| | |
|---|---|
| **Document Title** | Security Extract Template |
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 980 |

| | |
|---|---|
| **Document Status** | published |
| **Part of AUTOSAR Standard** | Foundation |
| **Part of Standard Release** | R22-11 |

**Document Change History**

| Date | Release | Changed by | Description |
|---|---|---|---|
| 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.

# Contents

# References

[1] Specification of Intrusion Detection System Manager
AUTOSAR_SWS_IntrusionDetectionSystemManager

[2] Diagnostic Extract Template
AUTOSAR_TPS_DiagnosticExtractTemplate

[3] System Template
AUTOSAR_TPS_SystemTemplate

[4] Specification of Manifest
AUTOSAR_TPS_ManifestSpecification

[5] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

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

[7] Specification of Cryptography
AUTOSAR_SWS_Cryptography

[8] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager

[9] Generic Structure Template
AUTOSAR_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 "binding 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 binding times of constraints are defined and the binding 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 Standardization Template, chapter Support for Traceability ([5]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([5]).

## 1.3 Requirements Tracing

Requirements against this document are exclusively stated in the corresponding requirements document.

The following table 1.1 references the requirements specified in the corresponding requirements document and provides information about individual specification items that fulfill a given requirement.

| Requirement | Description | Satisfied by |
|---|---|---|
| **[RS_SECXT_00001]** | Definition of Security Events | [TPS_SECXT_01000] [TPS_SECXT_01001] [TPS_SECXT_01002] [TPS_SECXT_01003] [TPS_SECXT_01004] [TPS_SECXT_01040] |

▽

△

| Requirement | Description | Satisfied by |
|---|---|---|
| **[RS_SECXT_00002]** | Filter Chains for Security Events | [TPS_SECXT_01006] [TPS_SECXT_01007] [TPS_SECXT_01008] [TPS_SECXT_01009] [TPS_SECXT_01010] [TPS_SECXT_01011] [TPS_SECXT_01012] [TPS_SECXT_01013] [TPS_SECXT_01019] [TPS_SECXT_01021] [TPS_SECXT_01023] [TPS_SECXT_01025] [TPS_SECXT_01044] [TPS_SECXT_01045] [TPS_SECXT_01046] [TPS_SECXT_01048] |
| **[RS_SECXT_00003]** | Limitation Filtering for Security Events | [TPS_SECXT_01014] [TPS_SECXT_01015] |
| **[RS_SECXT_00004]** | Association of Security Event with an ECU/Machine | [TPS_SECXT_01016] [TPS_SECXT_01034] [TPS_SECXT_01035] [TPS_SECXT_01036] [TPS_SECXT_01037] [TPS_SECXT_01040] |
| **[RS_SECXT_00005]** | Association of Security Event with a Communication Bus | [TPS_SECXT_01022] [TPS_SECXT_01023] [TPS_SECXT_01036] |
| **[RS_SECXT_00006]** | Support the Persistent Storage of Security Events | [TPS_SECXT_01041] |
| **[RS_SECXT_00007]** | Definition of Default Reporting Modes for Security Events | [TPS_SECXT_01013] [TPS_SECXT_01017] |
| **[RS_SECXT_00008]** | Association of Security Event with a Platform Module | [TPS_SECXT_01018] [TPS_SECXT_01019] [TPS_SECXT_01020] [TPS_SECXT_01021] [TPS_SECXT_01034] [TPS_SECXT_01035] |
| **[RS_SECXT_00009]** | Support optional Context Data for Security Events | [TPS_SECXT_01005] |
| **[RS_SECXT_00010]** | Derivation of related ECU-C parameters | [TPS_SECXT_01026] |
| **[RS_SECXT_00011]** | Specification of AUTOSAR Standardized Security Events | [TPS_SECXT_01043] |
| **[RS_SECXT_00012]** | Pre-Qualification Provision for Security Events | [TPS_SECXT_02000] |
| **[RS_SECXT_00013]** | Optional Configuration of IdsM Instances | [TPS_SECXT_01026] [TPS_SECXT_01027] [TPS_SECXT_01028] |
| **[RS_SECXT_00014]** | Optional Configuration of Timestamp Provisioning | [TPS_SECXT_01029] |
| **[RS_SECXT_00015]** | Configuration of Timestamp Format | [TPS_SECXT_01030] |
| **[RS_SECXT_00016]** | Optional Configuration of Authentication Provisioning for Security Event Messages | [TPS_SECXT_01031] [TPS_SECXT_01032] [TPS_SECXT_01033] |
| **[RS_SECXT_00017]** | Association of Network Configuration to an IdsM Instance | [TPS_SECXT_01038] [TPS_SECXT_01039] |
| **[RS_SECXT_00018]** | Support definition of Severity Levels at Mapping of Security Events | [TPS_SECXT_01042] |
| **[RS_SECXT_00019]** | Support definition of IDS scope and system boundaries | [TPS_SECXT_01043] |
| **[RS_SECXT_00020]** | Support partial and complete exchange of Security Extract definitions | [TPS_SECXT_01043] |
| **[RS_SECXT_00021]** | Association of Security Event with an Application | [TPS_SECXT_01024] [TPS_SECXT_01025] [TPS_SECXT_01037] |
| **[RS_SECXT_00023]** | Definition of Security Sensor ID for a Security Event | [TPS_SECXT_01047] |

**Table 1.1: RequirementsTracing**

# 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.

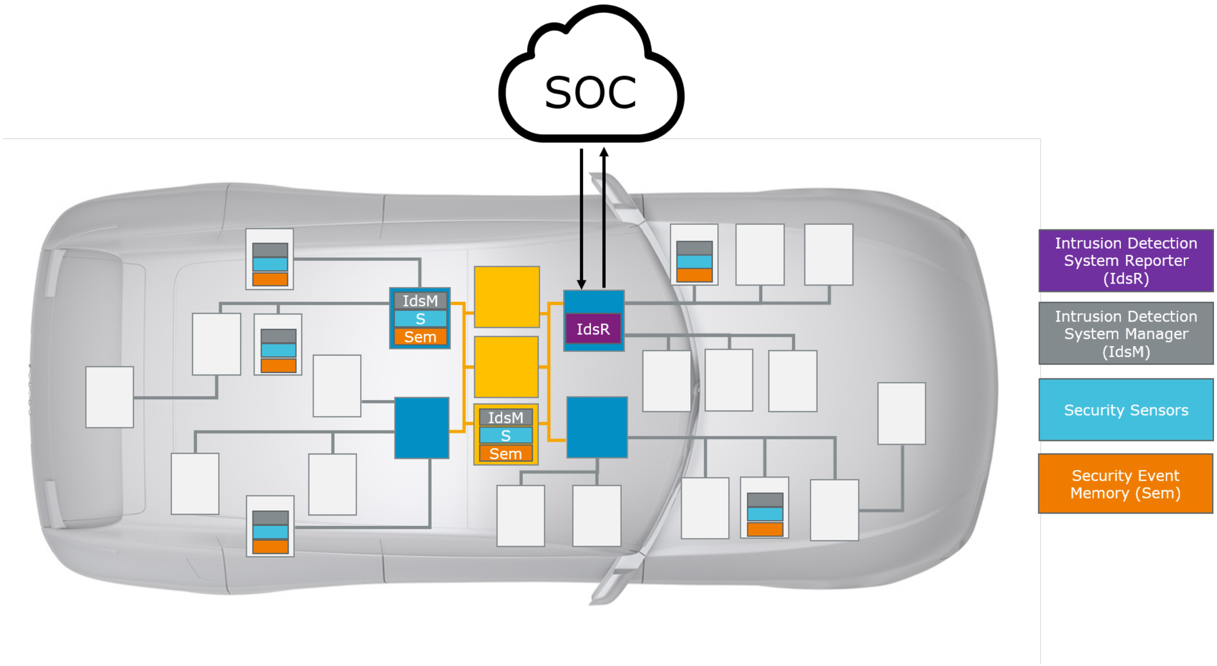
SOC
IdsM
S
Sem
IdsR
IdsM
S
Sem
Intrusion Detection System Reporter (IdsR)
Intrusion Detection System Manager (IdsM)
Security Sensors
Security Event Memory (Sem)

**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. B).

### 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.

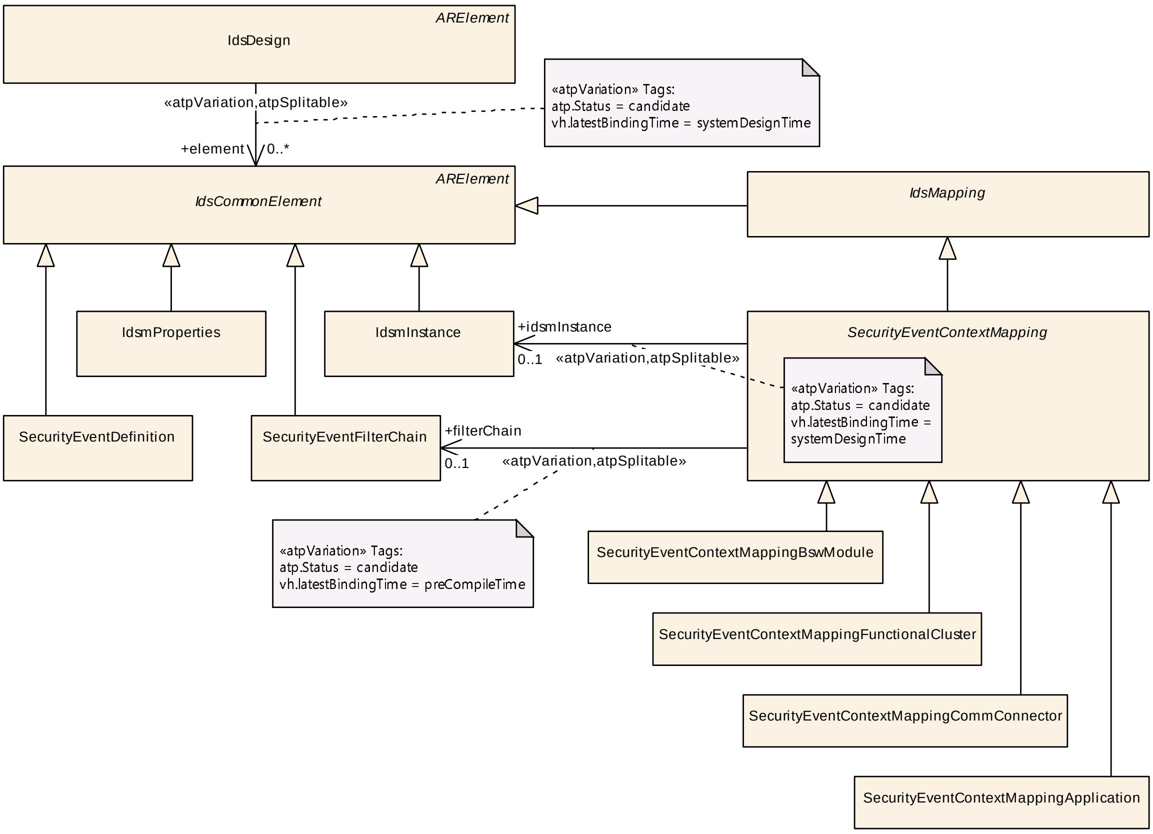
ARElement
IdsDesign
«atpVariation,atpSplitable»
+element
0..*
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
ARElement
IdsCommonElement
IdsMapping
IdsmProperties
IdsmInstance
+idsmInstance
0..1
«atpVariation,atpSplitable»
SecurityEventContextMapping
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime =
systemDesignTime
SecurityEventDefinition
SecurityEventFilterChain
+filterChain
0..1
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
SecurityEventContextMappingBswModule
SecurityEventContextMappingFunctionalCluster
SecurityEventContextMappingCommConnector
SecurityEventContextMappingApplication

**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]**{DRAFT} **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.⌋*(RS_SECXT_00019, RS_SECXT_00020, RS_SECXT_00011)*

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

| ***Class*** | **IdsDesign** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=IdsDesigns | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| element | IdsCommonElement | * | ref | This reference includes an element with IDS related definitions into the IdsDesign.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=element.idsCommonElement, element.variationPoint.shortLabel<br>atp.Status=candidate<br>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. SecurityEventDefinitions 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]**{DRAFT} **Semantics of SecurityEventDefinition** ⌈A SecurityEventDefinition represents the atomic unit of a security-related event with pre-defined properties that is reported by security sensors and further processed by the IdsM.⌋*(RS_SECXT_00001)*

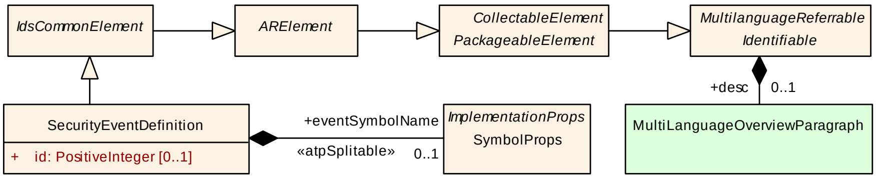

**Figure 4.2: Modeling of SecurityEventDefinition**

<table>
<tr><td><b><i>Class</i></b></td><td colspan="4"><b>SecurityEventDefinition</b></td></tr>
<tr><td><b><i>Package</i></b></td><td colspan="4">M2::AUTOSARTemplates::SecurityExtractTemplate</td></tr>
<tr><td><b><i>Note</i></b></td><td colspan="4">This meta-class defines a security-related event as part of the intrusion detection system.<br><b>Tags:</b><br>atp.Status=candidate<br>atp.recommendedPackage=SecurityEventDefinitions</td></tr>
<tr><td><b><i>Base</i></b></td><td colspan="4"><i>ARElement</i>, <i>ARObject</i>, <i>CollectableElement</i>, <i>Identifiable</i>, <i>IdsCommonElement</i>, <i>MultilanguageReferrable</i>, <i>PackageableElement</i>, <i>Referrable</i></td></tr>
<tr><td><b><i>Aggregated by</i></b></td><td colspan="4">ARPackage.element</td></tr>
<tr><td><b><i>Attribute</i></b></td><td><b><i>Type</i></b></td><td><b><i>Mult.</i></b></td><td><b><i>Kind</i></b></td><td><b><i>Note</i></b></td></tr>
<tr><td>eventSymbolName</td><td>SymbolProps</td><td>0..1</td><td>aggr</td><td>This aggregation defines optionally an alternative Event Name for the SecurityEventDefinition in case there is a collision of shortNames.<br><b>Stereotypes:</b> atpSplitable<br><b>Tags:</b><br>atp.Splitkey=eventSymbolName.shortName<br>atp.Status=candidate</td></tr>
<tr><td>id</td><td>PositiveInteger</td><td>0..1</td><td>attr</td><td>This attribute represents the numerical identification of the defined security event. The identification shall be unique within the scope of the IDS.<br><b>Tags:</b>atp.Status=candidate</td></tr>
</table>

**Table 4.2: SecurityEventDefinition**

### 4.3.1 Properties of a Security Event

**[TPS_SECXT_01002]**{DRAFT} **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.⌋*(RS_SECXT_00001)*

**[TPS_SECXT_01000]**{DRAFT} **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.⌋*(RS_SECXT_00001)*

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

**[TPS_SECXT_01003]**{DRAFT} **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).⌋*(RS_SECXT_00001)*

**[constr_5600]**{DRAFT} **Valid interval for attribute SecurityEventDefinition.id** ⌈The valid interval for attribute SecurityEventDefinition.id is 0..65535.⌋*()*

**[constr_5601]**{DRAFT} **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.⌋*()*

**[TPS_SECXT_01004]**{DRAFT} **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.⌋*(RS_SECXT_00001)*

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 SecurityEventDefinitions while the limitation filters (the third step) are modeled separately because it applies to the whole IdsM Instance with all its SecurityEventDefinitions.

**[TPS_SECXT_02000]**{DRAFT} **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.⌋*(RS_SECXT_00012)*

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]**{DRAFT} **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.⌋*(RS_SECXT_00002)*

**[TPS_SECXT_01007]**{DRAFT} **Applicability of SecurityEventFilterChain towards SecurityEventDefinitions** ⌈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.⌋(RS_SECXT_00002)

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.

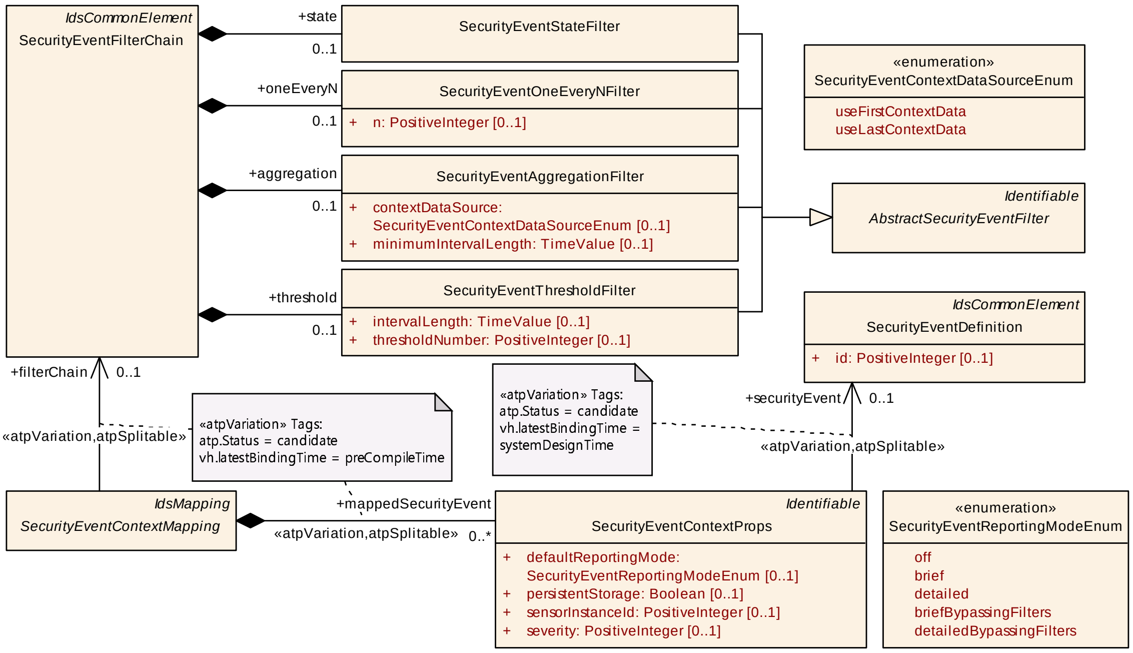
IdsCommonElement
SecurityEventFilterChain
+state
0..1
SecurityEventStateFilter
+oneEveryN
0..1
SecurityEventOneEveryNFilter
+ n: PositiveInteger [0..1]
+aggregation
0..1
SecurityEventAggregationFilter
+ contextDataSource: SecurityEventContextDataSourceEnum [0..1]
+ minimumIntervalLength: TimeValue [0..1]
+threshold
0..1
SecurityEventThresholdFilter
+ intervalLength: TimeValue [0..1]
+ thresholdNumber: PositiveInteger [0..1]
«enumeration»
SecurityEventContextDataSourceEnum
useFirstContextData
useLastContextData
Identifiable
AbstractSecurityEventFilter
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
+filterChain
0..1
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
+securityEvent
0..1
«atpVariation,atpSplitable»
IdsMapping
SecurityEventContextMapping
+mappedSecurityEvent
«atpVariation,atpSplitable»
0..*
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
«enumeration»
SecurityEventReportingModeEnum
off
brief
detailed
briefBypassingFilters
detailedBypassingFilters

**Figure 4.3: Modeling of SecurityEventFilterChain**

| | | | | |
|---|---|---|---|---|
| ***Class*** | **SecurityEventFilterChain** | | | |
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=SecurityFilterChains | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***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.<br>**Tags:**atp.Status=candidate |
| oneEveryN | SecurityEventOneEvery NFilter | 0..1 | aggr | This aggregation represents the sampling filter in the filter chain.<br>**Tags:**atp.Status=candidate |
| state | SecurityEventStateFilter | 0..1 | aggr | This aggregation represents the state filter in the event chain.<br>**Tags:**atp.Status=candidate |

▽

△

| *Class* | **SecurityEventFilterChain** | | | |
|---|---|---|---|---|
| threshold | SecurityEventThreshold Filter | 0..1 | aggr | This aggregation represents the threshold filter in the filter chain.<br>**Tags:**atp.Status=candidate |

**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) | | | |
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***Note*** | This meta-class acts as a base class for security event filters.<br>**Tags:**atp.Status=candidate | | | |
| ***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]**{DRAFT} **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 SecurityEventDefinition 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.⌋*(RS_SECXT_00002)*

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

**[constr_5613]**{DRAFT} **Unambiguous definition of SecurityEventStateFilter for CP or AP** ⌈For SecurityEventStateFilter, either the references in the role blockIfStateActiveCp or the references in the role blockIfStateActiveAp shall be defined in order to ensure the unambiguous applicability of the SecurityEventStateFilter towards the Classic or the Adaptive Platform.⌋*()*

**[constr_5615]**{DRAFT} **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 SecurityEventFilterChain 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.

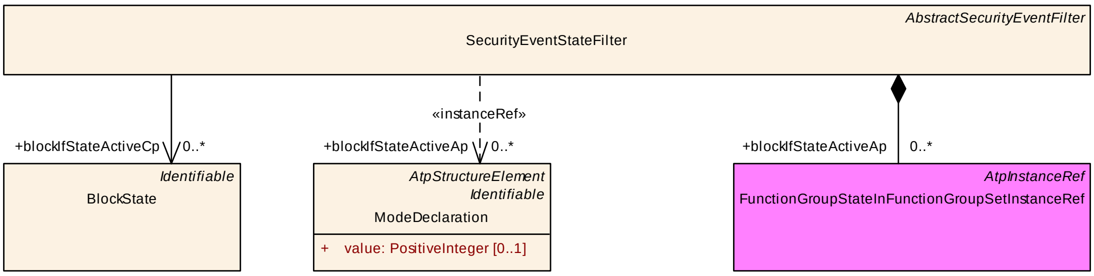

**Figure 4.4: Modeling overview of the SecurityEventStateFilter**

| ***Class*** | **SecurityEventStateFilter** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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).<br>**Tags:**atp.Status=candidate | | | |
| ***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.<br>**Tags:**atp.Status=candidate<br>**InstanceRef implemented by:**FunctionGroupStateIn FunctionGroupSetInstanceRef |

▽

△

| *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.<br>**Tags:**atp.Status=candidate |

**Table 4.5: SecurityEventStateFilter**

#### 4.4.2.1 SecurityEventStateFilter for the Classic Platform

**[TPS_SECXT_01045]**{DRAFT} **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 BlockStates referenced in the role blockIfStateActiveCp, then the IdsM shall discard the reported SecurityEventDefinition.⌋*(RS_SECXT_00002)*

#### 4.4.2.2 SecurityEventStateFilter for the Adaptive Platform

**[TPS_SECXT_01046]**{DRAFT} **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 machine state matches one of the machine states referenced in the role blockIfStateActiveAp, then the IdsM shall discard the reported SecurityEventDefinition.⌋*(RS_SECXT_00002)*

### 4.4.3 SecurityEventOneEveryNFilter

**[TPS_SECXT_01009]**{DRAFT} **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.⌋*(RS_SECXT_00002)*

**[constr_5602]**{DRAFT} **Valid interval for attribute SecurityEventOneEveryNFilter.n** ⌈The valid interval for attribute SecurityEventOneEveryNFilter.n is 1..65535.⌋*()*

| **Class** | **SecurityEventOneEveryNFilter** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **Note** | This meta-class represents the configuration of a sampling (i.e. every n-th event is sampled) filter for security events.<br>**Tags:**atp.Status=candidate | | | |
| **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.<br>**Tags:**atp.Status=candidate |

**Table 4.6: SecurityEventOneEveryNFilter**

### 4.4.4 SecurityEventAggregationFilter

**[TPS_SECXT_01010]**{DRAFT} **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.⌋*(RS_SECXT_00002)*

**[constr_5603]**{DRAFT} **Valid interval for attribute SecurityEventAggregationFilter.minimumIntervalLength** ⌈The valid interval for attribute SecurityEventAggregationFilter.minimumIntervalLength is ]0..INF[ seconds.⌋*()*

**[TPS_SECXT_01011]**{DRAFT} **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.⌋*(RS_SECXT_00002)*

| **Class** | **SecurityEventAggregationFilter** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **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.<br>**Tags:**atp.Status=candidate | | | |
| **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.<br>**Tags:**atp.Status=candidate |

**Table 4.7: SecurityEventAggregationFilter**

| ***Enumeration*** | **SecurityEventContextDataSourceEnum** |
|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate |
| ***Note*** | This enumeration controls the elements used to creating the resulting qualified security event<br>**Tags:**atp.Status=candidate |
| ***Aggregated by*** | SecurityEventAggregationFilter.contextDataSource |
| ***Literal*** | ***Description*** |
| useFirstContext Data | Context data of first received security event shall be used for resulting qualified security event.<br>**Tags:**<br>atp.EnumerationLiteralIndex=0<br>atp.Status=candidate |
| useLastContext Data | Context data of last received security event shall be used for resulting qualified security event.<br>**Tags:**<br>atp.EnumerationLiteralIndex=1<br>atp.Status=candidate |

**Table 4.8: SecurityEventContextDataSourceEnum**

### 4.4.5 SecurityEventThresholdFilter

**[TPS_SECXT_01012]**{DRAFT} **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.⌋*(RS_SECXT_00002)*

**[constr_5604]**{DRAFT} **Valid interval for attribute SecurityEventThresholdFilter.intervalLength** ⌈The valid interval for attribute SecurityEventThresholdFilter.intervalLength is ]0..INF[ seconds.⌋*()*

**[constr_5605]**{DRAFT} **Valid interval for attribute SecurityEventThresholdFilter.thresholdNumber** ⌈The valid interval for attribute SecurityEventThresholdFilter.thresholdNumber is 1..INF[.⌋*()*

| Class | SecurityEventThresholdFilter | | | |
|---|---|---|---|---|
| Package | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| 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.<br>**Tags:**atp.Status=candidate | | | |
| 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.<br>**Tags:**atp.Status=candidate |
| 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.<br>**Tags:**atp.Status=candidate |

**Table 4.9: SecurityEventThresholdFilter**

### 4.4.6 Final Qualification of a reported Security Event

**[TPS_SECXT_01013]**{DRAFT} **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)**.⌋*(RS_SECXT_00002, RS_SECXT_00007)*

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 SecurityEventDefinitions.

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.

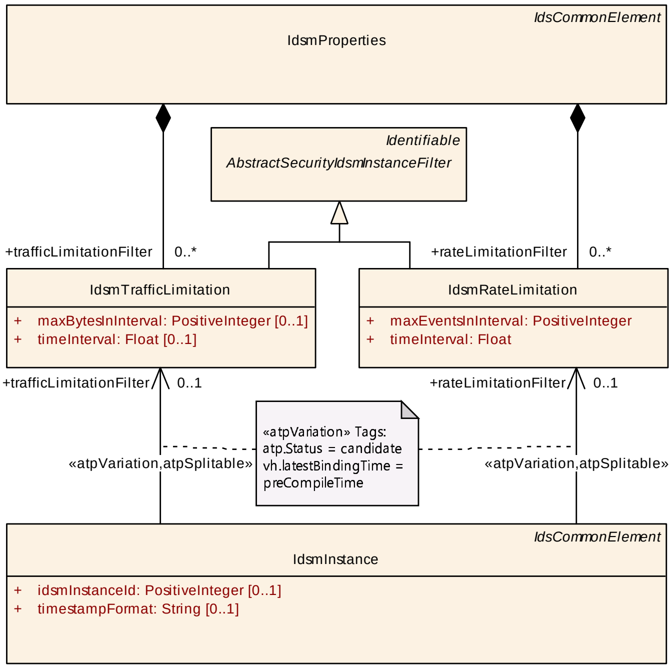
IdsCommonElement
IdsmProperties
Identifiable
AbstractSecurityIdsmInstanceFilter
+trafficLimitationFilter
0..*
+rateLimitationFilter
0..*
IdsmTrafficLimitation
+ maxBytesInInterval: PositiveInteger [0..1]
+ timeInterval: Float [0..1]
IdsmRateLimitation
+ maxEventsInInterval: PositiveInteger
+ timeInterval: Float
+trafficLimitationFilter
0..1
+rateLimitationFilter
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]

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

### 4.5.1 Rate Limitation Filter

**[TPS_SECXT_01014]**{DRAFT} **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.⌋*(RS_SECXT_00003)*

**[constr_5606]**{DRAFT} **Valid interval for attribute IdsmRateLimitation.timeInterval** ⌈The valid interval for attribute IdsmRateLimitation.timeInterval is 0..65535 seconds.⌋*()*

**[constr_5607]**{DRAFT} **Valid interval for attribute IdsmRateLimitation.maxEventsInInterval** ⌈The valid interval for attribute IdsmRateLimitation.maxEventsInInterval is $0..(2^{64} - 1)$.⌋*()*

| **Class** | **IdsmRateLimitation** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **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.<br>**Tags:**atp.Status=candidate | | | |
| **Base** | *ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| **Aggregated by** | IdsmProperties.rateLimitationFilter | | | |
| **Attribute** | **Type** | **Mult.** | **Kind** | **Note** |
| maxEventsInInterval | PositiveInteger | 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.<br>**Tags:**atp.Status=candidate |
| timeInterval | Float | 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.<br>**Tags:**atp.Status=candidate |

**Table 4.10: IdsmRateLimitation**

### 4.5.2 Traffic Limitation Filter

**[TPS_SECXT_01015]**{DRAFT} **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.⌋*(RS_SECXT_00003)*

**[constr_5608]**{DRAFT} **Valid interval for attribute IdsmTrafficLimitation.timeInterval** ⌈The valid interval for attribute IdsmTrafficLimitation.timeInterval is 0..65535 seconds.⌋*()*

**[constr_5609]**{DRAFT} **Valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval** ⌈The valid interval for attribute IdsmTrafficLimitation.maxBytesInInterval is $0..(2^{64} - 1)$.⌋*()*

| **Class** | **IdsmTrafficLimitation** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **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.<br>**Tags:**atp.Status=candidate | | | |
| **Base** | *ARObject*, *AbstractSecurityIdsmInstanceFilter*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| **Aggregated by** | IdsmProperties.trafficLimitationFilter | | | |
| **Attribute** | **Type** | **Mult.** | **Kind** | **Note** |

▽

△

| *Class* | **IdsmTrafficLimitation** | | | |
|---|---|---|---|---|
| maxBytesInInterval | 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.<br>**Tags:**atp.Status=candidate |
| 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.<br>**Tags:**atp.Status=candidate |

**Table 4.11: 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.

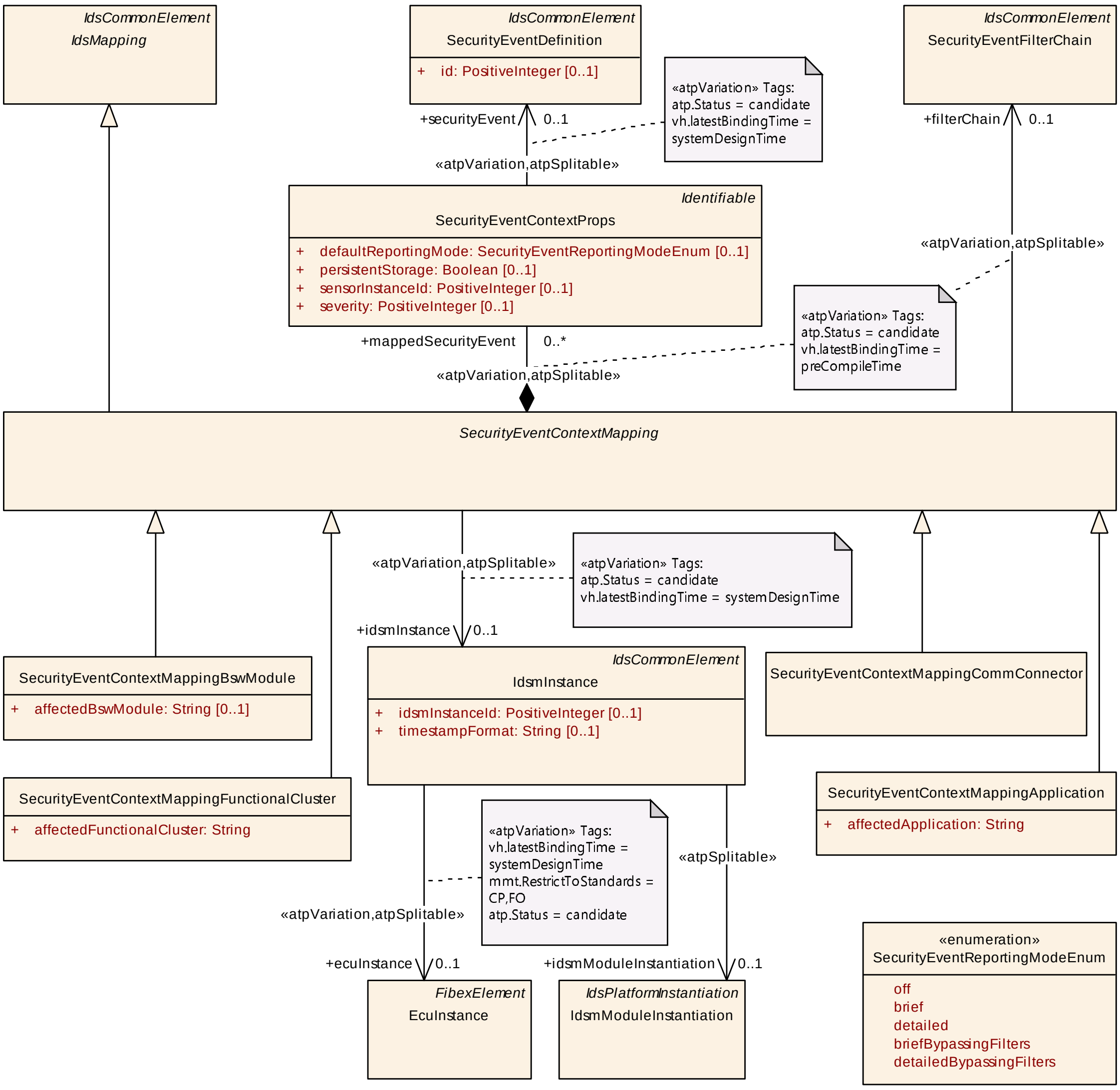
IdsCommonElement
IdsMapping
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
IdsCommonElement
SecurityEventFilterChain
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
+securityEvent
0..1
«atpVariation,atpSplitable»
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
+filterChain
0..1
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
+mappedSecurityEvent
0..*
«atpVariation,atpSplitable»
SecurityEventContextMapping
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
+idsmInstance
0..1
SecurityEventContextMappingBswModule
+ affectedBswModule: String [0..1]
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
SecurityEventContextMappingCommConnector
SecurityEventContextMappingFunctionalCluster
+ affectedFunctionalCluster: String
SecurityEventContextMappingApplication
+ affectedApplication: String
«atpVariation» Tags:
vh.latestBindingTime = systemDesignTime
mmt.RestrictToStandards = CP,FO
atp.Status = candidate
«atpSplitable»
«atpVariation,atpSplitable»
+ecuInstance
0..1
+idsmModuleInstantiation
0..1
FibexElement
EcuInstance
IdsPlatformInstantiation
IdsmModuleInstantiation
«enumeration»
SecurityEventReportingModeEnum
off
brief
detailed
briefBypassingFilters
detailedBypassingFilters

**Figure 4.6: Modeling overview on mapping of security events**

### 4.6.1 Mapping of Security Events to an IdsM Instance

**[TPS_SECXT_01016]**{DRAFT} **Semantics of SecurityEventContextMapping** ⌈The abstract meta-class SecurityEventContextMapping maps the SecurityEventDefinitions respectively referenced in the role securityEvent by the SecurityEventContextProps that are aggregated in the role mappedSecurityEvent to the IdsmInstance referenced in the role idsmInstance.⌋*(RS_SECXT_00004)*

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 SecurityEventDefinitions to an IdsmInstance implicitly defines the mapping of these SecurityEventDefinitions to an EcuInstance or to a Machine as well (for CP and AP, respectively).

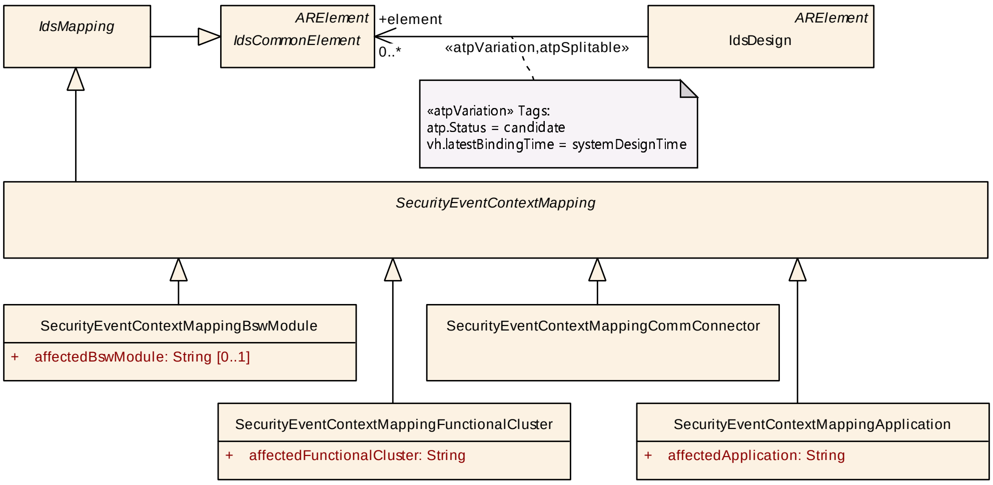
IdsMapping
ARElement
IdsCommonElement
+element
0..*
«atpVariation,atpSplitable»
ARElement
IdsDesign
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
SecurityEventContextMapping
SecurityEventContextMappingBswModule
+ affectedBswModule: String [0..1]
SecurityEventContextMappingCommConnector
SecurityEventContextMappingFunctionalCluster
+ affectedFunctionalCluster: String
SecurityEventContextMappingApplication
+ affectedApplication: String

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

| **Class** | ***SecurityEventContextMapping*** (abstract) | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **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.<br>**Tags:**atp.Status=candidate | | | |
| **Base** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| **Subclasses** | SecurityEventContextMappingApplication, SecurityEventContextMappingBswModule, SecurityEventContextMappingCommConnector, SecurityEventContextMappingFunctionalCluster | | | |
| **Aggregated by** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| filterChain | SecurityEventFilterChain | 0..1 | ref | This reference defines the filter chain to be applied to each of the referenced security events (depending on the reporting mode).<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=filterChain.securityEventFilterChain, filterChain.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=preCompileTime |
| idsmInstance | IdsmInstance | 0..1 | ref | This reference defines the IdsmInstance onto which the security events are mapped.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=idsmInstance.idsmInstance, idsmInstance.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=systemDesignTime |
| mappedSecurityEvent | SecurityEventContextProps | * | aggr | This aggregation represents (through further references) the SecurityEventDefinitions to be mapped to an IdsmInstance with additional mapping-dependent properties.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=mappedSecurityEvent.shortName, mappedSecurityEvent.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=preCompileTime |

**Table 4.12: SecurityEventContextMapping**

**[TPS_SECXT_01040]**{DRAFT} **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].⌋*(RS_SECXT_00001, RS_SECXT_00004)*

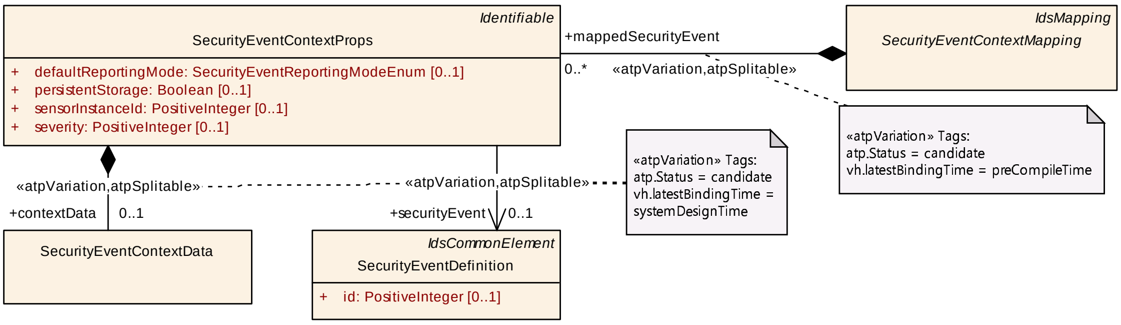
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
+mappedSecurityEvent
0..*
«atpVariation,atpSplitable»
IdsMapping
SecurityEventContextMapping
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime =
systemDesignTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
+contextData
0..1
+securityEvent
0..1
SecurityEventContextData
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]

**Figure 4.8: Modeling of SecurityEventContextProps**

| **Class** | **SecurityEventContextProps** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **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.<br>**Tags:**atp.Status=candidate | | | |
| **Base** | *ARObject, Identifiable, MultilanguageReferrable, Referrable* | | | |
| **Aggregated by** | *SecurityEventContextMapping.mappedSecurityEvent* | | | |
| **Attribute** | **Type** | **Mult.** | **Kind** | **Note** |
| contextData | SecurityEventContextData | 0..1 | aggr | This aggregation represents the definition of optional context data for security events.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=contextData, contextData.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=systemDesignTime |
| defaultReportingMode | SecurityEventReportingModeEnum | 0..1 | attr | This attribute defines the default reporting mode for the referenced security event.<br>**Tags:**atp.Status=candidate |
| persistentStorage | 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.<br>**Tags:**atp.Status=candidate |

▽

△

| *Class* | **SecurityEventContextProps** | | | |
|---|---|---|---|---|
| securityEvent | SecurityEventDefinition | 0..1 | ref | This reference defines the security event that is mapped and enriched by SecurityEventMappingProps with mapping dependent properties.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=securityEvent.securityEventDefinition, securityEvent.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=systemDesignTime |
| sensorInstanceId | PositiveInteger | 0..1 | attr | This attribute defines the ID of the security sensor that detects the referenced security event.<br>**Tags:**atp.Status=candidate |
| 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).<br>**Tags:**atp.Status=candidate |

**Table 4.13: SecurityEventContextProps**

#### 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]**{DRAFT} **Semantics of `SecurityEventContextData`** ⌈If additional context data can be added to a `SecurityEventDefinition` when it is reported to the IdsM, then `SecurityEventContextData` shall be aggregated by the `SecurityEventContextProps` which references the `SecurityEventDefinition` in the role `securityEvent`.⌋*(RS_SECXT_00009)*

Note: The aggregation of `SecurityEventContextData` by `SecurityEventContextData` means that the availability of context data for a `SecurityEventDefinition` is defined together with its mapping to an `IdsmInstance`, i.e. during the IDS Design phase and not during the Security Analysis phase (according to Ch. 3.1).

Modeling note: The aggregation of `SecurityEventContextData` which has (in this release) no attributes has been chosen as modeling approach in order to ensure better future extensibility.

#### 4.6.1.2 Default Reporting Mode definition

**[TPS_SECXT_01017]**{DRAFT} **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.

⌋*(RS_SECXT_00007)*

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

| ***Enumeration*** | **SecurityEventReportingModeEnum** |
|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate |
| ***Note*** | This enumeration controls the reporting mode of a security event.<br>**Tags:**atp.Status=candidate |
| ***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.<br>**Tags:**<br>atp.EnumerationLiteralIndex=1<br>atp.Status=candidate |
| briefBypassing Filters | The reported security event without its context data (if existing) is processed further but the filter chain is bypassed.<br>**Tags:**<br>atp.EnumerationLiteralIndex=3<br>atp.Status=candidate |
| detailed | The main properties and the context data (if existing) of the reported security event are processed further.<br>**Tags:**<br>atp.EnumerationLiteralIndex=2<br>atp.Status=candidate |
| detailedBypassing Filters | The reported security event including its context data (if existing) is processed further but the filter chain is bypassed.<br>**Tags:**<br>atp.EnumerationLiteralIndex=4<br>atp.Status=candidate |

▽

△

| *Enumeration* | **SecurityEventReportingModeEnum** |
|---|---|
| off | The reported security event is not further processed by the IdsM and therefore discarded.<br>**Tags:**<br>atp.EnumerationLiteralIndex=0<br>atp.Status=candidate |

**Table 4.14: SecurityEventReportingModeEnum**

#### 4.6.1.3 Persistent Storage definition

**[TPS_SECXT_01041]**{DRAFT} **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.

⌋*(RS_SECXT_00006)*

#### 4.6.1.4 Severity Level definition

**[TPS_SECXT_01042]**{DRAFT} **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].⌋*(RS_SECXT_00018)*

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]**{DRAFT} **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].⌋*(RS_SECXT_00023)*

### 4.6.2 Mapping of Security Events with BSW Module Context

**[TPS_SECXT_01018]**{DRAFT} **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.⌋*(RS_SECXT_00008)*

**[TPS_SECXT_01019]**{DRAFT} **Mapping of Security Events to Filter Chain by SecurityEventContextMappingBswModule** ⌈Each individual SecurityEventDefinition mapped through the SecurityEventContextPropss aggregated by SecurityEventContextMappingBswModule shall be input to the SecurityEventFilterChain referenced in the role filterChain by the same SecurityEventContextMappingBswModule.⌋*(RS_SECXT_00002, RS_SECXT_00008)*

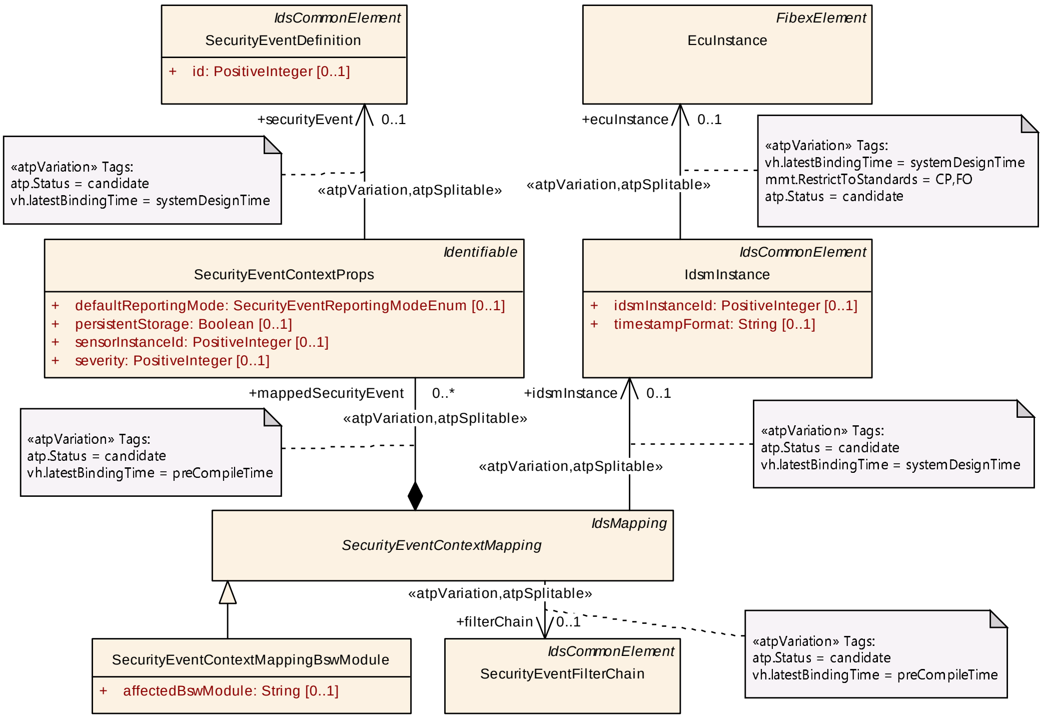
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
FibexElement
EcuInstance
+securityEvent
0..1
+ecuInstance
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
«atpVariation» Tags:
vh.latestBindingTime = systemDesignTime
mmt.RestrictToStandards = CP,FO
atp.Status = candidate
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
+mappedSecurityEvent
0..*
+idsmInstance
0..1
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
IdsMapping
SecurityEventContextMapping
«atpVariation,atpSplitable»
+filterChain
0..1
SecurityEventContextMappingBswModule
+ affectedBswModule: String [0..1]
IdsCommonElement
SecurityEventFilterChain
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime

**Figure 4.9: Modeling of SecurityEventContextMappingBswModule**

| ***Class*** | **SecurityEventContextMappingBswModule** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=SecurityEventContextMappingBswModules | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| 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.<br>**Tags:**atp.Status=candidate |

**Table 4.15: SecurityEventContextMappingBswModule**

### 4.6.3 Mapping of Security Events with Functional Cluster Context

**[TPS_SECXT_01020]**{DRAFT} **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`.⌋*(RS_SECXT_00008)*

**[TPS_SECXT_01021]**{DRAFT} **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`.⌋*(RS_SECXT_00002, RS_SECXT_00008)*

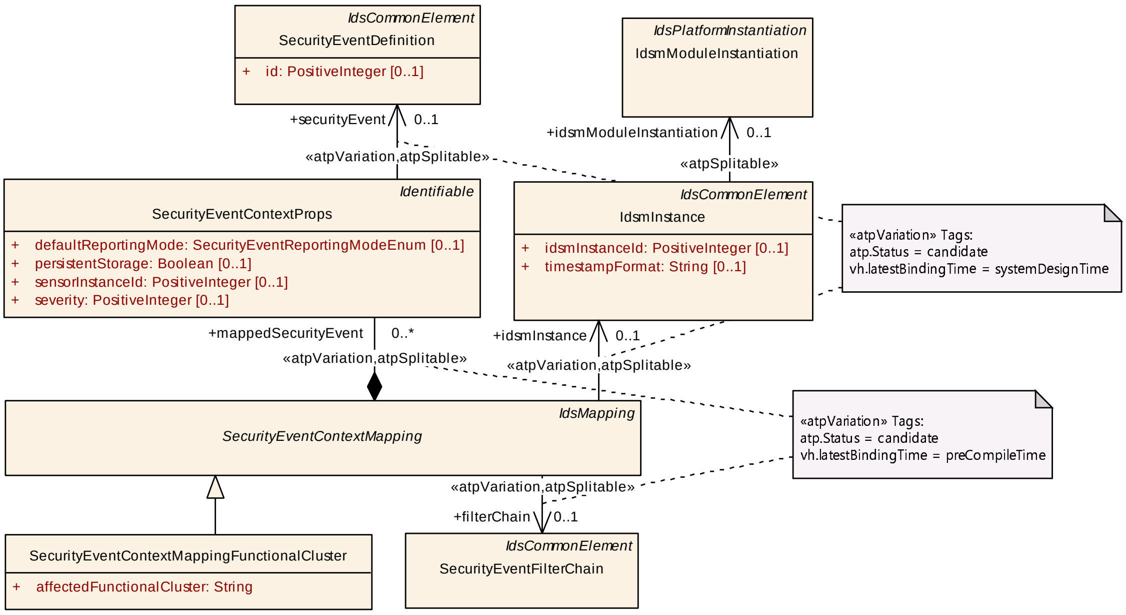
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
IdsPlatformInstantiation
IdsmModuleInstantiation
+securityEvent
0..1
«atpVariation,atpSplitable»
+idsmModuleInstantiation
0..1
«atpSplitable»
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
+mappedSecurityEvent
0..*
«atpVariation,atpSplitable»
+idsmInstance
0..1
«atpVariation,atpSplitable»
IdsMapping
SecurityEventContextMapping
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation,atpSplitable»
+filterChain
0..1
SecurityEventContextMappingFunctionalCluster
+ affectedFunctionalCluster: String
IdsCommonElement
SecurityEventFilterChain

Figure 4.10: Modeling of SecurityEventContextMappingFunctionalCluster

| *Class* | **SecurityEventContextMappingFunctionalCluster** | | | |
|---|---|---|---|---|
| *Package* | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| *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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=SecurityEventContextMappingFunctionalClusters | | | |
| *Base* | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping* | | | |
| *Aggregated by* | ARPackage.element | | | |
| *Attribute* | *Type* | *Mult.* | *Kind* | *Note* |
| affected Functional Cluster | String | 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.<br>**Tags:**atp.Status=candidate |

Table 4.16: SecurityEventContextMappingFunctionalCluster

### 4.6.4 Mapping of Security Events with Communication Connector Context

**[TPS_SECXT_01022]**{DRAFT} **Semantics of SecurityEventContextMappingCommConnector** ⌈SecurityEventContextMappingCommConnector defines that the mapped SecurityEventDefinitions can occur in the executional context related to the referenced CommunicationConnector in the role commConnector on the mapped IdsmInstance.⌋*(RS_SECXT_00005)*

**[TPS_SECXT_01023]**{DRAFT} **Mapping of Security Events to Filter Chain by `SecurityEventContextMappingCommConnector`** ⌈Each individual `SecurityEventDefinition` mapped through the `SecurityEventContextProps`s aggregated by `SecurityEventContextMappingCommConnector` shall be input to the `SecurityEventFilterChain` referenced in the role `filterChain` by the same `SecurityEventContextMappingCommConnector`.⌋*(RS_SECXT_00002, RS_SECXT_00005)*

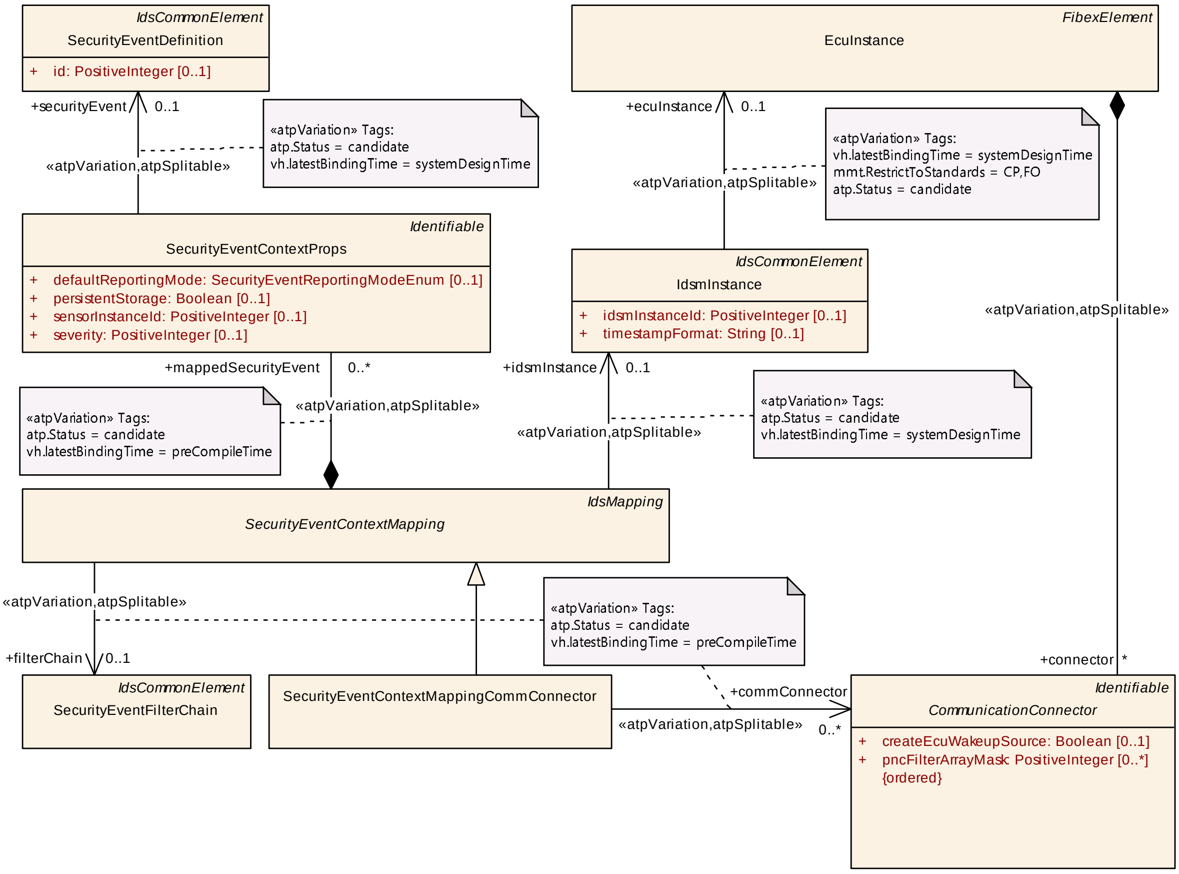
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
FibexElement
EcuInstance
+securityEvent
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
«atpVariation,atpSplitable»
+ecuInstance
0..1
«atpVariation» Tags:
vh.latestBindingTime = systemDesignTime
mmt.RestrictToStandards = CP,FO
atp.Status = candidate
«atpVariation,atpSplitable»
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
«atpVariation,atpSplitable»
+mappedSecurityEvent
0..*
+idsmInstance
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
IdsMapping
SecurityEventContextMapping
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
+filterChain
0..1
+connector
*
IdsCommonElement
SecurityEventFilterChain
SecurityEventContextMappingCommConnector
+commConnector
«atpVariation,atpSplitable»
0..*
Identifiable
CommunicationConnector
+ createEcuWakeupSource: Boolean [0..1]
+ pncFilterArrayMask: PositiveInteger [0..*]
{ordered}

**Figure 4.11: Modeling of `SecurityEventContextMappingCommConnector`**

| ***Class*** | **SecurityEventContextMappingCommConnector** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=SecurityEventContextMappingCommConnectors | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |

▽

△

| *Class* | **SecurityEventContextMappingCommConnector** | | | |
|---|---|---|---|---|
| comm Connector | Communication Connector | * | ref | This reference identifies the respective Communication Connector for which the collection of security events can be reported.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=commConnector.communicationConnector, commConnector.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=preCompileTime |

**Table 4.17: SecurityEventContextMappingCommConnector**

### 4.6.5 Mapping of Security Events with Application Context

**[TPS_SECXT_01024]**{DRAFT} **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.⌋*(RS_SECXT_00021)*

**[TPS_SECXT_01025]**{DRAFT} **Mapping of Security Events to Filter Chain by SecurityEventContextMappingApplication** ⌈Each individual SecurityEventDefinition mapped through the SecurityEventContextPropss aggregated by SecurityEventContextMappingApplication shall be input to the SecurityEventFilterChain referenced in the role filterChain by the same SecurityEventContextMappingApplication.⌋*(RS_SECXT_00002, RS_SECXT_00021)*

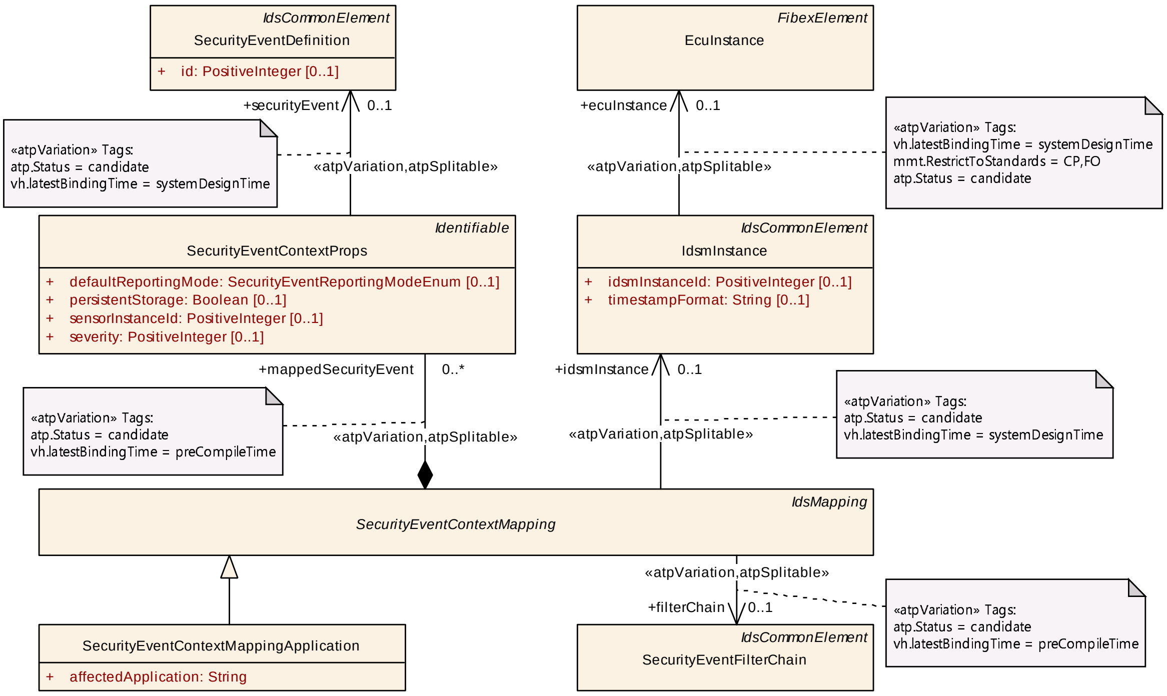
IdsCommonElement
SecurityEventDefinition
+ id: PositiveInteger [0..1]
FibexElement
EcuInstance
+securityEvent
0..1
+ecuInstance
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
«atpVariation» Tags:
vh.latestBindingTime = systemDesignTime
mmt.RestrictToStandards = CP,FO
atp.Status = candidate
Identifiable
SecurityEventContextProps
+ defaultReportingMode: SecurityEventReportingModeEnum [0..1]
+ persistentStorage: Boolean [0..1]
+ sensorInstanceId: PositiveInteger [0..1]
+ severity: PositiveInteger [0..1]
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
+mappedSecurityEvent
0..*
+idsmInstance
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = systemDesignTime
IdsMapping
SecurityEventContextMapping
«atpVariation,atpSplitable»
+filterChain
0..1
«atpVariation» Tags:
atp.Status = candidate
vh.latestBindingTime = preCompileTime
SecurityEventContextMappingApplication
+ affectedApplication: String
IdsCommonElement
SecurityEventFilterChain

Figure 4.12: Modeling of SecurityEventContextMappingApplication

| ***Class*** | **SecurityEventContextMappingApplication** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=SecurityEventContextMappingApplications | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *IdsMapping*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SecurityEventContextMapping* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| affected Application | String | 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).<br>**Tags:**atp.Status=candidate |

Table 4.18: SecurityEventContextMappingApplication

## 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]**{DRAFT} **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.⌋*(RS_SECXT_00013, RS_SECXT_00010)*

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 B.

**[TPS_SECXT_01027]**{DRAFT} **Semantics of IdsmInstance on AP** ⌈On the Adaptive Platform, the IdsmInstance represents an instance of the IdsM as defined by IdsmModuleInstantiation which is referenced in the role idsmModuleInstantiation.⌋*(RS_SECXT_00013)*

**[constr_5610]**{DRAFT} **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 idsmModuleInstantiation shall be defined in order to ensure that the platform (Classic or Adaptive) on which an IdsmInstance is targeted to run is unambiguously defined.⌋*()*

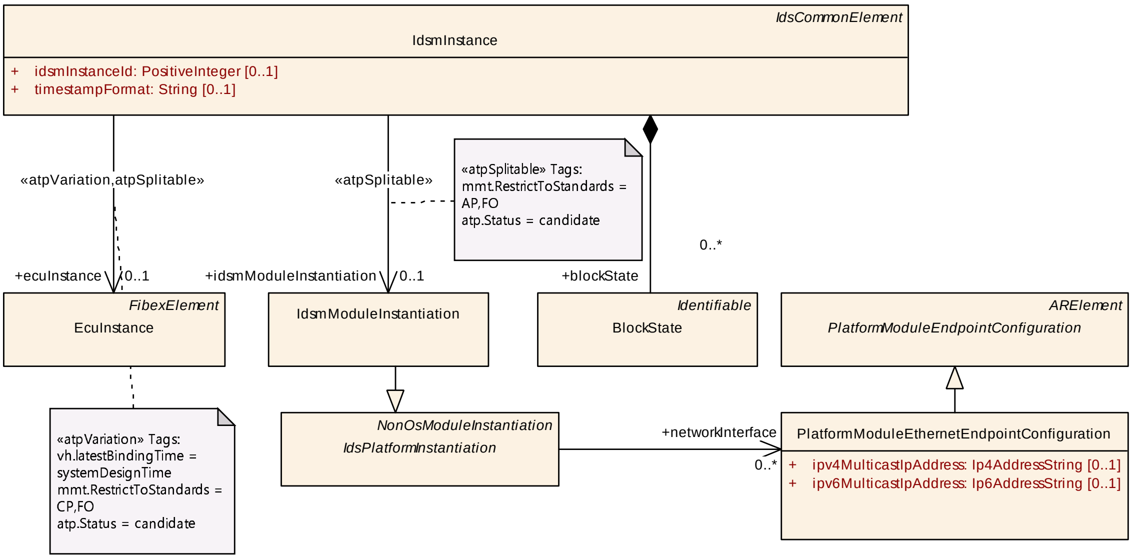

**Figure 4.13: Modeling overview of IdsmInstance**

| ***Class*** | **IdsmInstance** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=IdsmInstanceToEcuInstanceMappings | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| blockState | BlockState | * | aggr | This reference defines the BlockState in the collection BlockStateSet.<br>**Tags:**atp.Status=candidate |
| 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.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=systemDesignTime |
| idsmInstanceId | PositiveInteger | 0..1 | attr | This attribute is used to provide a source identification in the context of reporting security events..<br>**Tags:**atp.Status=candidate |
| idsmModuleInstantiation | IdsmModuleInstantiation | 0..1 | ref | This reference identifies the meta-class that defines the attributes for the IdsM configuration on a specific machine.<br>**Stereotypes:** atpSplitable<br>**Tags:**<br>atp.Splitkey=idsmModuleInstantiation<br>atp.Status=candidate |
| rateLimitationFilter | IdsmRateLimitation | 0..1 | ref | This reference identifies the applicable rate limitation filter for all security events on the related EcuInstance.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=rateLimitationFilter.idsmRateLimitation, rateLimitationFilter.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=preCompileTime |
| signatureSupportAp | IdsmSignatureSupportAp | 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 Adaptive Platform.<br>**Stereotypes:** atpSplitable<br>**Tags:**<br>atp.Splitkey=signatureSupportAp<br>atp.Status=candidate |

▽

△

| ***Class*** | **IdsmInstance** | | | |
|---|---|---|---|---|
| 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.<br>**Stereotypes:** atpSplitable<br>**Tags:**<br>atp.Splitkey=signatureSupportCp<br>atp.Status=candidate |
| 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.<br>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.<br>Note: A string defining a proprietary timestamp format shall be prefixed by a company-specific name fragment to avoid collisions.<br>**Tags:**atp.Status=candidate |
| trafficLimitation Filter | IdsmTrafficLimitation | 0..1 | ref | This reference identifies the applicable traffic limitation filter for all security events on the related EcuInstance.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=trafficLimitationFilter.idsmTrafficLimitation, trafficLimitationFilter.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=preCompileTime |

**Table 4.19: 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]**{DRAFT} **Semantics of attribute `IdsmInstance.idsmInstanceId`** ⌈The attribute `idsmInstanceId` of `IdsmInstance` defines the assigned identifier for the IdsM instance.⌋*(RS_SECXT_00013)*

#### 4.7.1.2 Timestamp in QSEv messages

**[TPS_SECXT_01029]**{DRAFT} **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.⌋*(RS_SECXT_00014)*

**[TPS_SECXT_01030]**{DRAFT} **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.

⌋*(RS_SECXT_00015)*

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]**{DRAFT} **Definition of signature support for an IdsmInstance** ⌈For an IdsmInstance, the existence of the reference in the role signatureSupportCp (for the Classic Platform) or in the role signatureSupportAp (for the Adaptive Platform) 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.⌋*(RS_SECXT_00016)*

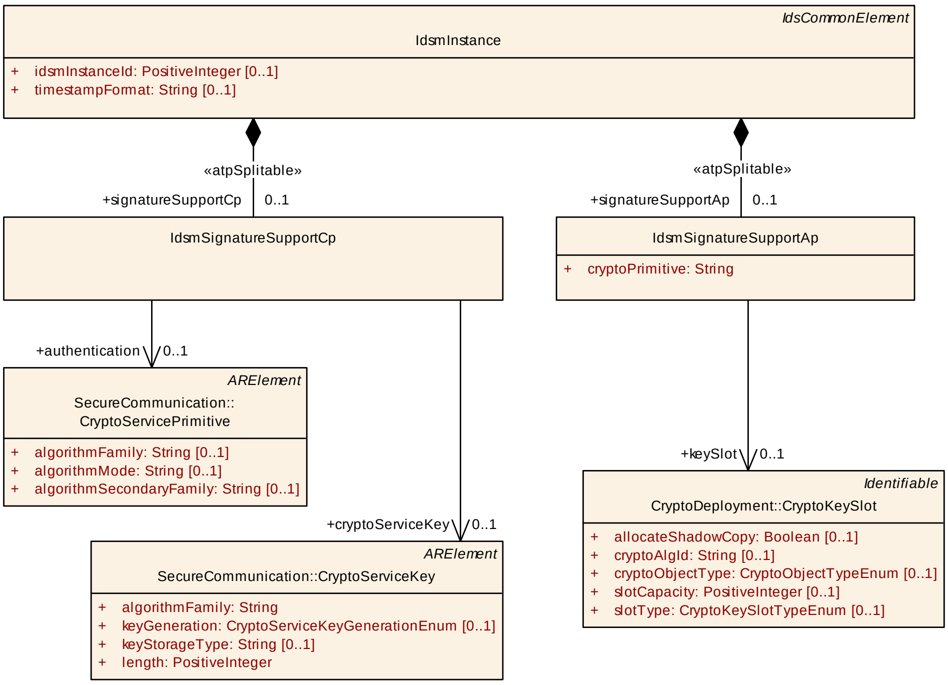
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
«atpSplitable»
+signatureSupportCp
0..1
«atpSplitable»
+signatureSupportAp
0..1
IdsmSignatureSupportCp
IdsmSignatureSupportAp
+ cryptoPrimitive: String
+authentication
0..1
ARElement
SecureCommunication::
CryptoServicePrimitive
+ algorithmFamily: String [0..1]
+ algorithmMode: String [0..1]
+ algorithmSecondaryFamily: String [0..1]
+keySlot
0..1
Identifiable
CryptoDeployment::CryptoKeySlot
+ allocateShadowCopy: Boolean [0..1]
+ cryptoAlgId: String [0..1]
+ cryptoObjectType: CryptoObjectTypeEnum [0..1]
+ slotCapacity: PositiveInteger [0..1]
+ slotType: CryptoKeySlotTypeEnum [0..1]
+cryptoServiceKey
0..1
ARElement
SecureCommunication::CryptoServiceKey
+ algorithmFamily: String
+ keyGeneration: CryptoServiceKeyGenerationEnum [0..1]
+ keyStorageType: String [0..1]
+ length: PositiveInteger

**Figure 4.14: Modeling overview on signature support for an IdsmInstance**

Depending on whether the IdsmInstance is deployed on the Classic or the Adaptive Platform, either IdsmSignatureSupportCp or IdsmSignatureSupportAp shall be used for configuration of signature calculation.

**[TPS_SECXT_01032]**{DRAFT} **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.

⌋*(RS_SECXT_00016)*

**[TPS_SECXT_01033]**{DRAFT} **Semantics of IdsmSignatureSupportAp** ⌈For the Adaptive Platform, IdsmSignatureSupportAp represents the configuration of signature support for the aggregating IdsmInstance:

- The attribute cryptoPrimitive defines the cryptographic algorithm to be used as specified by the Cryptographic Primitives Naming Convention in [7].
- The reference in the role keySlot to CryptoKeySlot defines the cryptographic key to be used.

⌋*(RS_SECXT_00016)*

**[constr_5611]**{DRAFT} **Unambiguous configuration of platform-dependent signature support for an IdsmInstance** ⌈For the meta-class IdsmInstance, either the aggregation of IdsmSignatureSupportCp or of IdsmSignatureSupportAp shall be defined in order to ensure that the platform-dependent signature support is unambiguously configured.⌋*()*

### 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]**{DRAFT} **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.⌋*(RS_SECXT_00004, RS_SECXT_00008)*

**[TPS_SECXT_01035]**{DRAFT} **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.⌋*(RS_SECXT_00004, RS_SECXT_00008)*

**[TPS_SECXT_01036]**{DRAFT} **Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingCommConnector** ⌈For all SecurityEventContextMappingCommConnector 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.⌋*(RS_SECXT_00004, RS_SECXT_00005)*

**[TPS_SECXT_01037]**{DRAFT} **Association of SecurityEventDefinitions with an IdsmInstance through SecurityEventContextMappingApplication** ⌈For all SecurityEventContextMappingApplication 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.⌋*(RS_SECXT_00004, RS_SECXT_00021)*

### 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]**{DRAFT} **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 idsmModuleInstantiation (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 GeneralPurposeIPdu to communicate with the PduR and thus send QSEv messages onto the network.

**[TPS_SECXT_01038]**{DRAFT} **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.⌋*(RS_SECXT_00017)*

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

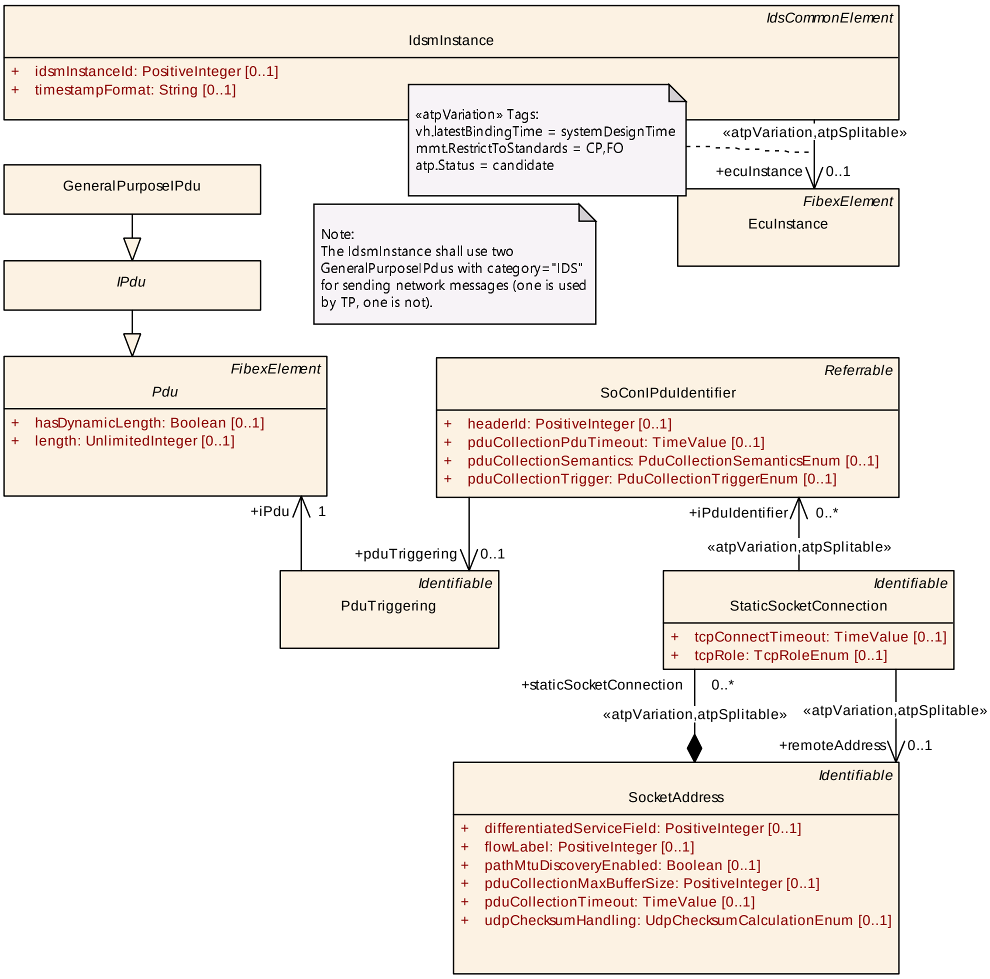
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
«atpVariation» Tags:
vh.latestBindingTime = systemDesignTime
mmt.RestrictToStandards = CP,FO
atp.Status = candidate
«atpVariation,atpSplitable»
+ecuInstance
0..1
FibexElement
EcuInstance
GeneralPurposeIPdu
IPdu
Note:
The IdsmInstance shall use two GeneralPurposeIPdus with category="IDS" for sending network messages (one is used by TP, one is not).
FibexElement
Pdu
+ hasDynamicLength: Boolean [0..1]
+ length: UnlimitedInteger [0..1]
Referrable
SoConIPduIdentifier
+ headerId: PositiveInteger [0..1]
+ pduCollectionPduTimeout: TimeValue [0..1]
+ pduCollectionSemantics: PduCollectionSemanticsEnum [0..1]
+ pduCollectionTrigger: PduCollectionTriggerEnum [0..1]
+iPdu
1
+pduTriggering
0..1
+iPduIdentifier
0..*
«atpVariation,atpSplitable»
Identifiable
PduTriggering
Identifiable
StaticSocketConnection
+ tcpConnectTimeout: TimeValue [0..1]
+ tcpRole: TcpRoleEnum [0..1]
+staticSocketConnection
0..*
«atpVariation,atpSplitable»
«atpVariation,atpSplitable»
+remoteAddress
0..1
Identifiable
SocketAddress
+ differentiatedServiceField: PositiveInteger [0..1]
+ flowLabel: PositiveInteger [0..1]
+ pathMtuDiscoveryEnabled: Boolean [0..1]
+ pduCollectionMaxBufferSize: PositiveInteger [0..1]
+ pduCollectionTimeout: TimeValue [0..1]
+ udpChecksumHandling: UdpChecksumCalculationEnum [0..1]

**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]**{DRAFT} **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 referenced by the IdsmInstance in the role idsmModuleInstantiation.⌋*(RS_SECXT_00017)*

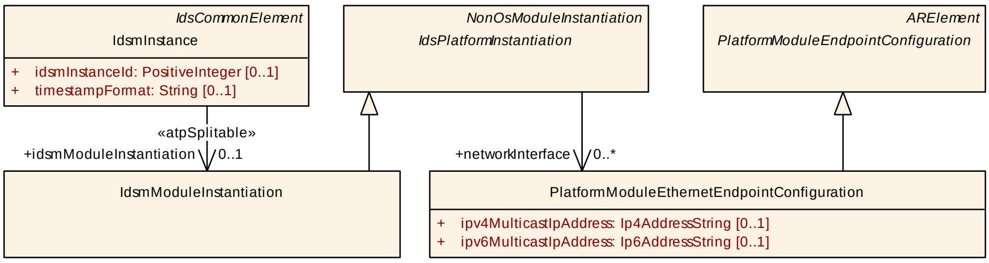
IdsCommonElement
IdsmInstance
+ idsmInstanceId: PositiveInteger [0..1]
+ timestampFormat: String [0..1]
NonOsModuleInstantiation
IdsPlatformInstantiation
ARElement
PlatformModuleEndpointConfiguration
«atpSplitable»
+idsmModuleInstantiation
0..1
+networkInterface
0..*
IdsmModuleInstantiation
PlatformModuleEthernetEndpointConfiguration
+ ipv4MulticastIpAddress: Ip4AddressString [0..1]
+ ipv6MulticastIpAddress: Ip6AddressString [0..1]

**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]**{DRAFT} **Definition of BlockStates on CP** ⌈On the Classic Platform, when a SecurityEventStateFilter is configured as part of a SecurityEventFilterChain, then the BlockStates 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.⌋ *(RS_SECXT_00002)*

Note: Since the BlockStates are named and identified using their respective shortNames, the uniqueness of their naming within an IdsmInstance is inherently given.

**[TPS_SECXT_01044]**{DRAFT} **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.⌋ *(RS_SECXT_00002)*

**[constr_5614]**{DRAFT} **Upper bound for multiplicity of BlockStates aggregated by IdsmInstance** ⌈For the meta-class IdsmInstance, the maximum number of aggregated BlockStates 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 [8].

Please also refer to Ch. 4.4.2.1.

| ***Class*** | **BlockState** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| ***Aggregated by*** | IdsmInstance.blockState | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table 4.20: BlockState**

# A Mentioned Class Tables

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

| ***Class*** | ***ARElement*** (abstract) | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage | | | |
| ***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*** | AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, *AutosarDataType*, *BaseType*, BlueprintMappingSet, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecification MappingSet, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, DataConstr, Data ExchangePoint, DataTransformationSet, DataTypeMappingSet, *DiagnosticCommonElement*, Diagnostic Connection, DiagnosticContributionSet, DltContext, DltEcu, Documentation, E2EProfileCompatibility Props, EndToEndProtectionSet, EthIpProps, EthTcpIpIcmpProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FunctionGroupSet, General PurposeConnection, HwCategory, HwElement, HwType, IPSecConfigProps, *IdsCommonElement*, Ids Design, InterpolationRoutineMappingSet, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKayProps, MacSecParticipantSet, McFunction, Mc Group, ModeDeclarationGroup, ModeDeclarationMappingSet, PhysicalDimension, PhysicalDimension MappingSet, *PlatformModuleEndpointConfiguration*, *PortInterface*, PortInterfaceMappingSet, Port PrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, SignalServiceTranslationPropsSet, SoftwareCluster, SomeipSd ClientEventGroupTimingConfig, SomeipSdClientServiceInstanceConfig, SomeipSdServerEventGroup TimingConfig, SomeipSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, *SwComponent Type*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, System, SystemSignal, System SignalGroup, *TimingExtension*, TlvDataIdDefinitionSet, TransformationPropsSet, Unit, UnitGroup, View MapSet | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.1: ARElement**

| ***Class*** | **ARPackage** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage | | | |
| ***Note*** | AUTOSAR package, allowing to create top level packages to structure the contained ARElements.<br>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.<br>This is an extended version of MSR's SW-SYSTEM. | | | |
| ***Base*** | *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.arPackage, AUTOSAR.arPackage | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |

▽

△

| ***Class*** | **ARPackage** | | | |
|---|---|---|---|---|
| arPackage | ARPackage | * | aggr | This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=arPackage.shortName, arPackage.variationPoint.shortLabel<br>vh.latestBindingTime=blueprintDerivationTime<br>xml.sequenceOffset=30 |
| element | PackageableElement | * | aggr | Elements that are part of this package<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=element.shortName, element.variationPoint.shortLabel<br>vh.latestBindingTime=systemDesignTime<br>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.<br>**Stereotypes:** atpSplitable<br>**Tags:**<br>atp.Splitkey=referenceBase.shortLabel<br>xml.sequenceOffset=10 |

**Table A.2: ARPackage**

| ***Class*** | **AUTOSAR** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::AutosarTopLevelStructure | | | |
| ***Note*** | Root element of an AUTOSAR description, also the root element in corresponding XML documents.<br>**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.<br>**Tags:**xml.sequenceOffset=10 |
| arPackage | ARPackage | * | aggr | This is the top level package in an AUTOSAR model.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=arPackage.shortName, arPackage.variationPoint.shortLabel<br>vh.latestBindingTime=blueprintDerivationTime<br>xml.sequenceOffset=30 |
| fileInfoComment | FileInfoComment | 0..1 | aggr | This represents a possibility to provide a structured comment in an AUTOSAR file.<br>**Stereotypes:** atpStructuredComment<br>**Tags:**<br>xml.roleElement=true<br>xml.sequenceOffset=-10<br>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.<br>**Tags:**xml.sequenceOffset=20 |

**Table A.3: AUTOSAR**

| | | | | |
|---|---|---|---|---|
| ***Class*** | ***CommunicationConnector*** (abstract) | | | |
| ***Package*** | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology | | | |
| ***Note*** | The connection between the referencing ECU and the referenced channel via the referenced controller.<br>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.<br>Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance. | | | |
| ***Base*** | *ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| ***Subclasses*** | *AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunicationConnector, UserDefinedCommunicationConnector | | | |
| ***Aggregated by*** | EcuInstance.connector, MachineDesign.communicationConnector | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| createEcuWakeupSource | 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. |
| pncFilterArrayMask (ordered) | PositiveInteger | * | attr | Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management.<br>**Tags:**atp.Status=draft |

**Table A.4: CommunicationConnector**

| | | | | |
|---|---|---|---|---|
| ***Class*** | **CryptoKeySlot** | | | |
| ***Package*** | M2::AUTOSARTemplates::AdaptivePlatform::PlatformModuleDeployment::CryptoDeployment | | | |
| ***Note*** | This meta-class represents the ability to define a concrete key to be used for a crypto operation.<br>**Tags:**atp.ManifestKind=MachineManifest | | | |
| ***Base*** | *ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| ***Aggregated by*** | CryptoProvider.keySlot | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| allocateShadowCopy | Boolean | 0..1 | attr | This attribute defines whether a shadow copy of this Key Slot shall be allocated to enable rollback of a failed Key Slot update campaign (see interface BeginTransaction). |
| cryptoAlgId | String | 0..1 | attr | This attribute defines a crypto algorithm restriction (kAlgId Any means without restriction). The algorithm can be specified partially: family & length, mode, padding.<br>Future Crypto Providers can support some crypto algorithms that are not well known/ standardized today, therefore AUTOSAR doesn't provide a concrete list of crypto algorithms' identifiers and doesn't suppose usage of numerical identifiers. Instead of this a provider supplier should provide string names of supported algorithms in accompanying documentation. The name of a crypto algorithm shall follow the rules defined in the specification of cryptography for Adaptive Platform. |
| cryptoObjectType | CryptoObjectTypeEnum | 0..1 | attr | Object type that can be stored in the slot. If this field contains "Undefined" then mSlotCapacity must be provided and larger then 0.<br>**Tags:**atp.Status=candidate |
| keySlotAllowedModification | CryptoKeySlotAllowedModification | 0..1 | aggr | Restricts how this keySlot may be used<br>**Tags:**atp.Status=candidate |
| keySlotContentAllowedUsage | CryptoKeySlotContentAllowedUsage | * | aggr | Restriction of allowed usage of a key stored to the slot.<br>**Tags:**atp.Status=candidate |

▽

△

| ***Class*** | **CryptoKeySlot** | | | |
|---|---|---|---|---|
| slotCapacity | PositiveInteger | 0..1 | attr | Capacity of the slot in bytes to be reserved by the stack vendor. One use case is to define this value in case that the cryptoObjectType is undefined and the slot size can not be deduced from cryptoObjectType and cryptoAlgId. "0" means slot size can be deduced from cryptoObject Type and cryptoAlgId. |
| slotType | CryptoKeySlotType Enum | 0..1 | attr | This attribute defines whether the keySlot is exclusively used by the Application; or whether it is used by Stack Services and managed by a Key Manager Application.<br>**Tags:**atp.Status=candidate |

**Table A.5: CryptoKeySlot**

| ***Class*** | **CryptoServiceKey** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SystemTemplate::SecureCommunication | | | |
| ***Note*** | This meta-class has the ability to represent a crypto key.<br>**Tags:**atp.recommendedPackage=CryptoDevelopmentKeys | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| algorithmFamily | String | 1 | attr | This attribute represent the description of the family of the applicable crypto algorithm. |
| development Value | ValueSpecification | 0..1 | aggr | This aggregation represents the ability to assign a specific value to the crypto key as part of the system description. This value can then be taken for the development of the respective ECU. |
| keyGeneration | CryptoServiceKey GenerationEnum | 0..1 | attr | This attribute describes how a the specific cryptographic key is created. |
| keyStorageType | String | 0..1 | attr | This attribute describes where the enclosing cryptographic key shall be stored. AUTOSAR reserves specific values for this attributes but it is possible to insert custom values as well. |
| length | PositiveInteger | 1 | attr | This attribute describes the length of the cryptographic key in bits. |

**Table A.6: CryptoServiceKey**

| ***Class*** | **CryptoServicePrimitive** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SystemTemplate::SecureCommunication | | | |
| ***Note*** | This meta-class has the ability to represent a crypto primitive.<br>**Tags:**atp.recommendedPackage=CryptoPrimitives | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable* | | | |
| ***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. |

▽

△

| ***Class*** | **CryptoServicePrimitive** | | | |
|---|---|---|---|---|
| algorithm Secondary Family | String | 0..1 | attr | This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive.<br>The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA. |

**Table A.7: CryptoServicePrimitive**

| ***Class*** | **EcuInstance** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology | | | |
| ***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.<br>**Tags:**atp.recommendedPackage=EcuInstances | | | |
| ***Base*** | *ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| associated Consumed Provided ServiceInstance Group | ConsumedProvided ServiceInstanceGroup | * | ref | With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=associatedConsumedProvidedService InstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstance Group.variationPoint.shortLabel<br>vh.latestBindingTime=postBuild |
| associatedPdur IPduGroup | PdurIPduGroup | * | ref | With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU. |
| 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. |
| commController | Communication Controller | 1..* | aggr | CommunicationControllers of the ECU.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=commController.shortName, comm Controller.variationPoint.shortLabel<br>vh.latestBindingTime=postBuild |
| connector | Communication Connector | * | aggr | All channels controlled by a single controller.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=connector.shortName, connector.variation Point.shortLabel<br>vh.latestBindingTime=postBuild |
| 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. |

▽

△

| **Class** | **EcuInstance** | | | |
|---|---|---|---|---|
| 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 A.8: EcuInstance**

| **Class** | **GeneralPurposeIPdu** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| **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.<br>**Tags:**atp.recommendedPackage=Pdus | | | |
| **Base** | *ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable* | | | |
| **Aggregated by** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.9: GeneralPurposeIPdu**

| **Class** | ***Identifiable*** (abstract) |
|---|---|
| **Package** | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable |
| **Note** | Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables. |
| **Base** | *ARObject*, *MultilanguageReferrable*, *Referrable* |
| **Subclasses** | ARPackage, *AbstractDoIpLogicAddressProps*, *AbstractEvent*, *AbstractImplementationDataTypeElement*, *AbstractSecurityEventFilter*, *AbstractSecurityIdsmInstanceFilter*, *AbstractServiceInstance*, Application Endpoint, ApplicationError, ArtifactChecksum, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpFeature*, AutosarOperationArgumentInstance, AutosarVariableInstance, BlockState, *BuildActionEntity*, BuildAction Environment, Chapter, ClassContentConditional, ClientIdDefinition, ClientServerOperation, Code, *CollectableElement*, ComManagementMapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, *Coupling PortStructuralElement*, CryptoKeySlot, *CryptoServiceMapping*, DataPrototypeGroup, Data Transformation, DependencyOnArtifact, *DiagEventDebounceAlgorithm*, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, DiagnosticFunctionInhibitSource, DiagnosticParameterElement, *DiagnosticRoutineSubfunction*, DltApplication, DltArgument, DltMessage, DoIpInterface, DoIpLogicAddress, DoIpRoutingActivation, EndToEndProtection, EthernetWakeupSleep OnDatalineConfig, EventHandler, ExclusiveArea, *ExecutableEntity*, *ExecutionTime*, FMAttributeDef, FM FeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeatureRelation, FM FeatureRestriction, FMFeatureSelection, FlexrayArTpNode, FlexrayTpPduPool, *FrameTriggering*, GeneralParameter, GlobalTimeGateway, *GlobalTimeMaster*, *GlobalTimeSlave*, *HeapUsage*, HwAttribute Def, HwAttributeLiteralDef, HwPin, HwPinGroup, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPdu Mapping, ISignalTriggering, *IdentCaption*, InternalTriggeringPoint, Keyword, LifeCycleState, Linker, Mac MulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, ModeDeclaration, Mode DeclarationMapping, ModeSwitchPoint, NetworkEndpoint, *NmCluster*, *NmNode*, *PackageableElement*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstance Memory, *PhysicalChannel*, PortGroup, *PortInterfaceMapping*, PossibleErrorReaction, Resource Consumption, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, Rpt ExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntityGroup, *Sdg Attribute*, SdgClass, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, ▽ |

▽

△

| ***Class*** | ***Identifiable*** (abstract) | | | |
|---|---|---|---|---|
| | △<br>SecurityEventContextProps, *ServiceNeeds*, SignalServiceTranslationEventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, *SpecElementReference*, *StackUsage*, StaticSocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SystemMapping, *TimeBaseResource*, *TimingClock*, TimingClockSyncAccuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, TimingModeInstance, Topic1, TpAddress, TraceableTable, TraceableText, *TracedFailure*, *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.<br>**Stereotypes:** atpSplitable<br>**Tags:**<br>atp.Splitkey=adminData<br>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.<br>**Tags:**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.<br>**Tags:**xml.sequenceOffset=-50 |
| desc | MultiLanguageOverviewParagraph | 0..1 | aggr | This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question.<br>More elaborate documentation, (in particular how the object is built or used) should go to "introduction".<br>**Tags:**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.<br>**Tags:**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.<br>**Tags:**xml.attribute=true |

**Table A.10: Identifiable**

| ***Class*** | ***IdsCommonElement*** (abstract) | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***Subclasses*** | *IdsMapping*, IdsmInstance, IdsmProperties, SecurityEventDefinition, SecurityEventFilterChain | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.11: IdsCommonElement**

| ***Class*** | ***IdsMapping*** (abstract) | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***Note*** | This meta-class serves as abstract base class for mappings related to an IDS design.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***Subclasses*** | *SecurityEventContextMapping* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.12: IdsMapping**

| ***Class*** | ***IdsPlatformInstantiation*** (abstract) | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::AdaptivePlatform::PlatformModuleDeployment::IntrusionDetectionSystem | | | |
| ***Note*** | This meta-class acts as an abstract base class for platform modules that implement the intrusion detection system.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARObject*, *AdaptiveModuleInstantiation*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *NonOsModuleInstantiation*, *Referrable* | | | |
| ***Subclasses*** | IdsmModuleInstantiation | | | |
| ***Aggregated by*** | *AtpClassifier*.atpFeature, Machine.moduleInstantiation | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| networkInterface | PlatformModuleEthernetEndpointConfiguration | * | ref | This association contains the network configuration that shall be applied to an instance of an IDS entity.<br>**Tags:**atp.Status=candidate |
| timeBase | TimeBaseResource | 0..1 | ref | This reference identifies the applicable time base resource.<br>**Stereotypes:** atpSplitable; atpVariation<br>**Tags:**<br>atp.Splitkey=timeBase.timeBaseResource, timeBase.variationPoint.shortLabel<br>atp.Status=candidate<br>vh.latestBindingTime=systemDesignTime |

**Table A.13: IdsPlatformInstantiation**

| | | | | |
|---|---|---|---|---|
| ***Class*** | **IdsmModuleInstantiation** | | | |
| ***Package*** | M2::AUTOSARTemplates::AdaptivePlatform::PlatformModuleDeployment::IntrusionDetectionSystem | | | |
| ***Note*** | This meta-class defines the attributes for the IdsM configuration on a specific machine.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARObject*, *AdaptiveModuleInstantiation*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *IdsPlatformInstantiation*, *MultilanguageReferrable*, *NonOsModuleInstantiation*, *Referrable* | | | |
| ***Aggregated by*** | *AtpClassifier*.atpFeature, Machine.moduleInstantiation | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.14: IdsmModuleInstantiation**

| | | | | |
|---|---|---|---|---|
| ***Class*** | **IdsmProperties** | | | |
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***Note*** | This meta-class provides the ability to aggregate filters for security events.<br>**Tags:**<br>atp.Status=candidate<br>atp.recommendedPackage=IdsMPropertiess | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *IdsCommonElement*, *MultilanguageReferrable*, *PackageableElement*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| rateLimitationFilter | IdsmRateLimitation | * | aggr | This aggregation represents the collection of rate limitation filters for security events in the enclosing SecurityFilterSet.<br>**Tags:**atp.Status=candidate |
| trafficLimitationFilter | IdsmTrafficLimitation | * | aggr | This aggregation represents the collection of traffic limitation filters for security events in the enclosing SecurityFilterSet.<br>**Tags:**atp.Status=candidate |

**Table A.15: IdsmProperties**

| | | | | |
|---|---|---|---|---|
| ***Class*** | **IdsmSignatureSupportAp** | | | |
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***Note*** | This meta-class defines, for the Adaptive Platform, the cryptographic algorithm and key to be used by the IdsM instance for providing signature information in QSEv messages.<br>**Tags:**atp.Status=candidate | | | |
| ***Base*** | *ARObject* | | | |
| ***Aggregated by*** | IdsmInstance.signatureSupportAp | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| cryptoPrimitive | String | 1 | attr | This attribute defines the cryptographic algorithm to be used for providing authentication information in QSEv messages. The content of this attribute shall comply to the "Cryptographic Primitives Naming Convention".<br>**Tags:**atp.Status=candidate |
| keySlot | CryptoKeySlot | 0..1 | ref | This reference denotes the cryptographic key to be used by the cryptographic algorithm for providing authentication information in QSEv messages.<br>**Tags:**atp.Status=candidate |

**Table A.16: IdsmSignatureSupportAp**

| ***Class*** | **IdsmSignatureSupportCp** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| ***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.<br>**Tags:**atp.Status=candidate | | | |
| ***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.<br>**Tags:**atp.Status=candidate |
| cryptoServiceKey | CryptoServiceKey | 0..1 | ref | This reference denotes the cryptographic key to be used by the cryptographic algorithm for providing authentication information in QSEv messages.<br>**Tags:**atp.Status=candidate |

**Table A.17: IdsmSignatureSupportCp**

| ***Class*** | **MultiLanguageOverviewParagraph** | | | |
|---|---|---|---|---|
| ***Package*** | M2::MSR::Documentation::TextModel::MultilanguageData | | | |
| ***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, ScaleConstr.desc, SdgCaption.desc, SwRecordLayoutGroup.desc, SwRecordLayoutV.desc, VariationPoint.desc | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| l2 | LOverviewParagraph | 1..* | aggr | This represents the text in one particular language.<br>**Tags:**<br>xml.roleElement=true<br>xml.roleWrapperElement=false<br>xml.sequenceOffset=20<br>xml.typeElement=false<br>xml.typeWrapperElement=false |

**Table A.18: MultiLanguageOverviewParagraph**

| ***Class*** | **PlatformModuleEthernetEndpointConfiguration** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::AdaptivePlatform::PlatformModuleDeployment::AdaptiveModuleImplementation | | | |
| ***Note*** | This meta-class defines the attributes for the configuration of a port, protocol type and IP address of the communication on a VLAN.<br>**Tags:**atp.recommendedPackage=PlatformModuleEndpointConfigurations | | | |
| ***Base*** | *ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PlatformModuleEndpointConfiguration*, *Referrable* | | | |
| ***Aggregated by*** | ARPackage.element | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| communicationConnector | EthernetCommunicationConnector | 0..1 | ref | Reference to the CommunicationConnector (VLAN) for which the network configuration is defined. |
| ipv4MulticastIpAddress | Ip4AddressString | 0..1 | attr | Multicast IPv4 Address to which the message will be transmitted. |
| ipv6MulticastIpAddress | Ip6AddressString | 0..1 | attr | Multicast IPv6 Address to which the message will be transmitted. |

**Table A.19: PlatformModuleEthernetEndpointConfiguration**

| **Class** | ***PortPrototype*** (abstract) | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| **Note** | Base class for the ports of an AUTOSAR software component.<br>The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports. | | | |
| **Base** | *ARObject*, *AtpBlueprintable*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable* | | | |
| **Subclasses** | *AbstractProvidedPortPrototype*, *AbstractRequiredPortPrototype* | | | |
| **Aggregated by** | *AtpClassifier*.atpFeature, *SwComponentType*.port | | | |
| **Attribute** | **Type** | **Mult.** | **Kind** | **Note** |
| – | – | – | – | – |

Table A.20: PortPrototype

| **Class** | ***Referrable*** (abstract) | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable | | | |
| **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, SocketConnectionBundle, TimeSyncServerConfiguration, 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.<br>**Stereotypes:** atpIdentityContributor<br>**Tags:**<br>xml.enforceMinMultiplicity=true<br>xml.sequenceOffset=-100 |
| shortNameFragment | ShortNameFragment | * | aggr | This specifies how the Referrable.shortName is composed of several shortNameFragments.<br>**Tags:**xml.sequenceOffset=-90 |

Table A.21: Referrable

| **Class** | **SecurityEventContextData** | | | |
|---|---|---|---|---|
| **Package** | M2::AUTOSARTemplates::SecurityExtractTemplate | | | |
| **Note** | This meta-class represents the possibility that context data can be attached to the aggregating SecurityEventDefinition. If this meta-class does not exist for a SecurityEventDefinition, then no context data shall be provided for this SecurityEventDefinition.<br>**Tags:**atp.Status=candidate | | | |
| **Base** | *ARObject* | | | |
| **Aggregated by** | SecurityEventContextProps.contextData | | | |
| **Attribute** | **Type** | **Mult.** | **Kind** | **Note** |
| – | – | – | – | – |

Table A.22: SecurityEventContextData

| ***Class*** | **SymbolProps** | | | |
|---|---|---|---|---|
| ***Package*** | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| ***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. AtomicSw ComponentType, that is a potential subject to a name clash on the level of RTE source code.<br>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, *PortInterface*.namespace, SecurityEventDefinition.eventSymbolName | | | |
| ***Attribute*** | ***Type*** | ***Mult.*** | ***Kind*** | ***Note*** |
| – | – | – | – | – |

**Table A.23: SymbolProps**

# B Upstream Mapping

## B.1 Introduction

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template, ECU Resource Template, Diagnostic Extract Template and Security Extract Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

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

Please note that the tables contain the following columns:

**bsw module:** Name of BSW module

**bsw context:** Reference to parameter container

**bsw type:** Type of parameter

**bsw param:** Name of the BSW parameter

**bsw desc:** Description from the configuration document

**m2 template:** System Template, SW Component Template, ECU Resource Template

**m2 param:** Name of the upstream template parameter

**m2 description:** Description from the upstream template definition

**mapping rule:** Textual description on how to transform between M2 and BSW domains

**mapping type:**

- local: no mapping needed since parameter local to BSW
- partial: some data can be automatically mapped but not all
- full: all data can be automatically mapped

## B.2 IdsM

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>IdsM</td><td colspan="3">IdsM/IdsMConfiguration</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">IdsMBlockState</td><td colspan="2">ECUC-PARAM-CONF-CONTAINER-DEF</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">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().</td></tr>
<tr><th colspan="4">Template Description</th></tr>
<tr><td colspan="4">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.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SecurityExtractTemplate::BlockState</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">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.</td><td>full</td></tr>
<tr><th colspan="3">Mapping Status</th><th>ECUC Parameter ID</th></tr>
<tr><td colspan="3">valid</td><td>[ECUC_IdsM_00020]</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>IdsM</td><td colspan="3">IdsM/IdsMConfiguration/IdsMBufferConfiguration</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">IdsMContextDataBuffer</td><td colspan="2">ECUC-PARAM-CONF-CONTAINER-DEF</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Buffer that is reserved to store the context data of SEvs.<br>Depending on the type of SEv that is processed, there can be significant differences in sizes of the context data.</td></tr>
<tr><th colspan="4">Template Description</th></tr>
<tr><td colspan="4">This meta-class represents the possibility that context data can be attached to the aggregating SecurityEventDefinition. If this meta-class does not exist for a SecurityEventDefinition, then no context data shall be provided for this SecurityEvent Definition.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SecurityExtractTemplate::SecurityEventContextData</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">In the SECXT, the context data availability is defined per SecurityEventContextProps while in the EcuC of IdsM, the context data buffers are configured "globally"(i.e. IdsM-wide). Therefore, the correct context data buffer configuration for the IdsM needs to be derived from all (M2) Security EventContextProps that are mapped to the (M2) IdsmInstance and which aggregate a (M2) SecurityEventContextData.</td><td>partial</td></tr>
<tr><th colspan="3">Mapping Status</th><th>ECUC Parameter ID</th></tr>
<tr><td colspan="3">valid</td><td>[ECUC_IdsM_00046]</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>IdsM</td><td colspan="3">IdsM/IdsMConfiguration/IdsMBufferConfiguration/IdsMContextDataBuffer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">IdsMContextDataBufferSize</td><td colspan="2">ECUC-INTEGER-PARAM-DEF</td></tr>
</table>

▽

△

| | |
|---|---|
| **BSW Description** | |
| Size of the context data buffer in bytes. It is recommended to configure buffers with an appropriate size depending on the configured SEvs. | |
| **Template Description** | |
| This meta-class represents the possibility that context data can be attached to the aggregating SecurityEventDefinition. If this meta-class does not exist for a SecurityEventDefinition, then no context data shall be provided for this SecurityEvent Definition. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::SecurityEventContextData | |
| **Mapping Rule** | **Mapping Type** |
| In the SECXT, the context data availability is defined per SecurityEventContextProps while in the EcuC of IdsM, the context data buffers are configured "globally"(i.e. IdsM-wide). Therefore, the correct context data buffer configuration for the IdsM needs to be derived from all (M2) Security EventContextProps that are mapped to the (M2) IdsmInstance and which aggregate a (M2) SecurityEventContextData. | partial |
| **Mapping Status** | **ECUC Parameter ID** |
| valid | [ECUC_IdsM_00047] |

| **BSW Module** | **BSW Context** |
|---|---|
| IdsM | IdsM/IdsMConfiguration/IdsMBufferConfiguration/IdsMContextDataBuffer |
| **BSW Parameter** | **BSW Type** |
| IdsMNumberOfContextDataBuffers | ECUC-INTEGER-PARAM-DEF |
| **BSW Description** | |
| The number of buffers with the configured buffer size specified in IdsMContextDataBufferSize. It is recommended to configure an appropriate number of buffers depending on the configured SEvs. | |
| **Template Description** | |
| This meta-class represents the possibility that context data can be attached to the aggregating SecurityEventDefinition. If this meta-class does not exist for a SecurityEventDefinition, then no context data shall be provided for this SecurityEvent Definition. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::SecurityEventContextData | |
| **Mapping Rule** | **Mapping Type** |
| In the SECXT, the context data availability is defined per SecurityEventContextProps while in the EcuC of IdsM, the context data buffers are configured "globally"(i.e. IdsM-wide). Therefore, the correct context data buffer configuration for the IdsM needs to be derived from all (M2) Security EventContextProps that are mapped to the (M2) IdsmInstance and which aggregate a (M2) SecurityEventContextData. | partial |
| **Mapping Status** | **ECUC Parameter ID** |
| valid | [ECUC_IdsM_00048] |

| **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** | |
| SecurityExtractTemplate::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** |
| 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:<br>Standarized SEv ID is defined by the AUTOSAR specification. This ID is usually derived from the SecXT. Standard ID range: 0x0000 - 0x8000<br>Generic User Event ID is defined by the user. Used when a SW-C / Application references the SEv. Generic ID range: 0x8000 - 0xFFFE.<br>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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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:<br>- 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** | |
| SecurityExtractTemplate::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** | |
| | |
| **Template Description** | |
| Only the main security event properties such as its ID are processed. Any additional context data (if existing) is discarded. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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** | |
| | |
| **Template Description** | |
| The reported security event without its context data (if existing) is processed further but the filter chain is bypassed. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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** | |
| | |
| **Template Description** | |
| The main properties and the context data (if existing) of the reported security event are processed further. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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** | |
| | |
| **Template Description** | |
| The reported security event including its context data (if existing) is processed further but the filter chain is bypassed. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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** | |
| | |
| **Template Description** | |
| The reported security event is not further processed by the IdsM and therefore discarded. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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.<br>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** | |
| SecurityExtractTemplate::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.<br>A filter receives a SEv, checks condition(s) and, e.g. - forwards SEv immediately/later - drops SEv - stores SEv - modifies SEv<br>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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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.<br>Instead, only the last or the first received SEv is stored in an aggregation buffer, depending on the configuration of "IdsMContextDataSourceSelector".<br>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.<br>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** | |
| SecurityExtractTemplate::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:<br>IDSM_FILTERS_CTX_USE_FIRST = ContextData of first received SEv is used for resulting QSEv.<br>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** | |
| SecurityExtractTemplate::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).<br>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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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).<br>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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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** |
| 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** | |
| SecurityExtractTemplate::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".<br>The limit is measured in number of incoming SEvs.<br>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.<br>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** | |
| SecurityExtractTemplate::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** | |
| SecurityExtractTemplate::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).<br>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** | |
| SecurityExtractTemplate::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".<br>The limit is measured in incoming amount of bytes.<br>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".<br>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** | |
| SecurityExtractTemplate::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** | |

▽

△

<table>
<tr><td colspan="2">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.</td></tr>
<tr><th colspan="2">M2 Parameter</th></tr>
<tr><td colspan="2">SecurityExtractTemplate::IdsmTrafficLimitation.maxBytesInInterval</td></tr>
<tr><th>Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td>1:1 mapping</td><td>full</td></tr>
<tr><th>Mapping Status</th><th>ECUC Parameter ID</th></tr>
<tr><td>valid</td><td>[ECUC_IdsM_00058]</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>IdsM</td><td colspan="3">IdsM/IdsMGeneral/IdsMGlobalRateLimitationFilters/IdsMFilterTrafficLimitation</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">IdsMTrafficLimitationTimeInterval</td><td colspan="2">ECUC-FLOAT-PARAM-DEF</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Length of the traffic limitation time interval (as float in seconds).<br>Note: Shall be configured as a multiple of the IdsM main function period.</td></tr>
<tr><th colspan="4">Template Description</th></tr>
<tr><td colspan="4">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.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SecurityExtractTemplate::IdsmTrafficLimitation.timeInterval</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">1:1 mapping</td><td>full</td></tr>
<tr><th colspan="3">Mapping Status</th><th>ECUC Parameter ID</th></tr>
<tr><td colspan="3">valid</td><td>[ECUC_IdsM_00057]</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>IdsM</td><td colspan="3">IdsM/IdsMGeneral</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">IdsMInstanceId</td><td colspan="2">ECUC-INTEGER-PARAM-DEF</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">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.<br>Note: There is only one IdsM (from the AUTOSAR Classic Platform) instance per ECU.</td></tr>
<tr><th colspan="4">Template Description</th></tr>
<tr><td colspan="4">This attribute is used to provide a source identification in the context of reporting security events..</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SecurityExtractTemplate::IdsmInstance.idsmInstanceId</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">1:1 mapping</td><td>full</td></tr>
<tr><th colspan="3">Mapping Status</th><th>ECUC Parameter ID</th></tr>
<tr><td colspan="3">valid</td><td>[ECUC_IdsM_00007]</td></tr>
</table>

| | |
|---|---|
| **BSW Module** | **BSW Context** |
| IdsM | IdsM/IdsMGeneral |
| **BSW Parameter** | **BSW Type** |
| IdsMSignature | ECUC-PARAM-CONF-CONTAINER-DEF |
| **BSW Description** | |
| If this container exists all qualified security events are signed 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** | |
| SecurityExtractTemplate::`IdsmSignatureSupportCp` | |
| **Mapping Rule** | **Mapping Type** |
| If the aggregation in the role (M2) signatureSupportCp exists, then the (M1) subcontainer IdsMSignature 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/IdsMSignature |
| **BSW Parameter** | **BSW Type** |
| IdsMSignatureLength | ECUC-INTEGER-PARAM-DEF |
| **BSW Description** | |
| This parameter defines the length of the signature in bytes calculated by the crypto service. | |
| **Template Description** | |
| **CryptoServicePrimitive:**<br>This meta-class has the ability to represent a crypto primitive.<br>**CryptoServiceKey:**<br>This meta-class has the ability to represent a crypto key. | |
| **M2 Parameter** | |
| SystemTemplate::SecureCommunication::`CryptoServicePrimitive`, SystemTemplate::SecureCommunication::`CryptoServiceKey` | |
| **Mapping Rule** | **Mapping Type** |
| The (M1) IdsMSignatureLength needs to be deduced from the configured (M2) CryptoServicePrimitive 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** |
| IdsMSignatureSupport | ECUC-BOOLEAN-PARAM-DEF |
| **BSW Description** | |
| This parameter enables/disables the functionality of sending messages to the network with a signature of encryption 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** | |
| SecurityExtractTemplate::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** |
| 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.<br>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.<br>Note: A string defining a proprietary timestamp format shall be prefixed by a company-specific name fragment to avoid collisions. | |
| **M2 Parameter** | |
| SecurityExtractTemplate::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.<br>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.<br>Note: A string defining a proprietary timestamp format shall be prefixed by a company-specific name fragment to avoid collisions. | |
| **M2 Parameter** | |

▽

△

| SecurityExtractTemplate::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] |

# C Splitable Elements in the Scope of this Document

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

Each entry in Table C.1 consists of the identification of the specific model element itself and the applicable value of the tagged value atp.Splitkey.

For more information about the concept of splitable model elements and how these shall be treated please refer to [9].

| ***Name of splitable element*** | ***Splitkey*** |
|---|---|
| IdsDesign.element | element.idsCommonElement, element.variationPoint.shortLabel |
| IdsmInstance.ecuInstance | ecuInstance.ecuInstance, ecuInstance.variationPoint.shortLabel |
| IdsmInstance.idsmModuleInstantiation | idsmModuleInstantiation |
| IdsmInstance.rateLimitationFilter | rateLimitationFilter.idsmRateLimitation, rateLimitationFilter.variationPoint.shortLabel |
| IdsmInstance.signatureSupportAp | signatureSupportAp |
| IdsmInstance.signatureSupportCp | signatureSupportCp |
| IdsmInstance.trafficLimitationFilter | trafficLimitationFilter.idsmTrafficLimitation, trafficLimitationFilter.variationPoint.shortLabel |
| SecurityEventContextMapping.filterChain | filterChain.securityEventFilterChain, filterChain.variationPoint.shortLabel |
| SecurityEventContextMapping.idsmInstance | idsmInstance.idsmInstance, idsmInstance.variationPoint.shortLabel |
| SecurityEventContextMapping.mappedSecurityEvent | mappedSecurityEvent.shortName, mappedSecurityEvent.variationPoint.shortLabel |
| SecurityEventContextMappingCommConnector.commConnector | commConnector.communicationConnector, commConnector.variationPoint.shortLabel |
| SecurityEventContextProps.contextData | contextData, contextData.variationPoint.shortLabel |
| SecurityEventContextProps.securityEvent | securityEvent.securityEventDefinition, securityEvent.variationPoint.shortLabel |
| SecurityEventDefinition.eventSymbolName | eventSymbolName.shortName |

**Table C.1: Usage of splitable elements**

# D 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 Table D.1 consists of the identification of the model element itself and the applicable value of the tagged value vh.latestBindingTime.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to [9].

| ***Variation Point*** | ***Latest Binding Time*** |
|---|---|
| IdsDesign.element | systemDesignTime |
| IdsmInstance.ecuInstance | systemDesignTime |
| IdsmInstance.rateLimitationFilter | preCompileTime |
| IdsmInstance.trafficLimitationFilter | preCompileTime |
| SecurityEventContextMapping.filterChain | preCompileTime |
| SecurityEventContextMapping.idsmInstance | systemDesignTime |
| SecurityEventContextMapping.mappedSecurityEvent | preCompileTime |
| SecurityEventContextMappingCommConnector.commConnector | preCompileTime |
| SecurityEventContextProps.contextData | systemDesignTime |
| SecurityEventContextProps.securityEvent | systemDesignTime |

**Table D.1: Usage of variation points**

# E History of Constraints and Specification Items

Please note that the lists in this chapter also include constraints and specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

## E.1 Constraint and Specification Item History of this document according to AUTOSAR Release R20-11

### E.1.1 Added Traceables in R20-11

| Number | Heading |
|---|---|
| [TPS_SECXT_-001043] | 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 SecurityEventDefinitions |
| [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.defaultReportingMode |
| [TPS_SECXT_01018] | Semantics of SecurityEventMappingContextBswModule |
| [TPS_SECXT_01019] | Mapping of Security Events to Filter Chain by SecurityEventMappingContextBswModule |
| [TPS_SECXT_01020] | Semantics of SecurityEventMappingContextFunctionalCluster |
| [TPS_SECXT_01021] | Mapping of Security Events to Filter Chain by SecurityEventMappingContextFunctionalCluster |

▽

△

| Number | Heading |
|---|---|
| [TPS_SECXT_01022] | Semantics of SecurityEventMappingContextCommConnector |
| [TPS_SECXT_01023] | Mapping of Security Events to Filter Chain by SecurityEventMappingContextCommConnector |
| [TPS_SECXT_01024] | Semantics of SecurityEventMappingContextApplication |
| [TPS_SECXT_01025] | Mapping of Security Events to Filter Chain by SecurityEventMappingContextApplication |
| [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 E.1: Added Traceables in R20-11**

### E.1.2 Changed Traceables in R20-11

none

### E.1.3 Deleted Traceables in R20-11

none

### E.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 E.2: Added Constraints in R20-11**

### E.1.5 Changed Constraints in R20-11

none

### E.1.6 Deleted Constraints in R20-11

none

## E.2 Constraint and Specification Item History of this document according to AUTOSAR Release R22-11

### E.2.1 Added Traceables in R22-11

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

**Table E.3: Added Traceables in R22-11**

### E.2.2 Changed Traceables in R22-11

none

### E.2.3 Deleted Traceables in R22-11

none

### E.2.4 Added Constraints in R22-11

none

### E.2.5 Changed Constraints in R22-11

none

### E.2.6 Deleted Constraints in R22-11

none

# F Glossary - Terms and Acronyms

## F.1 Terms

| 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 F.1: Terms**

## F.2 Acronyms

| Acronym | Description |
|---|---|
| ARXML | AUTOSAR XML, i.e. AUTOSAR Extensible Markup Language |
| ECU | Electronic Control Unit (in AUTOSAR context, an ECU runs a single AUTOSAR Basic Software of the Classic Platform) |
| ECU-HW | Electronic Control Unit Hardware, i.e. the physical housing of one or more (possibly virtual) Classic Platform ECUs and/or Adaptive Platform Machines |
| FC | Functional Cluster |
| IDS | Intrusion Detection System |
| IdsM | Intrusion Detection System Manager |
| IdsR | Intrusion Detection System Reporter |
| OEM | Original Equipment Manufacturer |
| SECXT | Security Extract |
| SEv | Security Event |
| QSEv | Qualified Security Event |
| Sem | Security Event Memory |
| SIEM | Security Incident and Event Management |

| Acronym | Description |
|---|---|
| SOC | Security Operation Centre |
| SOP | Start Of Production |
| SWCL | Software Cluster |

**Table F.2: Acronyms**