float in seconds). Note: Shall be configured as a multiple of the IdsM main function period.	
Template Description	
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.	
M2 Parameter	
IdsmRateLimitation.timeInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00054]

BSW Module	BSW Context
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters
BSW Parameter	**BSW Type**
IdsMFilterTrafficLimitation	ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description	
The traffic limitation filter forwards all the incoming SEvs until reaching the limit "IdsMTrafficLimitationMaximumBytes". The limit is measured in incoming amount of bytes. This filter forwards SEvs only, if the accumulated sizes of all incoming SEvs in the current traffic limitation time interval up until the current SEv is smaller or equal than a configurable maximum number of bytes "IdsMTrafficLimitationMaximumBytes". The length of the traffic limitation time interval is configurable in "IdsMTrafficLimitationTimeInterval". This filter is not specific to a single SEv but it applies to all SEvs handled by the current IdsM instance.	
Template Description	
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.	
M2 Parameter	
IdsmTrafficLimitation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00056]

BSW Module	BSW Context
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters/IdsMFilterTrafficLimitation
BSW Parameter	**BSW Type**
IdsMTrafficLimitationMaximumBytes	ECUC-INTEGER-PARAM-DEF
BSW Description	
The maximum number of bytes to be sent out by the IdsM in a single traffic limitation time interval.	
Template Description	

▽

△

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.	
M2 Parameter	
IdsmTrafficLimitation.maxBytesInInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00058]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters/IdsMFilterTrafficLimitation
BSW Parameter	**BSW Type**
IdsMTrafficLimitationTimeInterval	ECUC-FLOAT-PARAM-DEF
BSW Description	
Length of the traffic limitation time interval (as float in seconds). Note: Shall be configured as a multiple of the IdsM main function period.	
Template Description	
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.	
M2 Parameter	
IdsmTrafficLimitation.timeInterval	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00057]

BSW Module	**BSW Context**
IdsM	IdsM/IdsMGeneral
BSW Parameter	**BSW Type**
IdsMInstanceId	ECUC-INTEGER-PARAM-DEF
BSW Description	
The unique identifier of the sending IdsM instance. This ID helps identifying the origin of a SEv, together with the SEv configuration parameters: ExternalEventId and the IdsMSensorInstanceId. Note: There is only one IdsM (from the AUTOSAR Classic Platform) instance per ECU.	
Template Description	
This attribute is used to provide a source identification in the context of reporting security events..	
M2 Parameter	
IdsmInstance.idsmInstanceId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**ECUC Parameter ID**
valid	[ECUC_IdsM_00007]

BSW Module	BSW Context	
IdsM	IdsM/IdsMGeneral	
BSW Parameter		**BSW Type**
IdsMTimestamp		ECUC-PARAM-CONF-CONTAINER-DEF
BSW Description		
If this container exists a timestamp field is added to all qualified security events.		
Template Description		
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.		
M2 Parameter		
IdsmInstance.timestampFormat		
Mapping Rule		**Mapping Type**
If the (M2) attribute timestampFormat exists, then the (M1) subcontainer IdsMTimestamp shall be instantiated to hold the respective timestamp configuration.		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00060]

BSW Module	BSW Context	
IdsM	IdsM/IdsMGeneral/IdsMTimestamp	
BSW Parameter		**BSW Type**
IdsMTimestampOption		ECUC-ENUMERATION-PARAM-DEF
BSW Description		
This parameter speciies if the origin of the timestamp is from the AUTOSAR stack or from the application (custom timestamp).		
Template Description		
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.		
M2 Parameter		
IdsmInstance.timestampFormat		
Mapping Rule		**Mapping Type**
If (M2) timestampFormat is "AUTOSAR", then IdsMTimeStampOption = "AUTOSAR". Otherwise, IdsMTimeStampOption = "Custom"		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_IdsM_00012]

E.3 KeyM

BSW Module	BSW Context	
KeyM	KeyM/KeyMGeneral/KeyMSecurityEventRefs	
BSW Parameter	**BSW Type**	
KeyMSEvCertificateMaxLength	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Specify the maximum length in bytes of the certificate contained in the context data of a reported security event.		
Template Description		
Describes the maximal length of the context data in case of Arrays and Strings.		
M2 Parameter		
SecurityEventContextDataElement.maxLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_KeyM_00090]

E.4 TcpIp

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpGeneral/TcpIpSecurityEventRefs	
BSW Parameter	**BSW Type**	
TcpIpSEvReceivedPacketMaxLength	ECUC-INTEGER-PARAM-DEF	
BSW Description		
Specifies the maximum length in bytes of the Received Packet contained in the context data of a reported security event.		
Template Description		
Describes the maximal length of the context data in case of Arrays and Strings.		
M2 Parameter		
SecurityEventContextDataElement.maxLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**ECUC Parameter ID**
valid		[ECUC_TcpIp_00347]

F Non-Splitable Elements in the Scope of this Document

This chapter contains a table of all non-splitable model elements in the scope of this document.

For more information about the concept of splitable model elements and how these shall be treated please refer to the [12, TPS Generic Structure Template].

Name of non splitable element
SecurityEventContextDataDefinition.contextDataElement
SecurityEventContextDataElement.nestedContextData

Table F.1: Usage of non splitable elements

G Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped ≪atpVariation≫ in the scope of this document.

Each entry in the table consists of the identification of the specific model element itself and the applicable value of the tagged value vh.latestBindingTime.

For more information about the concept of variation points and how model elements that contain variation points shall be processed in a tool, please refer to [12]

Variation Point	*Latest Binding Time*
IdsDesign.element	systemDesignTime
IdsmInstance.ecuInstance	systemDesignTime
IdsmInstance.rateLimitationFilter	preCompileTime
IdsmInstance.trafficLimitationFilter	preCompileTime
SecurityEventContextDataDefinition.contextDataElement	systemDesignTime
SecurityEventContextMapping.filterChain	preCompileTime
SecurityEventContextMapping.idsmInstance	systemDesignTime
SecurityEventContextMapping.mappedSecurityEvent	preCompileTime
SecurityEventContextMappingCommConnector.commConnector	preCompileTime
SecurityEventContextProps.securityEvent	systemDesignTime
SecurityEventDefinition.securityEventContextDataDefinition	systemDesignTime
SecurityEventReportInstanceValue.contextDataElementIdentification	systemDesignTime

Table G.1: Usage of variation points

H Change History

The content of this appendix chapter is *informative* in nature and shall **not** be considered as *normative* content.

This chapter provides the change history of traceable items in this document. The lists also include traceable items that have been removed from the document in a later version. These items do not appear as hyperlinks in the document.

H.1 Change History of this document according to AUTOSAR Release R20-11

H.1.1 Added Specification Items in R20-11

Number	Heading
[TPS_SECXT_01043]	Semantics of `IdsDesign`
[TPS_SECXT_01000]	Semantics of `SecurityEventSet`
[TPS_SECXT_01001]	Semantics of `SecurityEventDefinition`
[TPS_SECXT_01002]	EventName of `SecurityEventDefinition`
[TPS_SECXT_01003]	Semantics of attribute `SecurityEventDefinition.id`
[TPS_SECXT_01004]	Textual description of `SecurityEventDefinition`
[TPS_SECXT_01005]	Semantics of SecurityEventContextData
[TPS_SECXT_01006]	Filtering Semantics of `SecurityEventFilterChain`
[TPS_SECXT_01007]	Applicability of `SecurityEventFilterChain` towards `SecurityEventDefinition`s
[TPS_SECXT_01008]	Semantics of `SecurityEventStateFilter`
[TPS_SECXT_01009]	Semantics of `SecurityEventOneEveryNFilter`
[TPS_SECXT_01010]	Semantics of `SecurityEventAggregationFilter`
[TPS_SECXT_01011]	Semantics of attribute `SecurityEventAggregationFilter.contextDataSource`
[TPS_SECXT_01012]	Semantics of `SecurityEventThresholdFilter`
[TPS_SECXT_01013]	Final Qualification of a `SecurityEventDefinition`
[TPS_SECXT_01014]	Semantics of `IdsmRateLimitation`
[TPS_SECXT_01015]	Semantics of `IdsmTrafficLimitation`
[TPS_SECXT_01016]	Semantics of `SecurityEventMapping`
[TPS_SECXT_01017]	Semantics of attribute `SecurityEventMapping.defaultReportingModeSecurityEventMapping`
[TPS_SECXT_01018]	Semantics of `SecurityEventMappingContextBswModule`
[TPS_SECXT_01019]	Mapping of Security Events to Filter Chain by `SecurityEventMappingContextBswModule`
[TPS_SECXT_01020]	Semantics of `SecurityEventMappingContextFunctionalCluster`

▽

△

Number	Heading
[TPS_SECXT_01021]	Mapping of Security Events to Filter Chain by SecurityEventMapping-ContextFunctionalCluster
[TPS_SECXT_01022]	Semantics of SecurityEventMappingContextCommConnector
[TPS_SECXT_01023]	Mapping of Security Events to Filter Chain by SecurityEventMapping-ContextCommConnector
[TPS_SECXT_01024]	Semantics of SecurityEventMappingContextApplication
[TPS_SECXT_01025]	Mapping of Security Events to Filter Chain by SecurityEventMapping-ContextApplication
[TPS_SECXT_01026]	Semantics of IdsmInstance on CP
[TPS_SECXT_01027]	Semantics of IdsmInstance on AP
[TPS_SECXT_01028]	Semantics of attribute IdsmInstance.idsmInstanceId
[TPS_SECXT_01029]	Semantics of attribute IdsmInstance.timestampSupport
[TPS_SECXT_01030]	Semantics of attribute IdsmInstance.timestampFormat
[TPS_SECXT_01031]	Semantics of attribute IdsmInstance.signatureSupport
[TPS_SECXT_01032]	Semantics of IdsmSignatureSupportCp
[TPS_SECXT_01033]	Semantics of IdsmSignatureSupportAp
[TPS_SECXT_01034]	Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventMappingContextBswModule on CP
[TPS_SECXT_01035]	Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventMappingContextFunctionalCluster on AP
[TPS_SECXT_01036]	Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventMappingContextCommConnector
[TPS_SECXT_01037]	Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventMappingContextApplication
[TPS_SECXT_01038]	Network configuration of an IdsmInstance on CP
[TPS_SECXT_01039]	Network configuration of an IdsmInstance on AP
[TPS_SECXT_01040]	Semantics of SecurityEventMappingProps
[TPS_SECXT_01041]	Semantics of attribute SecurityEventMapping.persistentStorage
[TPS_SECXT_01042]	Semantics of attribute SecurityEventMappingProps.severity

Table H.1: Added Specification Items in R20-11

H.1.2 Changed Specification Items in R20-11

none

H.1.3 Deleted Specification Items in R20-11

none

H.1.4 Added Constraints in R20-11

Number	Heading
[constr_5600]	Valid interval for attribute `SecurityEventDefinition.id`
[constr_5601]	Uniqueness of `SecurityEventDefinition.id`
[constr_5602]	Valid interval for attribute `SecurityEventOneEveryNFilter.n`
[constr_5603]	Valid interval for attribute `SecurityEventAggregationFilter.minimumIntervalLength`
[constr_5604]	Valid interval for attribute `SecurityEventThresholdFilter.intervalLength`
[constr_5605]	Valid interval for attribute `SecurityEventThresholdFilter.thresholdNumber`
[constr_5606]	Valid interval for attribute `IdsmRateLimitation.timeInterval`
[constr_5607]	Valid interval for attribute `IdsmRateLimitation.maxEventsInInterval`
[constr_5608]	Valid interval for attribute `IdsmTrafficLimitation.timeInterval`
[constr_5609]	Valid interval for attribute `IdsmTrafficLimitation.maxBytesInInterval`
[constr_5610]	Unambiguous definition of execution platform for an `IdsmInstance`
[constr_5611]	Unambiguous configuration of platform-dependent signature support for an `IdsmInstance`
[constr_5612]	Unambiguous definition of platform-dependent network configuration for an `IdsmInstance`

Table H.2: Added Constraints in R20-11

H.1.5 Changed Constraints in R20-11

none

H.1.6 Deleted Constraints in R20-11

none

H.2 Change History of this document according to AUTOSAR Release R22-11

H.2.1 Added Specification Items in R22-11

Number	Heading
[TPS_SECXT_02000]	Definition of pre-qualified security events

Table H.3: Added Specification Items in R22-11

H.2.2 Changed Specification Items in R22-11

none

H.2.3 Deleted Specification Items in R22-11

none

H.2.4 Added Constraints in R22-11

none

H.2.5 Changed Constraints in R22-11

none

H.2.6 Deleted Constraints in R22-11

none

H.3 Change History of this document according to AUTOSAR Release R23-11

H.3.1 Added Specification Items in R23-11

none

H.3.2 Changed Specification Items in R23-11

none

H.3.3 Deleted Specification Items in R23-11

none

H.3.4 Added Constraints in R23-11

none

H.3.5 Changed Constraints in R23-11

none

H.3.6 Deleted Constraints in R23-11

none

H.3.7 Added Advisories in R23-11

none

H.3.8 Changed Advisories in R23-11

none

H.3.9 Deleted Advisories in R23-11

none

H.4 Change History of this document according to AUTOSAR Release R24-11

H.4.1 Added Specification Items in R24-11

Number	Heading
[TPS_SECXT_03000]	Identification of security event context data
[TPS_SECXT_03001]	Semantics of meta-class SecurityEventReportInstanceValue
[TPS_SECXT_03002]	Semantics of the reference in the role SecurityEventReportInstanceValue. contextDataElementIdentification
[TPS_SECXT_03003]	Specifying the sequence of context elements in attribute SecurityEventReportInstanceDefinition.contextTypeList
[TPS_SECXT_03004]	Semantics of meta-class SecurityEventReportInstanceDefinition
[TPS_SECXT_04000]	SecurityEventContextDataDefinition with aggregated SecurityEventContextDataElements
[TPS_SECXT_04001]	SecurityEventContextDataElement.networkRepresentation
[TPS_SECXT_04002]	Description of a primitive as networkRepresentation of a SecurityEventContextDataElement

▽

△

Number	Heading
[TPS_SECXT_04003]	SecurityEventContextDataDefinition that is structured into SecurityEventContextDataElements
[TPS_SECXT_04004]	SecurityEventContextDataElement that is structured into SecurityEventContextDataElements
[TPS_SECXT_04005]	Description of a String as networkRepresentation of a SecurityEventContextDataElement
[TPS_SECXT_04006]	Description of one dimensional array as networkRepresentation of a SecurityEventContextDataElement
[TPS_SECXT_04007]	Description of a n-dimensional array as networkRepresentation of a SecurityEventContextDataElement

Table H.4: Added Specification Items in R24-11

H.4.2 Changed Specification Items in R24-11

Number	Heading
[TPS_SECXT_01005]	Semantics of SecurityEventContextDataDefinition

Table H.5: Changed Specification Items in R24-11

H.4.3 Deleted Specification Items in R24-11

none

H.4.4 Added Constraints in R24-11

Number	Heading
[constr_12000]	Usage of references in the context of SecurityEventReportInstanceValue
[constr_12001]	Existence of attribute SecurityEventReportInstanceValue.id
[constr_12002]	Existence of attribute SecurityEventReportInstanceDefinition.targetType
[constr_12003]	Existence of reference SecurityEventReportInstanceValue.contextDataElementIdentification
[constr_9339]	SecurityEventContextDataElement.maxLength usage restriction
[constr_9340]	Datatype of an Array
[constr_9341]	CompuMethod in SecurityEventContextDataElement.networkRepresentation
[constr_9342]	Allowed range of SecurityEventContextDataDefinition.version

Table H.6: Added Constraints in R24-11

H.4.5 Changed Constraints in R24-11

none

H.4.6 Deleted Constraints in R24-11

none

H.4.7 Added Advisories in R24-11

none

H.4.8 Changed Advisories in R24-11

none

H.4.9 Deleted Advisories in R24-11

none

H.5 Change History of this document according to AUTOSAR Release R25-11

H.5.1 Added Specification Items in R25-11

Number	Heading
[TPS_SECXT_02001]	Content of `SecurityEventContextMappingBswModule.affectedBswModule`
[TPS_SECXT_02002]	Content of `SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster`

Table H.7: Added Specification Items in R25-11

H.5.2 Changed Specification Items in R25-11

none

H.5.3 Deleted Specification Items in R25-11

none

H.5.4 Added Constraints in R25-11

Number	Heading
[constr_12004]	Valid interval for attribute SecurityEventContextProps.sensorInstanceId
[constr_12005]	Valid interval for attribute IdsmInstance.idsmInstanceId
[constr_12006]	Existence of attribute IdsmInstance.idsmInstanceId
[constr_12007]	Existence of attribute IdsmInstance.rateLimitationFilter
[constr_12008]	Existence of attribute IdsmInstance.trafficLimitationFilter
[constr_12009]	Existence of attribute IdsmTrafficLimitation.maxBytesInInterval
[constr_12010]	Existence of attribute IdsmTrafficLimitation.timeInterval
[constr_12011]	Existence of attribute SecurityEventDefinition.id
[constr_12012]	Existence of attribute SecurityEventAggregationFilter.contextDataSource
[constr_12013]	Existence of attribute SecurityEventAggregationFilter.minimumIntervalLength
[constr_12014]	Existence of attribute SecurityEventOneEveryNFilter.n
[constr_12015]	Existence of attribute SecurityEventThresholdFilter.intervalLength
[constr_12016]	Existence of attribute SecurityEventThresholdFilter.thresholdNumber
[constr_12017]	Existence of attribute SecurityEventContextMapping.idsmInstance
[constr_12018]	Existence of attribute SecurityEventContextMapping.mappedSecurityEvent
[constr_12019]	Existence of attribute SecurityEventContextProps.defaultReportingMode
[constr_12020]	Existence of attribute SecurityEventContextProps.securityEvent
[constr_12021]	Existence of attribute SecurityEventContextProps.sensorInstanceId

Table H.8: Added Constraints in R25-11

H.5.5 Changed Constraints in R25-11

Number	Heading
[constr_5600]	Valid interval for attribute SecurityEventDefinition.id
[constr_5602]	Valid interval for attribute SecurityEventOneEveryNFilter.n
[constr_5605]	Valid interval for attribute SecurityEventThresholdFilter.thresholdNumber
[constr_5607]	Valid interval for attribute IdsmRateLimitation.maxEventsInInterval
[constr_5609]	Valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval

Table H.9: Changed Constraints in R25-11

H.5.6 Deleted Constraints in R25-11

none

H.6 Change History of this document according to AUTOSAR Release R26-11

H.6.1 Added Specification Items in R26-11

none

H.6.2 Changed Specification Items in R26-11

Number	Heading
[TPS_SECXT_01027]	Semantics of `IdsmInstance` on AP
[TPS_SECXT_01031]	Definition of signature support for an `IdsmInstance` on the AUTOSAR classic platform
[TPS_SECXT_01039]	Network configuration of an `IdsmInstance` on AP
[TPS_SECXT_01046]	Semantics of `SecurityEventStateFilter` for AP

Table H.10: Changed Specification Items in R26-11

H.6.3 Deleted Specification Items in R26-11

Number	Heading
[TPS_SECXT_01033]	Semantics of `IdsmSignatureSupportAp`

Table H.11: Deleted Specification Items in R26-11

H.6.4 Added Constraints in R26-11

Number	Heading
[constr_12022]	Existence of role `IdsmRateLimitation.maxEventsInInterval`
[constr_12023]	Existence of role `IdsmRateLimitation.timeInterval`
[constr_12024]	Existence of role `SecurityEventContextMappingFunctionalCluster.affectedFunctionalCluster`
[constr_12025]	Existence of role `SecurityEventContextMappingApplication.affectedApplication`

Table H.12: Added Constraints in R26-11

H.6.5 Changed Constraints in R26-11

Number	Heading
[constr_5610]	Unambiguous definition of execution platform for an `IdsmInstance`
[constr_5612]	Unambiguous definition of platform-dependent network configuration for an `IdsmInstance`

Table H.13: Changed Constraints in R26-11

H.6.6 Deleted Constraints in R26-11

Number	Heading
[constr_5611]	Unambiguous configuration of platform-dependent signature support for an `IdsmInstance`

Table H.14: Deleted Constraints in R26-11

H.6.7 Added Advisories in R26-11

none

H.6.8 Changed Advisories in R26-11

none

H.6.9 Deleted Advisories in R26-11

none