| Document Title | Specification of Firewall for Classic Platform |
|---|---|
| **Document Category** | Software specification (SWS) |
| **Document UID** | 1084 |
| **Document Status** | published |
| **Part of AUTOSAR Standard** | Classic Platform |
| **Part of Standard Release** | R26-11 |

| Document Change History | | | |
|---|---|---|---|
| **Date** | **Release** | **Changed by** | **Description** |
| 2026-11-26 | R26-11 | AUTOSAR Release Management | • Removed GetVersionInfo API<br>• IdsM related interface and imported types are listed again<br>• Removed upwards tracing to `FO_RS_Firewall`<br>• Removed extended production errors and restructured chapter 8 |
| 2025-11-27 | R25-11 | AUTOSAR Release Management | • Updated context data for security events generated by the firewall |
| 2024-11-27 | R24-11 | AUTOSAR Release Management | • Changed reception interface from TcpIp to LSduR<br>• Updated SEv context data specification table |
| 2023-11-23 | R23-11 | AUTOSAR Release Management | • Initial release |

**Disclaimer**

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

# Table of Contents

1 Introduction and functional overview 6
2 Acronyms and Abbreviations 7
2.1 Acronyms . . . 7
2.2 Abbreviations . . . 8
3 Related documentation 9
3.1 Input documents & related standards and norms . . . 9
3.2 Related specification . . . 10
4 Constraints and assumptions 11
4.1 Known Limitations . . . 11
4.2 Applicability to car domains . . . 11
5 Dependencies to other modules 12
6 Requirements Tracing 13
7 Functional specification 14
7.1 Overview . . . 14
7.2 Module handling . . . 15
7.2.1 Initialization . . . 15
7.2.2 Timing Related Functionality . . . 16
7.3 Network packet inspection . . . 16
7.3.1 Stateless packet inspection . . . 17
7.3.1.1 Inspection of not modeled protocols . . . 19
7.3.2 Stateful packet inspection . . . 20
7.3.3 Deep packet inspection . . . 20
7.3.3.1 SOME/IP . . . 21
7.3.3.2 DDS . . . 23
7.3.3.3 DoIP . . . 23
7.3.3.4 Generic inspection . . . 24
7.4 Network packet filtering . . . 24
7.4.1 Allowlists and Blocklists . . . 25
7.4.2 Rate limiting . . . 26
7.4.3 State dependent filtering . . . 27
7.5 Firewall interaction with the switch . . . 28
7.5.1 Packet inspection by AUTOSAR firewall module . . . 30
7.5.2 Network packets blocked by the switch core . . . 30
7.5.2.1 SEvs on protocol level . . . 31
7.5.2.2 Switch firewall rule counting statistics SEv . . . 31
7.5.3 Management of firewall rules in the switch core . . . 33
7.6 Security Events . . . 33
7.6.1 SEvs raised by the firewall . . . 34

7.6.2 Raising SEvs . . . 41
7.7 Error Classification . . . 45
7.7.1 Development Errors . . . 45
7.7.2 Runtime Errors . . . 45
7.7.3 Production Errors . . . 45
8 API specification 46
8.1 Imported types . . . 46
8.2 Type definitions . . . 46
8.2.1 ConfigType . . . 46
8.3 Function definitions . . . 46
8.3.1 Functions provided for EcuM and BswM . . . 47
8.3.1.1 Fw_Init . . . 47
8.3.2 Functions provided for BswM . . . 47
8.3.2.1 Fw_SetFirewallState . . . 48
8.3.3 Functions provided for LSduR . . . 48
8.3.3.1 Fw_RxIndication . . . 48
8.3.4 Functions provided for EthIf . . . 49
8.3.4.1 Fw_StreamStatisticsIndication . . . 49
8.3.4.2 Fw_StreamStateIndication . . . 50
8.4 Scheduled functions . . . 50
8.4.1 MainFunction . . . 50
8.5 Expected interfaces . . . 51
8.5.1 Mandatory interfaces . . . 51
8.5.2 Optional interfaces . . . 51
8.6 Service Interfaces . . . 52
9 Sequence diagrams 53
9.1 Switch core filter rule extraction . . . 53
9.2 Switch core filter rule counter statistics . . . 53
9.3 Switch core filter rule management . . . 53
10 Configuration specification 54
10.1 How to read this chapter . . . 54
10.2 Containers and configuration parameters . . . 54
10.2.1 FirewallGeneral . . . 55
10.2.2 Firewall Pdu Routing . . . 58
10.2.3 Connection to BswM . . . 62
10.2.4 Filter Rules . . . 66
10.2.4.1 Data link layer configuration . . . 71
10.2.4.2 IPv4 configuration . . . 76
10.2.4.3 IPv6 configuration . . . 84
10.2.4.4 ICMP configuration . . . 87
10.2.4.5 TCP configuration . . . 89
10.2.4.6 UDP configuration . . . 94

10.2.4.7 SOME/IP configuration . . . 96
10.2.4.8 SOME/IP-SD configuration . . . 104
10.2.4.9 DDS configuration . . . 107
10.2.4.10 DoIP configuration . . . 113
10.2.4.11 Payload Byte Pattern configuration . . . 119
10.2.5 Switch bucket counting mechanism . . . 122
10.2.6 Security Events . . . 125
10.3 Published Information . . . 133
A Not applicable requirements 134
B Change history of AUTOSAR traceable items 135
B.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 135
B.1.1 Added Specification Items in R26-11 . . . 135
B.1.2 Changed Specification Items in R26-11 . . . 135
B.1.3 Deleted Specification Items in R26-11 . . . 135
B.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 135
B.2.1 Added Specification Items in R25-11 . . . 135
B.2.2 Changed Specification Items in R25-11 . . . 135
B.2.3 Deleted Specification Items in R25-11 . . . 135
B.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 136
B.3.1 Added Specification Items in R24-11 . . . 136
B.3.2 Changed Specification Items in R24-11 . . . 136
B.3.3 Deleted Specification Items in R24-11 . . . 136
B.4 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 136
B.4.1 Added Specification Items in R23-11 . . . 136
B.4.2 Changed Specification Items in R23-11 . . . 137
B.4.3 Deleted Specification Items in R23-11 . . . 137

# 1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the AUTOSAR Basic Software module Firewall.

The Firewall filters network traffic based on pre-defined firewall rules to protect the host from malicious messages. To this end, the Firewall supports stateless packet inspection, stateful packet inspection and deep packet inspection. Additionally, the Firewall offers interfaces to adapt the Firewall rule configuration during runtime, e.g. to adapt for different vehicle states or to support Intrusion Prevention Systems.

The Firewall also supports deployment scenarios directly on the switch: A Classic AUTOSAR stack can be deployed on a smart switch (containing a dedicated CPU), where the Firewall module can filter messages on network level rather than host level. The Firewall supports interfaces to the switch core to leverage hardware-accelerated network packet filtering on the switch core, e.g., by the means of TCAM rules.

# 2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Firewall module that are not included in the [1, AUTOSAR glossary].

## 2.1 Acronyms

| Acronym: | Description: |
|---|---|
| Firewall | An automotive Ethernet firewall is a network security device that monitors incoming and outgoing network traffic and grants or rejects network access between two or more Electronic Control Units (ECU) or between network zones (e.g. vehicle domain (ADAS, infotainment, diagnostics etc), trusted/non-trusted zones). |
| Firewall Rule | Pattern of expected values for a network packet together with an associated action in case a network packet matches the pattern (e.g., block or allow the network packet). |
| Firewall State | The Firewall State reflects the current state of the vehicle (e.g. driving, in a diagnostic session, ...) and can be set by a user application. Based on the currently active Firewall State, a specific set of `Firewall Rule`s matching the current vehicle state is active. |
| Allowlist | Collection of Firewall Rules where the network packet is allowed in case of a pattern match. |
| Blocklist | Collection of Firewall Rules where the network packet is blocked in case of a pattern match. |
| OSI Layer | Network layer according to the ISO OSI model as specified in ISO/IEC 7498. |

**Table 2.1: Acronyms used in the scope of this Document**

## 2.2 Abbreviations

| Abbreviation: | Description: |
|---|---|
| BswM | Basic Software Mode Manager |
| DDS | Data Distribution Service |
| DDS-RTPS | DDS Real-Time Publish Subscribe Protocol |
| DoIP | Diagnostics over IP |
| EthIf | Ethernet Interface |
| IDS | Intrusion Detection System |
| IdsM | IDS Manager |
| IdsR | IDS Reporter |
| IP | Internet Protocol |
| SEv | Security Event |
| SOME/IP | Service oriented Middleware over IP |
| TCAM | Ternary content-addressable memory |
| TCP | Transmission control protocol |
| UCM | Update & Configuration Management |
| UDP | User datagram protocol |

**Table 2.2: Abbreviations used in the scope of this Document**

# 3 Related documentation

This document provides the software specification for the Firewall module. The following document complement this specification:

- **CP_TPS_SystemTemplate** [2]: System-level description of the Firewall configuration.

## 3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_FO_TR_Glossary

[2] System Template
AUTOSAR_CP_TPS_SystemTemplate

[3] General Specification of Basic Software Modules
AUTOSAR_CP_SWS_BSWGeneral

[4] General Requirements on Basic Software Modules
AUTOSAR_CP_RS_BSWGeneral

[5] IEEE Standard for Ethernet
https://ieeexplore.ieee.org/document/7428776

[6] SOME/IP Protocol Specification
AUTOSAR_FO_PRS_SOMEIPProtocol

[7] SOME/IP Service Discovery Protocol Specification
AUTOSAR_FO_PRS_SOMEIPServiceDiscoveryProtocol

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

[9] ISO 13400-2:2019 – Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 2: Network and transport layer requirements and services (Edition 2, Release 2019-12)
https://www.iso.org/standard/74785.html

[10] Specification of Ethernet Switch Driver
AUTOSAR_CP_SWS_EthernetSwitchDriver

[11] Security Extract Template
AUTOSAR_FO_TPS_SecurityExtractTemplate

## 3.2 Related specification

AUTOSAR provides a General Specification on Basic Software modules [3, SWS BSW General], which is also valid for the Firewall.

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

# 4 Constraints and assumptions

## 4.1 Known Limitations

The firewall supports only filtering of ingress traffic.

## 4.2 Applicability to car domains

No limitation with regards to applicability to specific car domains.

# 5 Dependencies to other modules

The Firewall has connections to the following modules:

- **LSduR**: The Firewall receives network packets from the LSduR for inspection and passes them back to the LSduR if the network packet is allowed to continue in the network stack.
- **EthIf**: The Firewall uses the EthIf module to communicate with Ethernet Switch Drivers in the case of a deployment on a switch.
- **IdsM**: The Firewall module raises Security Events to the IdsM in the case of blocked network packets.
- **BswM**: The BswM manages the state of the Firewall (see Chapter 7.4.3 for more details)

# 6 Requirements Tracing

The following tables reference the requirements specified in [4] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

| Requirement | Description | Satisfied by |
|---|---|---|
| **[SRS_BSW_00337]** | Classification of development errors | [CP_SWS_Fw_91000] |

**Table 6.1: Requirements Tracing**

# 7 Functional specification

## 7.1 Overview

The AUTOSAR basic software module Firewall serves as an additional security layer that inspects network traffic and filters it based on a given rule set. The general behavior of a Firewall can be described as follows: The Firewall manages a list of expected network packet patterns, where each pattern is associated with a respective action (e.g. allow or block the network packet). The combination of network packet pattern and action is called a `FirewallRule`. For every network packet that passes the network stack (ingress and egress), the Firewall compares the network packet against the list of patterns. In case of a pattern match, the Firewall carries out the action associated with the pattern. If no pattern matches (no-match case), the Firewall carries out a default action.

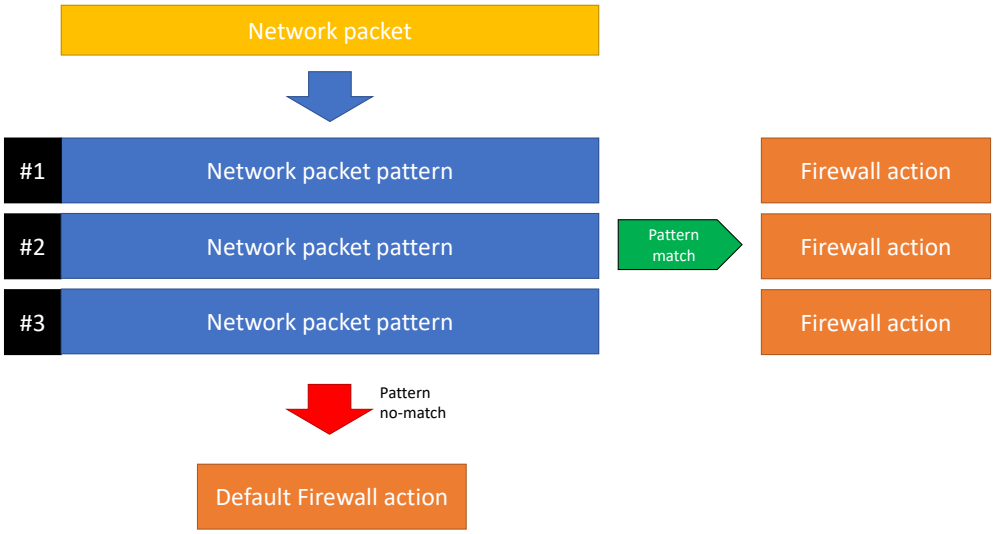



**Figure 7.1: Pattern matching mechanism**

The Firewall interfaces with the LSduR to receive network packets. After inspection, the Firewall returns the network packet to LSduR if it is allowed to continue in the network stack or otherwise dropped by the Firewall. The `FirewallRule`s are generally static, but the Firewall offers a mechanism to dynamically enable/disable `FirewallRule`s during runtime: The Firewall is connected to the `BswM`, which switches the `FirewallState` to allow for dynamic firewall behavior based on the current vehicle state (e.g. driving, parking, in a diagnostic session). More details can be found in Section 7.4.3. Furthermore, the Firewall supports also the intrusion detection system by raising security events. The integration of the Firewall into the AUTOSAR stack can hence be represented as follows:

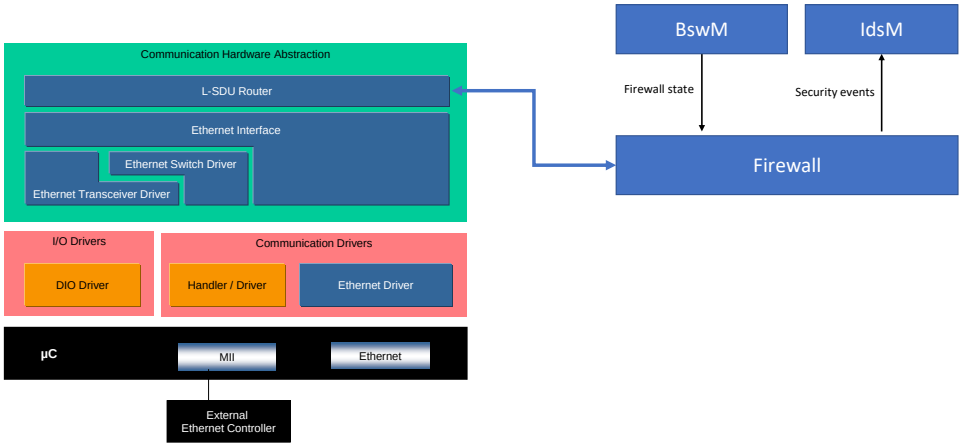



**Figure 7.2: Integration of the Firewall module into the AUTOSAR stack**

The Firewall also supports scenarios where an AUTOSAR stack is directly deployed on a smart switch (containing a dedicated programmable CPU). In this case, the Firewall can be employed to filter network traffic directly on the switch, thus protecting directly the in-vehicle network. This deployment scenario and its implications is described in detail in Section 7.5.

This chapter is structured as follows:

- Sec. 7.2 describes the handling of the Firewall module
- Sec. 7.3 describes the network packet inspection, i.e. the pattern-matching part of the `FirewallRule`s
- Sec. 7.4 describes the filtering aspect if the Firewall, i.e. which actions to carry out in case of a pattern match. This section also contains the use-cases of rate limiting and filtering based on the vehicle state
- Sec. 7.5 describes the deployment scenario of a Firewall on a smart switch
- Sec. 7.6 describes the security events raised by the Firewall
- Sec. 7.7 describes errors raised by the Firewall

## 7.2 Module handling

### 7.2.1 Initialization

The Firewall module is initialized via Fw_Init. Except for Fw_Init, the API functions of the Firewall module may only be called after the module has been properly initialized.

The Firewall follows the specification from the CP_SWS_BSWGeneral [3] with regards to the module initialization, especially SWS_BSW_00071, SWS_BSW_00243 and SWS_BSW_00231.

### 7.2.2 Timing Related Functionality

To be able to handle asynchronous calls correctly, the Firewall module is triggered cyclically via the Fw_MainFunction.

## 7.3 Network packet inspection

The Firewall manages a list of firewall rules, which consist of an expected network packet pattern and actions to be carried out in case of a pattern match. The firewall rules are modeled as `FirewallRule`s in the AUTOSAR methodology. For every network packet that passes the network stack, the firewall compares the network packet with all configured expected patterns and carries out the action associated with the `FirewallRule` in case of a pattern match. The `FirewallRule`s are ordered based on the Metamodel configuration and the firewall shall iterate through the `FirewallRule`s in the configured order until the first pattern match.

**[CP_SWS_Fw_30027] Pattern matching algorithm**

*Status:* DRAFT

⌈Upon invocation of Fw_RxIndication, the firewall shall inspect the received network packet and compare it against the ordered list of expected patterns defined in `FirewallRule`s. In case of a pattern match, the firewall stops with the comparison against the expected patterns and carries out the action associated with the matching rule.⌋

The possible actions in case of a pattern match are described in Sec. 7.4.

The firewall supports different filtering mechanisms:

- **Stateless filtering:** Inspection of field values (e.g. header fields) and comparison against statically defined values
- **Stateful filtering:** Filtering on specific aspects of the stateful nature of the underlying protocol (e.g. allowed state transitions, number of open connections)
- **Deep packet inspection:** Inspection of application layer protocols (e.g. SOME/IP, DDS, DoIP). This can also include generic inspection of the network packet payload based on offset and expected value

The firewall performs the inspection on the complete network packet. Hence, the pattern description is comprised of expected patterns for different protocols. This is modeled by individual configuration parts for every OSI Layer (`FirewallDataLinkFilterConfig`, `FirewallNetworkLayerFilterConfig`, `Firewall-`

TransportLayerFilterConfig etc.) that are aggregated by FirewallRules in the AUTOSAR Metamodel.

**[CP_SWS_Fw_30002]**

*Status:* DRAFT

⌈A FirewallRule is considered a match if all aggregated FirewallDataLinkFilterConfigs, FirewallNetworkLayerFilterConfigs, FirewallTransportLayerFilterConfigs, FirewallSomeipProtocolFilterConfigs, FirewallSomeipSdFilterConfigs, FirewallDdsFilterConfigs, FirewallDoipFilterConfigs and FirewallPayloadBytePatternFilterConfigs generate a match for their respective protocol.⌋

### 7.3.1 Stateless packet inspection

For stateless packet inspection, the Firewall inspects the network protocol headers up to OSI layer 4 and compares them against expected values.

**[CP_SWS_Fw_30003]**

*Status:* DRAFT

⌈The Firewall shall compare the expected values defined in FirewallDataLinkFilterConfig of every FirewallRule against the header fields in the network packet. If all values match, the FirewallDataLinkFilterConfig is considered a match. Otherwise the FirewallDataLinkFilterConfig is considered a no-match.⌋

**[CP_SWS_Fw_30004]**

*Status:* DRAFT

⌈The Firewall shall compare the expected values defined in FirewallNetworkLayerFilterConfig of every FirewallRule against the header fields in the network packet. If all values match, the FirewallNetworkLayerFilterConfig is considered a match. Otherwise the FirewallNetworkLayerFilterConfig is considered a no-match.⌋

**[CP_SWS_Fw_30005]**

*Status:* DRAFT

⌈The Firewall shall compare the expected values defined in FirewallTransportLayerFilterConfig of every FirewallRule against the header fields in the network packet. If all values match, the FirewallTransportLayerFilterConfig is considered a match. Otherwise the FirewallTransportLayerFilterConfig is considered a no-match.⌋

The Firewall shall only inspect the parameters that were configured within a FirewallRule. Parameters that are available within the Metamodel but are not configured shall be ignored.

In some cases, it is useful to not limit the expected pattern to specific values, but to also allow for values to be in a specific range. Ranges can either be defined by subnets (e.g., for MAC and IP addresses) or by defining the minimal and maximal value of the parameter (e.g., for ports).

**[CP_SWS_Fw_30006]**

*Status:* DRAFT

⌈If a FirewallDataLinkFilterConfig defines a subnet by means of FirewallFilterMACSrcAddress.FirewallFilterMACAddressMask or FirewallFilterMACDestAddress.FirewallFilterMACAddressMask , all addresses within the network packet that fall within this subnet are considered a match for this FirewallDataLinkFilterConfig.⌋

**[CP_SWS_Fw_30007]**

*Status:* DRAFT

⌈If an FirewallNetworkLayerIpv4FilterConfig defines a subnet by means of FirewallFilterIPSrcAddress.FirewallFilterIPAddressMask or FirewallFilterIPDestAddress.FirewallFilterIPAddressMask, all addresses within the network packet that fall within this subnet are considered a match for this FirewallNetworkLayerIpv4FilterConfig.⌋

**[CP_SWS_Fw_30008]**

*Status:* DRAFT

⌈If an FirewallNetworkLayerIpv6FilterConfig defines a subnet by means of FirewallFilterIPSrcAddress.FirewallFilterIPAddressMask or FirewallFilterIPDestAddress.FirewallFilterIPAddressMask, all addresses within the network packet that fall within this subnet are considered a match for this FirewallNetworkLayerIpv6FilterConfig.⌋

**[CP_SWS_Fw_30009]**

*Status:* DRAFT

⌈If an FirewallNetworkLayerIpv4FilterConfig defines a range by means of FirewallFilterIPv4Ttl.FirewallIpv4TtlMin and FirewallFilterIPv4Ttl.FirewallIpv4TtlMax, all values within the network packet that fall within this range (including the minimal and maximal value) are considered a match for this FirewallNetworkLayerIpv4FilterConfig.⌋

**[CP_SWS_Fw_30010]**

*Status:* DRAFT

⌈If a FirewallTransportLayerFilterConfig defines a range by means of FirewallFilterSrcPort.FirewallFilterPortLowerValue and Firewall-

FilterSrcPort.FirewallFilterPortUpperValue or by means of FirewallFilterDestPort.FirewallFilterPortLowerValue and FirewallFilterDestPort.FirewallFilterPortLowerValue, all values within the network packet that fall within this range (including the minimal and maximal value) are considered a match for this FirewallTransportLayerFilterConfig.⌋

The Firewall shall also be able to verify if the checksum of the respective protocol is valid.

**[CP_SWS_Fw_30011]**

*Status:* DRAFT

⌈If FirewallNetworkLayerIpv4FilterConfig.FirewallChecksumVerification , FirewallNetworkLayerIcmpConfig.FirewallChecksumVerification or FirewallTransportLayerTcpFilterConfig.FirewallChecksumVerification is set to true, the Firewall shall check if the checksum field for the respective protocol is available in the network packet. If the checksum is available, the respective FirewallNetworkLayerIpv4FilterConfig, FirewallNetworkLayerIcmpConfig or FirewallTransportLayerFilterConfig is considered a match.⌋

#### 7.3.1.1 Inspection of not modeled protocols

For stateless packet inspection, the Firewall natively supports the modeled protocols Ethernet, IPv4, IPv6, ICMP, TCP and UDP. Additional protocols can be added by two mechanisms:

**EtherType inspection:** Many protocols can already be identified on data link layer by means of the EtherType (as defined in IEEE 802.3 [5]). These protocols can therefore be blocked by the Firewall by configuring FirewallFilterEtherType within a FirewallRule. Examples for protocols that can be identified based on EtherTypes can be found in Table 7.1.

| *EtherType* | *Protocol* |
|---|---|
| 0x0806 | Address Resolution protocol over IPv4 (ARP) |
| 0x22EA | Stream Reservation Protocol (SRP) |
| 0x22F0 | Audio Video Transport Protocol (AVTP) |
| 0x88F7 | Precision Time Protocol (PTP) over IEEE 802.3 Ethernet |
| 0xF1C1 | Redundancy Tag (as defined in IEEE 802.1CB Frame Replication and Elimination for Reliability) |

**Table 7.1: EtherType examples**

**Generic inspection based on byte pattern:** The Firewall supports generic inspection of network packets based on expected byte-values at given offsets. This feature is

specified in Sec. 7.3.3.4 and allows for detailed inspection of protocols that are not modeled within the Firewall as well as inspection of payload data.

### 7.3.2 Stateful packet inspection

In stateful packet inspection, the FC Firewall takes into account the stateful nature of TCP and performs additional checks to identify timeouts, limit the number of open connections and perform checks against the TCP state machine.

**[CP_SWS_Fw_30012]**

*Status:* DRAFT

⌈If the parameter `FirewallTimeoutCheck` is set, the Firewall shall store the time of the latest network packet for the respective communication peer. If the time between the latest and current network packet is smaller than the value of `FirewallTimeoutCheck`, the `FirewallTransportLayerTcpFilterConfig` is considered a match.⌋

**[CP_SWS_Fw_30013]**

*Status:* DRAFT

⌈If the parameter `FirewallNumberOfParallelTcpSessions` is set, the Firewall shall keep track of the number of open TCP connections. If a network packet wants to open a new TCP session and the number of open TCP sessions including the newly opened TCP session is smaller than `FirewallNumberOfParallelTcpSessions`, the `FirewallTransportLayerTcpFilterConfig` is considered a match.⌋

**[CP_SWS_Fw_30014]**

*Status:* DRAFT

⌈If the parameter `FirewallStateManagementBasedOnTcpFlags` is set to true, the Firewall shall check whether the network packet wants to perform an allowed TCP state transition according to RFC 793. If this state transition is allowed, the `FirewallTransportLayerTcpFilterConfig` is considered a match.⌋

### 7.3.3 Deep packet inspection

The Firewall supports also inspection of application layer protocols to perform deep packet inspection of network packets. To this end, the Firewall supports deep packet inspection of the following protocols:

- `SOME/IP` (including SOME/IP-SD)
- `DDS`
- `DoIP`
- Generic deep packet inspection

#### 7.3.3.1 SOME/IP

For SOME/IP [6] the inspection focuses on the SOME/IP header fields. The header fields also include service-specific information like Service ID, Method ID etc., so the deep packet inspection of SOME/IP packets can be used to perform access control to individual services.

It is possible that multiple SOME/IP messages are transported within one TCP or UDP frame. Within the Firewall metamodel, every FirewallRule can aggregate at most one SOME/IP message. If a network packet contains more than one SOME/IP message, the Firewall has thus to check that for every SOME/IP message within the network packet a valid FirewallRule exists.

**[CP_SWS_Fw_30015]**

*Status:* DRAFT

⌈If the network packet to be inspected contains one or multiple SOME/IP messages, the Firewall shall find the subset of FirewallRules, where the respective FirewallDataLinkFilterConfig, FirewallNetworkLayerFilterConfig and FirewallTransportLayerFilterConfig have provided a match and a FirewallSomeipProtocolFilterConfig is aggregated.⌋

**[CP_SWS_Fw_30016]**

*Status:* DRAFT

⌈For this subset, the Firewall shall compare their expected values against the SOME/IP header fields of the SOME/IP messages in the network packet. If all values match and if for all FirewallRules the FirewallAction from the referenced FirewallActionForMatchingRules is the same, the respective FirewallRules are considered to be matches.⌋

Additionally, the Firewall supports length verification, i.e. to check whether the TCP/UDP payload length matches the combined length of all included SOME/IP messages

**[CP_SWS_Fw_30017]**

*Status:* DRAFT

⌈If the parameter FirewallSomeipLengthVerification is set to true, the Firewall shall compare the TCP/UDP payload size with the cumulative length of all included SOME/IP messages. If both values match, the FirewallSomeipProtocolFilterConfig is considered a match. Otherwise the FirewallSomeipProtocolFilterConfig is considered a no-match.⌋

The Firewall also supports inspection of the SOME/IP service discovery protocol [7]. Similar to regular SOME/IP inspection, it is also possible to group multiple SOME/IP-SD messages within one network packet. Hence, the Firewall implements a similar logic to inspect network packets with multiple SOME/IP-SD messages.

**[CP_SWS_Fw_30018]**

*Status:* DRAFT

⌈If the network packet to be inspected contains one or multiple SOME/IP-SD messages, the Firewall shall find the subset of FirewallRules, where the respective FirewallDataLinkFilterConfig, FirewallNetworkLayerFilterConfig and FirewallTransportLayerFilterConfig have provided a match and a FirewallSomeipSdFilterConfig is aggregated.⌋

**[CP_SWS_Fw_30019]**

*Status:* DRAFT

⌈For this subset, the Firewall shall compare their expected values against the SOME/IP-SD header fields of the SOME/IP-SD messages in the network packet. If all values match and if for all FirewallRules the FirewallAction from the referenced FirewallActionForMatchingRules is the same, the respective FirewallRules are considered to be matches.⌋

**[CP_SWS_Fw_30020]**

*Status:* DRAFT

⌈If a FirewallSomeipSdFilterConfig is aggregated in a FirewallRule, the Firewall shall compare the SOME/IP header fields of all SOME/IP-SD messages within the network packet against the default values defined in PRS_SOMEIPServiceDiscoveryProtocol [7]. If all values match, the FirewallSomeipSdFilterConfig is considered a match. Otherwise the FirewallSomeipSdFilterConfig is considered a no-match⌋

Similar to the stateless network packet inspection on lower layers, it is also possible to define ranges of allowed values by using minimal and maximal values. In case such a range is defined, all values from the network packet that fall within this range are a match.

**[CP_SWS_Fw_30021]**

*Status:* DRAFT

⌈If a FirewallSomeipSdFilterConfig defines a range by means of FirewallSomeipMinorVersion.FirewallMinorVersionMinValue and FirewallSomeipMinorVersion.FirewallMinorVersionMaxValue or by means of FirewallSomeipMajorVersion.FirewallMajorVersionMinValue and FirewallSomeipMajorVersion.FirewallMajorVersionMaxValue, all values within the network packet that fall within this range (including the minimal and maximal value) are considered a match for this FirewallSomeipSdFilterConfig.⌋

Note that the Firewall is only able to allow and block complete network packets. If multiple SOME/IP messages are transported within one TCP/UDP frame and only one SOME/IP message shall be blocked by the Firewall, the Firewall will nonetheless block the complete network packet including the other SOME/IP messages. The same behavior holds true for SOME/IP-SD, where multiple service discovery messages can

be contained within one TCP/UDP frame and the firewall will either allow or block the complete network packet.

#### 7.3.3.2 DDS

Deep packet inspection of DDS messages is based on the DDS Interoperability Wire Protocol (DDS-RTPS [8]), which specifies the representation of DDS messages within network packets: DDS-RTPS defines a packet format that consists of a RTPS header and multiple RTPS submessages that can be accumulated within one RTPS message. Additionally, DDS allows also for multiple RTPS messages within one TCP or UDP packet. In analogy to SOME/IP, the Firewall allows only the configuration of a single RTPS header and submessage within a FirewallRule and the Firewall has hence to compare the network packet against all configured RTPS rules.

**[CP_SWS_Fw_30022]**

*Status:* DRAFT

⌈If the network packet to be inspected contains one or multiple DDSI-RTPS messages, the Firewall shall find the subset of FirewallRules, where the respective FirewallDataLinkFilterConfig, FirewallNetworkLayerFilterConfig and FirewallTransportLayerFilterConfig have provided a match and a FirewallDdsFilterConfig is aggregated.⌋

**[CP_SWS_Fw_30023]**

*Status:* DRAFT

⌈For this subset, the Firewall shall compare their expected values against the fields of the DDS-RTPS messages and submessages in the network packet. If all values match and if for all FirewallRules the FirewallAction from the referenced FirewallActionForMatchingRules is the same, the respective FirewallRules are considered to be matches.⌋

#### 7.3.3.3 DoIP

The Firewall supports deep packet inspection of DoIP messages [9], where the firewall inspects the DoIP header as well as parts of the payload (DoIP source/destination address, UDS services). The Firewall does not, however, perform deep packet inspection of the UDS protocol, i.e., inspection on the level of individual DIDs, RIDs etc. Nevertheless, these kind of checks are still possible to implement by means of the generic inspection feature described in Sec. 7.3.3.4.

**[CP_SWS_Fw_30024]**

*Status:* DRAFT

⌈The Firewall shall compare the expected values defined in FirewallDoipFilterConfig of every FirewallRule against the DoIP header fields in the network packet.

If all values match, the FirewallDoipFilterConfig is considered a match. Otherwise the FirewallDoipFilterConfig is considered a no-match.⌋

Similar to the stateless network packet inspection on lower layers, it is also possible to define ranges of allowed values by using minimal and maximal values. In case such a range is defined, all values from the network packet that fall within this range are a match.

**[CP_SWS_Fw_30025]**

*Status:* DRAFT

⌈If a FirewallDoipFilterConfig defines a range by means of FirewallDoipSrcAddress.FirewallDoipSrcAddressLowerValue and FirewallDoipSrcAddress.FirewallDoipSrcAddressUpperValue or by means of FirewallDoipDestAddress.FirewallDoipDestAddressLowerValue and FirewallDoipDestAddress.FirewallDoipDestAddressUpperValue, all values within the network packet that fall within this range (including the minimal and maximal value) are considered a match for this FirewallDoipFilterConfig.⌋

#### 7.3.3.4 Generic inspection

The Firewall allows for generic inspection of the network packets (e.g. to perform payload inspection or to inspect protocols that are not natively supported by the firewall). To this end, every FirewallRule can aggregate multiple FirewallPayloadBytePatternFilterConfigs, which specify the expected byte values at a specific offset within the network packet.

**[CP_SWS_Fw_30026]**

*Status:* DRAFT

⌈The Firewall shall compare the expected values defined in the FirewallPayloadBytePatternFilterConfigs of every FirewallRule against the values at the specified offsets in the network packet. If all values match, the FirewallPayloadBytePatternFilterConfigs are considered matches.⌋

## 7.4 Network packet filtering

After describing the rule-based network packet inspection process based on pattern-matching in chapter 7.3, this chapter specifies the associated filtering mechanisms supported by the Firewall. Section 7.4.1 describes the pattern-matching-based filtering approach using Allowlists and Blocklists, Section 7.4.2 specifies the rate limiting feature of the Firewall and Section 7.4.3 outlines the state-dependent filtering mechanism based on configurable Firewall States.

**[CP_SWS_Fw_40100] Allowed network packets**

*Status:* DRAFT

⌈If a network packet shall be allowed to continue in the network stack, the Firewall shall call LsduR_FwRxIndication with the same parameters used when receiving the network packet via Fw_RxIndication.⌋

**[CP_SWS_Fw_40101] Blocked network packets**

*Status:* DRAFT

⌈If a network packet shall be blocked from continuing in the network stack, the Firewall shall drop the network packet. If Pdu.KeepLocalBuffer is set to True, the Firewall shall additionally call LSduR_FwRxReleaseBuffer to release the buffer.⌋

### 7.4.1 Allowlists and Blocklists

Firewalls can generally be categorized into two groups: `Allowlist` and `Blocklist` firewalls. In an `Allowlist` firewall, all network traffic that is allowed to pass the firewall is specified (i.e. patterns are defined), all network packets without a matching pattern are blocked. `Blocklist` firewalls implement the inverse approach: Only explicitly defined network packets are blocked, whereas traffic without a matching pattern is allowed to pass the firewall.

The action to be carried out in the case of a match of a `FirewallRule` is defined by the parameter `FirewallAction` in the referenced `FirewallActionForMatchingRules`.

**[CP_SWS_Fw_40102] Allow condition for a network packet**

*Status:* DRAFT

⌈If a `FirewallRule` is a match and `FirewallAction` in the referenced `FirewallActionForMatchingRules` is set to allow, the Firewall shall allow the network packet as defined in [CP_SWS_Fw_40100].⌋

**[CP_SWS_Fw_40103] Block condition for a network packet**

*Status:* DRAFT

⌈If a `FirewallRule` is a match and `FirewallAction` in the referenced `FirewallActionForMatchingRules` is set to block, the Firewall shall block the network packet as defined in [CP_SWS_Fw_40101].⌋

In addition, it has to be defined how the Firewall shall behave in the case that no `FirewallRule` generated a match:

**[CP_SWS_Fw_40104] Default block condition for a network packet**

*Status:* DRAFT

⌈If no `FirewallRule` matches the network packet and `FirewallDefaultAction` is set to block, the Firewall shall block the network packet as defined in [CP_SWS_Fw_40101].⌋

**[CP_SWS_Fw_40106] Default allow condition for a network packet**

*Status:* DRAFT

⌈If no `FirewallRule` matches the network packet and `FirewallDefaultAction` is set to allow, the Firewall shall allow the network packet as defined in [CP_SWS_Fw_40100].⌋

The Firewall allows also for mixed Allow-/Blocklist Firewalls: it is possible to define `FirewallRule`s that block a network packet upon a pattern match together with `FirewallRule`s that allow a network packet to pass upon a pattern match. This seems redundant at first, since network packets that provide no match are caught by the Firewalls default behavior, but there is one specific reason for this design: The explicit definition of network packet patterns allows for the usage of the pattern matching algorithm, which in turn allows for a dedicated mapping of IDS security events for these network packets. See Sec. 7.6 for more details.

### 7.4.2 Rate limiting

The Firewall supports rate limiting based on the pattern matching algorithm to identify off-frequency cyclic messages, that can be caused by, e.g., a man-in-the-middle attack or a faulty ECU. To realize this, the Firewall implements the leaky bucket algorithm, which is also supported on HW side by some products.

**[CP_SWS_Fw_40004]**

*Status:* DRAFT

⌈If the parameters `FirewallBucketSize` and `FirewallRefillAmount` are configured for a `FirewallRule`, the Firewall shall keep track of the number of pattern matches by means of a leaky bucket algorithm, where `FirewallRefillAmount` defines the decrement rate of the leaky bucket algorithm and the counter is increased by one for every pattern match.⌋

**[CP_SWS_Fw_40105] Overflowing leaky bucket behaviour**

*Status:* DRAFT

⌈In the case of a pattern match and if the leaky bucket counter is bigger than `FirewallBucketSize`, the Firewall shall block the network packet as defined in [CP_SWS_Fw_40101].⌋

**[CP_SWS_Fw_40012]**

*Status:* DRAFT

⌈The firewall shall keep the current leaky bucket counter also when the firewall state is switched according to [CP_SWS_Fw_40009].⌋

### 7.4.3 State dependent filtering

The in-vehicle traffic can strongly depend on the vehicle's situation (e.g. driving, parking, in a diagnostic session etc.), which also renders the expected network packets to be different depending on the current vehicle state. The Firewall supports this use-case by being state-dependent: `FirewallRule`s can be associated with specific `Firewall State`s, that are pre-configured on a project-specific basis by the integrator and that can be managed by a user application. Within the AUTOSAR Meta Model, this feature is realized by `FirewallStateDependentRules`s that aggregate a set of `FirewallRule`s. Only one of the `FirewallStateDependentRules`s can be active, which means that only the `FirewallRule`s associated with that `FirewallStateDependentRules` are active

**[CP_SWS_Fw_40007]**

*Status:* DRAFT

⌈Only the `FirewallRule`s referenced by the currently active `FirewallStateDependentRules` shall be taken into account for the network packet inspection. `FirewallRule`s that are not referenced by the currently active `FirewallStateDependentRules` shall be ignored.⌋

**[CP_SWS_Fw_40008]**

*Status:* DRAFT

⌈For no-match cases, the `FirewallDefaultAction` defined in the currently active `FirewallStateDependentRules` shall be used.⌋

The Firewall provides the Fw_SetFirewallState API to switch the currently active `FirewallStateDependentRules`. This API is called by the `BswM`.

**[CP_SWS_Fw_40009]**

*Status:* DRAFT

⌈If a `FirewallState` is reported to the Firewall by means of Fw_SetFirewallState, the `FirewallStateDependentRules` referenced by `FirewallStateRef` shall be considered as active.⌋

**[CP_SWS_Fw_40011]**

*Status:* DRAFT

⌈If no `FirewallState` has been reported to the Firewall, the Firewall shall consider the `FirewallStateDependentRules` as active where the referenced `FirewallState` is also referenced by the `FirewallInitialStateRef`.⌋

## 7.5 Firewall interaction with the switch

Firewalls are not limited to deployments on endpoints/Host-ECUs, but make also a lot of sense on central network entities like gateways and switches. Switches come typically with some basic firewall functionality, oftentimes based on TCAM rules. TCAM rules allow to perform stateless packet inspection, but cannot be used for stateful and deep packet inspection. The latter two inspection categories can be realized on a switch when it contains a dedicated CPU that runs a firewall in SW and that performs the inspection parts that cannot be realized by TCAM rules.

From a technical perspective, these so-called smart switches behave like any other uC: they have a CPU, memory and are connected to the switch core by an Ethernet connection (see also Fig. 7.3). This allows to also run an AUTOSAR stack directly on the switch. Switches are typically very resource-constrained devices, so it may allow to only run a stripped-down version of AUTOSAR, but this still allows to re-use the standardized AUTOSAR modules, the AUTOSAR tooling for configuration and the AUTOSAR firewall module to extend the basic firewall functionality based on TCAMs.

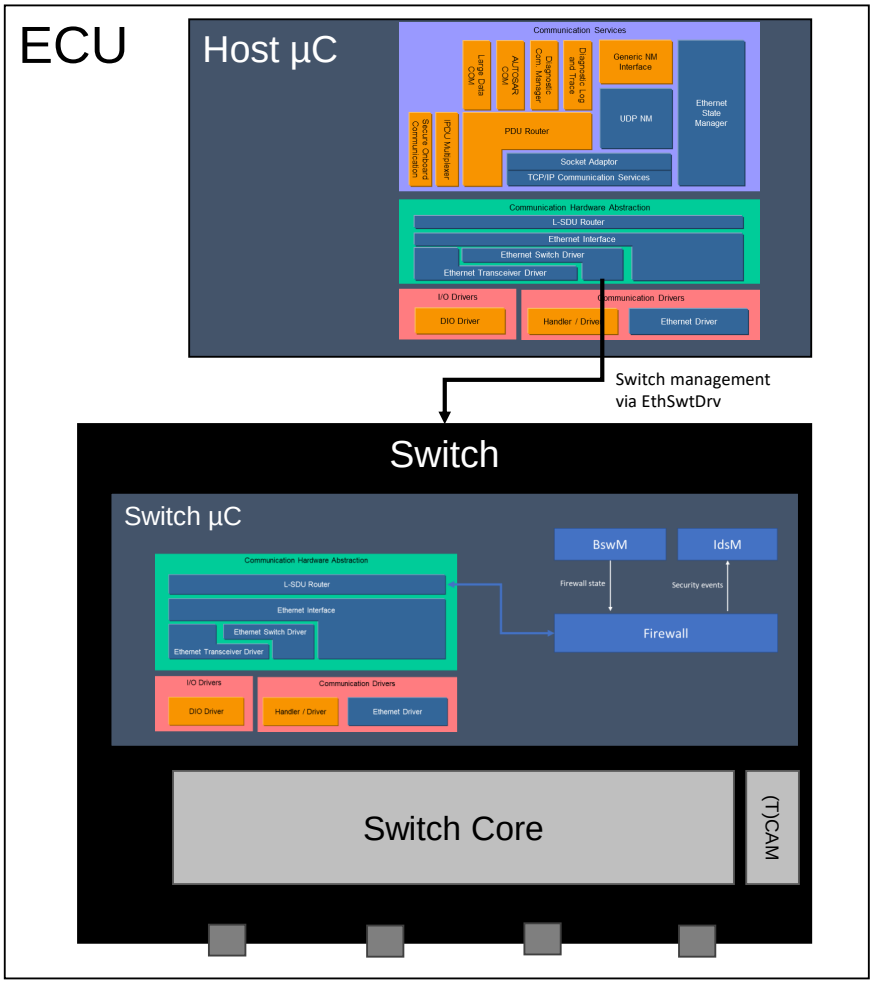



**Figure 7.3: Switch architecture incl. AUTOSAR deployment**

This chapter covers the aspects of the firewall module that were explicitly introduced for the deployment directly on the switch. The focus is on the interaction with the switch core and the firewall functionality available therein. The firewall rules in the switch core can be configured using the per-stream filtering functionality specified in the Ethernet Switch Driver [10]. Hence, the already available terminology is re-used and individual firewall rules on the switch core are addressed by their `EthSwtStreamIdentification` as defined in the AUTOSAR Ethernet switch driver [10].

The remainder of this chapter is structured in a way to address the individual filter scenarios separately:

- A network packet is received, completely inspected by the switch core and is allowed to pass. No additional inspection by the AUTOSAR firewall is required, hence the network packet is completely handled within the switch core. No additional functionality in the firewall module is required in this case.
- A network packet is received and partly inspected by the switch core. Additional inspection is needed by the firewall module and the network packet is hence passed to the switch CPU. This case is described in Sec. 7.5.1.

- A network packet is blocked already by the switch core. No additional inspection is required by the firewall module on the switch CPU, but a Security Event has to be raised. This case is described in Sec. 7.5.2.
- The BswM changes the state of the firewall, i.e., the active firewall rules are changed. This implies also a change in the firewall rules on the switch core. This case is described in Sec. 7.5.3.

### 7.5.1 Packet inspection by AUTOSAR firewall module

This section describes the case where a network packet was received by the switch and it passed inspection on the switch core, but additional inspection is needed by the firewall module on the switch CPU. The inspection by the firewall module can be performed quicker by taking into account that parts of the network packet were already inspected by the switch core. Hence, the firewall has to iterate only over all rules that are compatible with the filter rules associated with the `EthSwtStreamIdentification` that allowed the network packet to pass. To this end, the firewall module receives the index of the switch core filter rule via the parameter FIREWALL_RULE_ID_16 of the PDU Metadata.

**[CP_SWS_Fw_50010] Reduced set of firewall rules due to pre-filtering by switch core**

*Status:* DRAFT

⌈If `FirewallingWithPerStreamFiltering` is enabled, the firewall shall consider only the `FirewallRule`s for the inspection of the network packet that are compatible with the value stored at the FIREWALL_RULE_ID_16 of the PDU Metadata.⌋

Note that the mapping between the `FirewallRule`s managed by the firewall module and the filter rules for the switch core are not explicitly modeled. It is left open for the stack/tool vendors to provide support for this mapping to allow for a most efficient combination of filtering within the switch core and filtering on the switch CPU.

The content of the StreamHandleIdxPtr is generated in the following way: The switch core modifies the network packet header and inserts switch vendor specific metadata containing the filter rule identifier that allowed to pass the network packet. The network packet is received by the switch CPU via the Ethernet stack and passed to the Ethernet Switch Driver to parse the added metadata and extract the filter rule identifier, which is then stored in the PDU Metadata under FIREWALL_RULE_ID_16. This process is also shown in the sequence diagram in Sec. 9.1.

### 7.5.2 Network packets blocked by the switch core

This section describes the case where a network packet was received by the switch and blocked by the filter rules in the switch core. The network packet needs not to be

inspected by the firewall on the switch CPU, but the firewall shall raise a security event (SEv) for the blocked network packet.

To this end, the firewall supports two mechanisms to raise SEvs:

- Fine-grained SEvs indicating the network protocol for which a pattern mismatch was observed. This mechanism is outlined in detail in Section 7.6 and requires an inspection of the network packet by the firewall module.
- Reading out counting statistics from the switch core about applications of switch firewall rules on a regular basis and raise a SEv when counters are increased.

Both approaches have their advantages and disadvantages with respect to CPU load, memory consumption, detail of information etc. Both approaches are supported by the firewall and it is up to the project to decide which approach to follow. Both approaches are described in detail in the following sections.

#### 7.5.2.1 SEvs on protocol level

Section 7.6 specifies the mechanism for raising security events based on the inspection result. Since the SEvs are very fine-grained, the network packet needs to be inspected by the firewall on the switch CPU to identify the correct SEv to raise. Hence, even when the network packet should already be blocked by the switch core, it needs to be forwarded to the switch CPU for the firewall to inspect it.

However, the sole purpose of the network packet inspection by the firewall module is to identify the correct SEv to raise. It was already decided by the switch core that the network packet shall be dropped, which is typically done by the default firewall rule in the switch core. Hence, the firewall can directly block the network packet according to [CP_SWS_Fw_40101].

**[CP_SWS_Fw_50011] Default firewall rule in the switch core**

*Status:* DRAFT

⌈If Fw_RxIndication is invoked with FIREWALL_RULE_ID_16 containing the default firewall rule in the switch core, the Firewall shall block the network packet as defined in [CP_SWS_Fw_40101].⌋

#### 7.5.2.2 Switch firewall rule counting statistics SEv

Many switches support counting statistics for the firewall rules on the switch core. To this end, the rules are bundles within buckets that count the number of filter rule matches of all referenced rules combined (see Fig. 7.4). The buckets are configured on switch level and are not part of the Firewall specification.

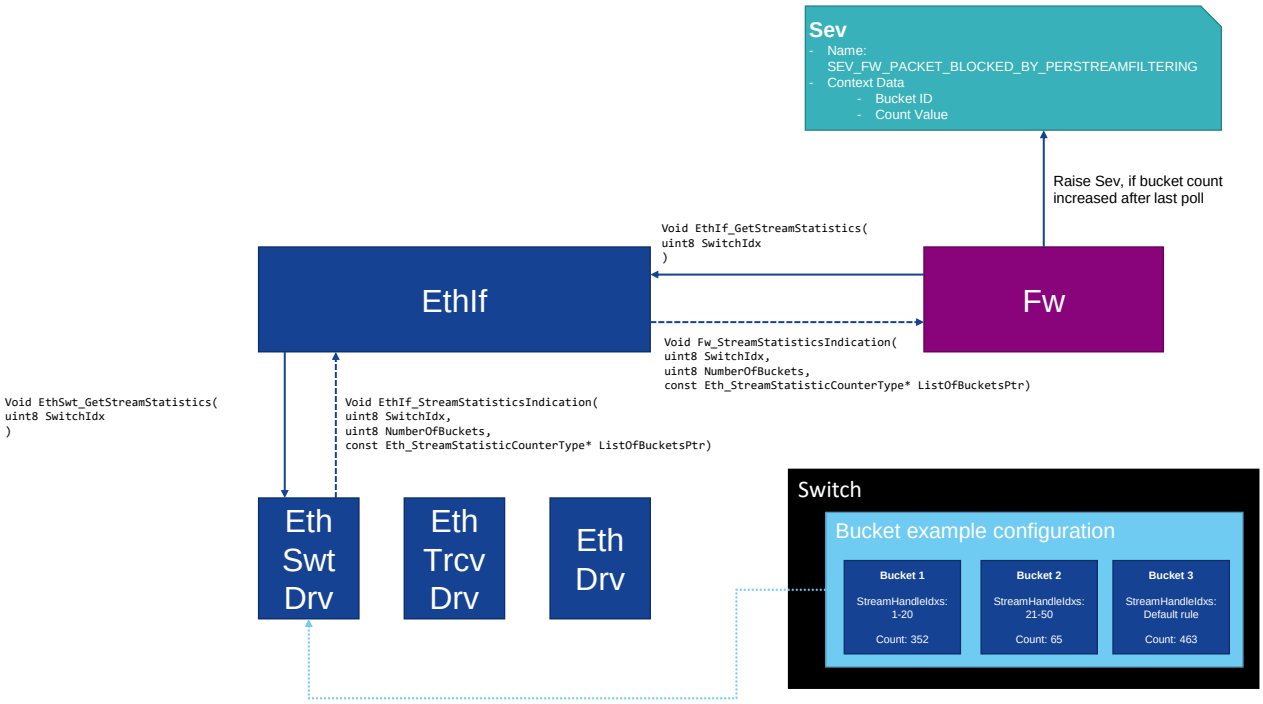



**Figure 7.4: Switch filter rule bucket counting mechanism**

The firewall can poll the count values of these buckets on a regular basis and raise a SEv if the count values have changed, i.e., if at least one network packet was blocked by the switch. The sequence diagram for this mechanism is shown in Sec. 9.2.

**[CP_SWS_Fw_50003]**

*Status:* DRAFT

⌈If the container `FirewallSwitchBucketCounterStatistics` is configured, the firewall shall invoke EthIf_GetStreamHandleIdxStatistics for all configured SwitchIdxs every `FwSwitchBucketCounterPollingInterval` seconds.⌋

**[CP_SWS_Fw_50004]**

*Status:* DRAFT

⌈When Fw_StreamStatisticsIndication is called, the firewall shall extract the bucket counter values from ListOfBucketsPtr. The firewall shall cache the count values for every SwitchIdx until updated count values are reported via Fw_StreamStatisticsIndication.⌋

**[CP_SWS_Fw_50005]**

*Status:* DRAFT

⌈The firewall shall compare the updated count values with the cached count values. If a count value has been increased, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING. If multiple count values have been increased, the firewall shall raise one SEv for each increased counter.⌋

**[CP_SWS_Fw_50006]**

*Status:* DRAFT

⌈If no cached counter values are available (e.g., because Fw_StreamStatisticsIndication is called the first time), the firewall shall assume a count value of zero.⌋

### 7.5.3 Management of firewall rules in the switch core

This section describes the case when the BswM changes the firewall state, so that the set of active firewall filter rules is changed. This has also an impact in the firewall rules in the switch core, which need to be synchronized with the firewall rules on the switch CPU to ensure correct network packet filtering. To this end, the firewall supports the (de-)activation of individual rules on the switch core during runtime. The firewall does not change the actual filter rule during runtime; it is assumed that the set of filter rules are statically configured and the firewall only switches them on/off. A sequence diagram showing the interaction with the switch core can be found in Sec. 9.3.

**[CP_SWS_Fw_50007]**

*Status:* DRAFT

⌈If the firewall state is switched by the BswM according to [CP_SWS_Fw_40009], the firewall module shall invoke EthIf_SetStreamState for all applicable SwitchIdx and StreamHandleIdx and set their activity status using StreamActivityStatus to the value required by the active firewall state.⌋

**[CP_SWS_Fw_50008]**

*Status:* DRAFT

⌈If no response via Fw_StreamStateIndication is received within `FirewallSwitchRuleMgmtTimeout` seconds or if the StreamHandleIdxActivityStatus does not match the expected value, the firewall shall retry setting the correct activity status following [CP_SWS_Fw_50007].⌋

**[CP_SWS_Fw_50009]**

*Status:* DRAFT

⌈In the case of [CP_SWS_Fw_50008] and if development error reporting is enabled (see FwDevErrorDetect), the firewall shall call Det_ReportError with the error code FW_E_SWITCHRULEMGMT_FAILED.⌋

## 7.6 Security Events

Firewalls are a crucial part of Intrusion Detection Systems (`IDS`), as they are monitoring the complete network traffic and are thus able to identify attacks within the in-vehicle

network. AUTOSAR specifies the vehicle part of an IDS within the IdsM (IDS Manager), which aggregates and qualifies security events raised by IDS sensors and forwards them to the configured sink, either the persistent memory or the vehicle-central IDS instance (IdsR in the AUTOSAR IDS concept).

The Firewall supports the IDS by acting as an IDS sensor and raising security events (SEvs) to the IdsM. To this end, the Firewall specifies a set of SEvs (see Sec. 7.6.1) as well as conditions on when to raise them (see Sec. 7.6.2).

### 7.6.1 SEvs raised by the firewall

The IdsM specifies SEvs to consist of a unique SEv ID and associated context data, that provides more details about the nature of the incident. The IdsM qualifies these SEvs by running them through a filter chain. During this process, the IdsM can also aggregate multiple SEvs with the same SEv IDs, where only the context data of one SEv is kept. This behavior can cause information loss and needs to be reflected when designing the SEvs raised by the Firewall - the SEvs need to be fine-grained enough to limit information loss as much as possible while still being precise and clear in their specification. To this end, the Firewall specifies a set of SEvs that is focusing on the individual protocols that are inspected by the Firewall:

**[CP_SWS_Fw_61000] Security events for firewall (CP)** ⌈

| *Name* | *Description* | *ID* |
|---|---|---|
| SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH | A network packet was blocked due to a rule mismatch on IPv4 layer. | 51 |
| SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH | A network packet was blocked due to a rule mismatch on IPv6 layer. | 52 |
| SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH | A network packet was blocked due to a rule mismatch within the ICMP protocol. | 53 |
| SEV_FW_PACKET_BLOCKED_TCP_MISMATCH | A network packet was blocked due to a rule mismatch on TCP layer. | 54 |
| SEV_FW_PACKET_BLOCKED_UDP_MISMATCH | A network packet was blocked due to a rule mismatch on UDP layer. | 55 |
| SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH | A network packet was blocked due to a rule mismatch in the SOME/IP protocol. | 56 |
| SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH | A network packet was blocked due to a rule mismatch in the SOME/IP SD protocol. | 57 |
| SEV_FW_PACKET_BLOCKED_DDS_MISMATCH | A network packet was blocked due to a rule mismatch in the DDS-RTPS protocol. | 58 |
| SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH | A network packet was blocked due to a rule mismatch in the DoIP protocol. | 59 |
| SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH | A network packet was blocked due to a rule mismatch on generic inspection level. | 60 |
| SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS | A network packet was blocked due to the maximal number of open TCP connections was reached. | 61 |
| SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT | A network packet was blocked due to TCP timeout. | 62 |
| SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION | A network packet was blocked due to an invalid TCP state transition. | 63 |

▽

△

| Name | Description | ID |
|---|---|---|
| SEV_FW_PACKET_BLOCKED_RATELIMIT | A network packet was blocked due to the rate limit was reached. | 64 |
| SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH | A network packet was blocked due to a rule mismatch on data link layer. | 77 |
| SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING | A network packet was blocked due to per-stream filtering in the switch. | 83 |

⌋

The Firewall provides the following context data for the SEvs:

### [CP_SWS_Fw_60001] Security event context data definition: SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH ⌈

| SEV Name | SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH | |
|---|---|---|
| ID | 77 | |
| Description | A network packet was blocked due to a rule mismatch on data link layer. | |
| Context Data Version | 2 | |
| Context Data | Data Type | Allowed Values |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>• CP: FwSEvEthernetFrameMaxLength<br>• AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60020] Security event context data definition: SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH ⌈

| SEV Name | SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH | |
|---|---|---|
| ID | 51 | |
| Description | A network packet was blocked due to a rule mismatch on IPv4 layer. | |
| Context Data Version | 2 | |
| Context Data | Data Type | Allowed Values |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60021] Security event context data definition: SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH* | |
|---|---|---|
| *ID* | 52 | |
| *Description* | A network packet was blocked due to a rule mismatch on IPv6 layer. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60022] Security event context data definition: SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH* | |
|---|---|---|
| *ID* | 53 | |
| *Description* | A network packet was blocked due to a rule mismatch within the ICMP protocol. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60023] Security event context data definition: SEV_FW_PACKET_BLOCKED_TCP_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_TCP_MISMATCH* | |
|---|---|---|
| *ID* | 54 | |
| *Description* | A network packet was blocked due to a rule mismatch on TCP layer. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |

▽

△

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_TCP_MISMATCH* | |
|---|---|---|
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60024] Security event context data definition: SEV_FW_PACKET_BLOCKED_UDP_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_UDP_MISMATCH* | |
|---|---|---|
| *ID* | 55 | |
| *Description* | A network packet was blocked due to a rule mismatch on UDP layer. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60025] Security event context data definition: SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH* | |
|---|---|---|
| *ID* | 56 | |
| *Description* | A network packet was blocked due to a rule mismatch in the SOME/IP protocol. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60026] Security event context data definition: SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH* | |
|---|---|---|
| *ID* | 57 | |
| *Description* | A network packet was blocked due to a rule mismatch in the SOME/IP SD protocol. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60027] Security event context data definition: SEV_FW_PACKET_BLOCKED_DDS_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_DDS_MISMATCH* | |
|---|---|---|
| *ID* | 58 | |
| *Description* | A network packet was blocked due to a rule mismatch in the DDS-RTPS protocol. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60028] Security event context data definition: SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH* | |
|---|---|---|
| *ID* | 59 | |
| *Description* | A network packet was blocked due to a rule mismatch in the DoIP protocol. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |

▽

△

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH* | |
|---|---|---|
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60029] Security event context data definition: SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH* | |
|---|---|---|
| *ID* | 60 | |
| *Description* | A network packet was blocked due to a rule mismatch on generic inspection level. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60002] Security event context data definition: SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS* | |
|---|---|---|
| *ID* | 61 | |
| *Description* | A network packet was blocked due to the maximal number of open TCP connections was reached. | |
| *Context Data Version* | 2 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60030] Security event context data definition: SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT ⌈

| ***SEV Name*** | ***SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT*** | |
|---|---|---|
| ***ID*** | 62 | |
| ***Description*** | A network packet was blocked due to TCP timeout. | |
| ***Context Data Version*** | 2 | |
| ***Context Data*** | ***Data Type*** | ***Allowed Values*** |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60031] Security event context data definition: SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION ⌈

| ***SEV Name*** | ***SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION*** | |
|---|---|---|
| ***ID*** | 63 | |
| ***Description*** | A network packet was blocked due to an invalid TCP state transition. | |
| ***Context Data Version*** | 2 | |
| ***Context Data*** | ***Data Type*** | ***Allowed Values*** |
| Length | uint16 | Length of EthernetFrame byte array |
| EthernetFrame | uint8 [54] | Received EthernetFrame, truncated to the first bytes according to<br>- CP: FwSEvEthernetFrameMaxLength<br>- AP: maxLength of the ContextDataElement Ethernet Frame from the SecurityExtract |
| | MAX-LENGTH | Truncation length can be defined on a project specific basis. AUTOSAR has defined a default value, as given in the EthernetFrame byte array definition above. |

⌋

### [CP_SWS_Fw_60003] Security event context data definition: SEV_FW_PACKET_BLOCKED_RATELIMIT ⌈

| ***SEV Name*** | ***SEV_FW_PACKET_BLOCKED_RATELIMIT*** | |
|---|---|---|
| ***ID*** | 64 | |
| ***Description*** | A network packet was blocked due to the rate limit was reached. | |
| ***Context Data Version*** | 2 | |
| ***Context Data*** | ***Data Type*** | ***Allowed Values*** |
| MAC_Address | uint8 [6] | |

⌋

**[CP_SWS_Fw_60032] Security event context data definition: SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING** ⌈

| *SEV Name* | *SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING* | |
|---|---|---|
| *ID* | 83 | |
| *Description* | A network packet was blocked due to per-stream filtering in the switch. | |
| *Context Data Version* | 1 | |
| *Context Data* | *Data Type* | *Allowed Values* |
| BucketId | uint8 | |
| CountValue | uint32 | |

⌋

### 7.6.2 Raising SEvs

With regards to the general pattern matching process, the Firewall can raise SEvs in two cases: Either the network packet does not match any FirewallRule and the default action is performed or the network packet matches a defined FirewallRule and the respective action is performed. In this release, SEvs are only raised in the first case, i.e. if no FirewallRule matches. The second case will be added in a later release. In the no-match case, SEvs make only sense when the firewall is configured to block unspecified network packets as default action.

In this case, the Firewall has to identify on which network protocol the violation occurred to raise the corresponding SEv. To this end, the Firewall has to identify the rule that fits the no-matched network packet best by calculating the least distance as follows:

**[CP_SWS_Fw_60004]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action, the Firewall shall identify the network protocol that was not matching the FirewallRules. To this end, the Firewall shall iterate over all FirewallRules and identify the rules for which most of the protocol fields have matched the actual network packet data starting from the lowest ISO OSI Layer and going the ISO OSI Layers upwards. The protocol of the first ISO OSI Layer, starting from the lowest ISO OSI Layer, which has a non match is the network protocol that shall be considered not to match the FirewallRules.⌋

The following example illustrates the mechanism

| Protocol<br>*Field* | IP<br>*IP addr* | TCP<br>*Port* | SOME/IP<br>*Service ID* |
|---|---|---|---|
| Network Packet | 1.2.3.4 | 1000 | 0xABCD |
| FW Rule #1 | 1.2.3.4 | 1000 | 0x1234 |
| FW Rule #2 | 1.2.3.4 | 1000 | 0x3456 |
| FW Rule #3 | 1.2.3.4 | 2000 | 0x5678 |
| FW Rule #4 | 5.6.7.8 | 3000 | 0x5678 |
| FW Rule #5 | 5.6.7.8 | 3000 | 0xABCD |

**Figure 7.5: SEV protocol matching process**

The incoming network packet matches none of the defined rules, so the default action applies here. The network packet matches the `FirewallNetworkLayerIpv4FilterConfig` and `FirewallTransportLayerFilterConfig` for rule number 1 and 2, only `FirewallNetworkLayerIpv4FilterConfig` for rule number 3 and only `FirewallSomeipProtocolFilterConfig` for rule number 5. Rule 1 and 2 have the most succeeding matching ISO `OSI Layer`s starting from the lowest network layer (in contrast to Rule 5, for example, that has a match on SOME/IP layer but no matches on lower layers.). The rule mismatch is hence occurring on the SOME/IP layer and a `SEv` shall be raised for this protocol.

**[CP_SWS_Fw_60033]**

*Status:* DRAFT

⌈If security event reporting has been enabled for the Firewall module (`FirwallEnableSecurityEventReporting` = true) the respective security events shall be reported to the IdsM via the interfaces defined in CP_SWS_BSWGeneral [3].⌋

**[CP_SWS_Fw_60034] Firewall security event context data**

*Status:* DRAFT

⌈When reporting a security event to the IdsM, the Firewall shall provide the Ethernet frame of the corresponding network packet truncated to the first `FirewallSEvEthernetFrameMaxLength` bytes as context data.⌋

**[CP_SWS_Fw_60005]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the `FirewallRule`s is Ethernet, the Firewall shall raise the `SEv` `SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH` to the `IdsM`.⌋

**[CP_SWS_Fw_60006]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is IPv4, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60007]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is IPv6, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60008]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is ICMP, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60009]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is TCP, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_TCP_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60010]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is UDP, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_UDP_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60011]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is SOME/IP, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60012]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is SOME/IP-SD, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60013]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is DDS, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_DDS_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60014]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and the network protocol that was not matching the FirewallRules is DoIP, the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH to the IdsM.⌋

**[CP_SWS_Fw_60015]**

*Status:* DRAFT

⌈If a network packet is blocked by the default action and no network protocol that was not matching the FirewallRules could be identified (e.g. because there was a mismatch in the payload using a FirewallPayloadBytePatternFilterConfig), the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH to the IdsM.⌋

In addition to pattern mismatches, the Firewall shall also raise SEvs for network packets that have been blocked due to the stateful nature of TCP

**[CP_SWS_Fw_60016]**

*Status:* DRAFT

⌈If a network packet is blocked due to the maximum number of connections reached (described in [CP_SWS_Fw_30013]), the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS to the IdsM.⌋

**[CP_SWS_Fw_60017]**

*Status:* DRAFT

⌈If a network packet is blocked due to the TCP timeout filter described in [CP_SWS_Fw_30011], the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT to the IdsM.⌋

**[CP_SWS_Fw_60018]**

*Status:* DRAFT

⌈If a network packet is blocked due to the TCP state transition filter described in [CP_SWS_Fw_30014], the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION to the IdsM.⌋

Finally, network packets can also be dropped due to the rate limiting feature described in Sec. 7.4.2

**[CP_SWS_Fw_60019]**

*Status:* DRAFT

⌈If a network packet is blocked due to the rate limiting feature described in [CP_SWS_Fw_40105], the Firewall shall raise the SEv SEV_FW_PACKET_BLOCKED_RATELIMIT to the IdsM.⌋

Note that the trigger condition for the SEv SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING is described in Chapter 7.5.2.2 within [CP_SWS_Fw_50005].

## 7.7 Error Classification

Chapter [3, General Specification of Basic Software Modules] 7.2 *"Error Handling"* describes the error handling of the Basic Software in detail. Above all, it constitutes a classification scheme consisting of five error types which may occur in BSW modules.

Based on this foundation, the following section specifies particular errors arranged in the respective subsections below.

### 7.7.1 Development Errors

**[CP_SWS_Fw_91000] Definition of development errors in module Fw**

*Status:* DRAFT
*Upstream requirements:* SRS_BSW_00337

⌈

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| API function called before Fw has been fully initialized. | FW_E_PARAM_UNINIT | 0x00 |
| The service Fw_Init is called while the module is already initialized. | FW_E_ALREADY_INITIALIZED | 0x01 |
| The (de-)activation of switch core firewall rules has failed. | FW_E_SWITCHRULEMGMT_FAILED | 0x02 |

⌋

### 7.7.2 Runtime Errors

There are no runtime errors.

### 7.7.3 Production Errors

There are no production errors.

# 8 API specification

## 8.1 Imported types

In this chapter all types included from the following files are listed.

**[CP_SWS_Fw_91012] Definition of imported datatypes of module Fw** ⌈

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| Comtype | ComStack_Types.h | PduIdType |
| | ComStack_Types.h | PduInfoType |
| | ComStack_Types.h | PduLengthType |
| Eth | Eth_GeneralTypes.h | Eth_StreamStatisticCounterType |
| IdsM | IdsM.h | IdsM_SecurityEventIdType |
| | Rte_IdsM_Type.h | IdsM_TimestampDataType |
| Std | Std_Types.h | Std_ReturnType |

⌋

## 8.2 Type definitions

### 8.2.1 ConfigType

**[CP_SWS_Fw_91001] Definition of datatype Fw_ConfigType**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Name*** | Fw_ConfigType (draft) | |
| ***Kind*** | Structure | |
| ***Elements*** | Implementation specific | |
| | ***Type*** | – |
| | ***Comment*** | – |
| ***Description*** | Configuration data structure of the Fw module<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw.h | |

⌋

## 8.3 Function definitions

This chapter contains all APIs provided by the module.

### 8.3.1 Functions provided for EcuM and BswM

This chapter contains all APIs which are provided to EcuM and BswM via the module header (Fw.h).

#### 8.3.1.1 Fw_Init

**[CP_SWS_Fw_91003] Definition of API function Fw_Init**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | Fw_Init (draft) | |
| ***Syntax*** | `void Fw_Init (`<br>`  const Fw_ConfigType* configPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x00 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | configPtr | Component configuration structure |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Service to initialize the module Fw. It initializes all variables and sets the module state to initialized.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw.h | |

⌋

### 8.3.2 Functions provided for BswM

This chapter contains all APIs which are provided to BswM via the module header (Fw.h).

#### 8.3.2.1 Fw_SetFirewallState

**[CP_SWS_Fw_91007] Definition of API function Fw_SetFirewallState**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | Fw_SetFirewallState (draft) | |
| ***Syntax*** | `void Fw_SetFirewallState ( uint16 FirewallState )` | |
| ***Service ID [hex]*** | 0x04 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | FirewallState | State into which the firewall shall go |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function is invoked by the BswM to indicate ECU state changes.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw.h | |

⌋

### 8.3.3 Functions provided for LSduR

This chapter contains all APIs which are provided to LSduR via the module header (Fw_Cbk.h).

#### 8.3.3.1 Fw_RxIndication

**[CP_SWS_Fw_91006] Definition of callback function Fw_RxIndication**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | Fw_RxIndication (draft) | |
| ***Syntax*** | `void Fw_RxIndication ( PduIdType RxPduId, const PduInfoType* PduInfoPtr )` | |
| ***Service ID [hex]*** | 0x42 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant for different PduIds. Non reentrant for the same PduId. | |
| ***Parameters (in)*** | RxPduId | ID of the received PDU. |

▽

△

| | | |
|---|---|---|
| | PduInfoPtr | Contains the length (SduLength) of the received PDU, a pointer to a buffer (SduDataPtr) containing the PDU, and the MetaData related to this PDU. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Indication of a received PDU from a lower layer communication interface module.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw.h | |

⌋

### 8.3.4 Functions provided for EthIf

This chapter contains all APIs which are provided to EthIF via the module header (Fw_Cbk.h).

#### 8.3.4.1 Fw_StreamStatisticsIndication

**[CP_SWS_Fw_91008] Definition of callback function Fw_StreamStatisticsIndication**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | Fw_StreamStatisticsIndication (draft) | |
| ***Syntax*** | `void Fw_StreamStatisticsIndication (`<br>`  uint8 SwitchIdx,`<br>`  uint8 NumberOfBuckets,`<br>`  const Eth_StreamStatisticCounterType* ListOfBucketsPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x05 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | SwitchIdx | Index of the switch within the context of the Ethernet Switch Driver |
| | NumberOfBuckets | Number of counting buckets in the switch |
| | ListOfBucketsPtr | Pointer to the bucket counter values |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | The function is called by the lower layer once it has successfully retrieved the stream statistics (i.e. bucket counter values) from the EthSwt driver given with SwitchIdx<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw_Cbk.h | |

⌋

#### 8.3.4.2 Fw_StreamStateIndication

**[CP_SWS_Fw_91009] Definition of callback function Fw_StreamStateIndication**

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| ***Service Name*** | Fw_StreamStateIndication (draft) | |
| ***Syntax*** | `void Fw_StreamStateIndication (`<br>`  uint8 SwitchIdx,`<br>`  uint8 StreamHandleIdxPtr,`<br>`  boolean StreamActivityStatus`<br>`)` | |
| ***Service ID [hex]*** | 0x06 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | SwitchIdx | Index of the switch within the context of the Ethernet Switch Driver |
| | StreamHandleIdxPtr | Pointer to the StreamHandleIdx for which the current status is returned |
| | StreamActivityStatus | Activity status of the StreamHandleIdx (True = active, False = inactive) |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | The function is called by the EthIf once it has successfully set the StreamHandleIdx in the switch.<br>**Tags:** atp.Status=draft | |
| ***Available via*** | Fw_Cbk.h | |

⌋

## 8.4 Scheduled functions

These functions are directly called by Basic Software Scheduler. The following functions shall have no return value and no parameter. All functions shall be non reentrant.

### 8.4.1 MainFunction

**[CP_SWS_Fw_91005] Definition of scheduled function Fw_MainFunction**

*Status:* DRAFT

⌈

| | |
|---|---|
| ***Service Name*** | Fw_MainFunction (draft) |
| ***Syntax*** | `void Fw_MainFunction (`<br>`  void`<br>`)` |
| ***Service ID [hex]*** | 0x02 |

▽

△

| | |
|---|---|
| ***Description*** | This function is called periodically. It is used to perform asynchronous function calls (e.g. to the switch driver).<br>**Tags:** atp.Status=draft |
| ***Available via*** | Fw.h |

⌋

## 8.5 Expected interfaces

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

### 8.5.1 Mandatory interfaces

This section defines all interfaces, which are required to fulfill the core functionality of the module.

**[CP_SWS_Fw_91011] Definition of mandatory interfaces required by module Fw**
⌈

| ***API Function*** | ***Header File*** | ***Description*** |
|---|---|---|
| LSduR_FwRxIndication (draft) | LSduR_<module>.h | Indication of a received PDU from a lower layer communication interface module. |

⌋

### 8.5.2 Optional interfaces

This section defines all interfaces, which are required to fulfill an optional functionality of the module.

**[CP_SWS_Fw_91010] Definition of optional interfaces requested by module Fw**
⌈

| ***API Function*** | ***Header File*** | ***Description*** |
|---|---|---|
| Det_ReportError | Det.h | Service to report development errors. |
| EthIf_GetStreamStatistics (draft) | EthIf.h | Requests the statistics (bucket counter values) of an Ethernet switch of all configured streams.<br>**Tags:** atp.Status=draft |
| EthIf_SetStreamState (draft) | EthIf.h | This function is called by the Firewall module to control the activity status of a stream in the Ethernet switch.<br>**Tags:** atp.Status=draft |
| IdsM_ReportSecurityEvent | IdsM.h | This API is the interface to report security events to the IdsM. |

⌋

## 8.6 Service Interfaces

No service interfaces are required by the Firewall

# 9 Sequence diagrams

## 9.1 Switch core filter rule extraction

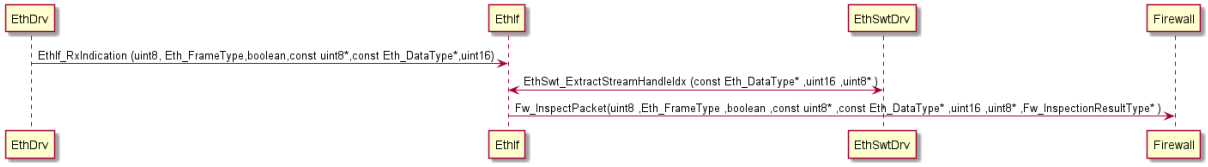



**Figure 9.1: Extraction of the switch filter rule from the modified network packet header**

## 9.2 Switch core filter rule counter statistics

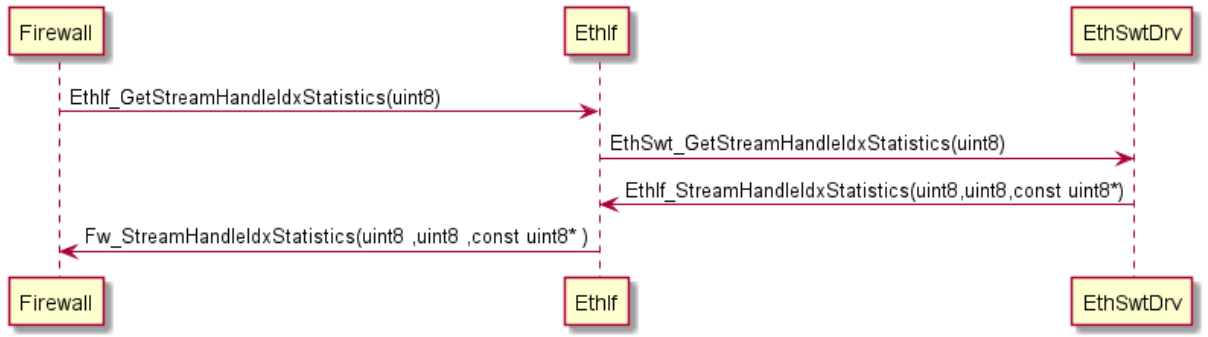



**Figure 9.2: Polling mechanism to retrieve the switch core filter rule counter values**

## 9.3 Switch core filter rule management

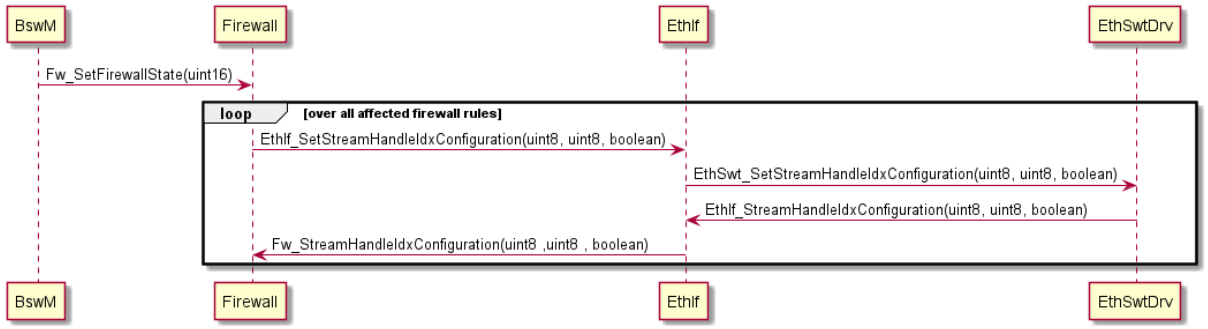



**Figure 9.3: Management mechanism for switch core filter rules**

# 10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

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

Chapter 10.3 specifies published information of the module Firewall.

## 10.1 How to read this chapter

For details refer to [3] Chapter 10.1 *"Introduction to configuration specification"*.

## 10.2 Containers and configuration parameters

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

### 10.2.1 FirewallGeneral

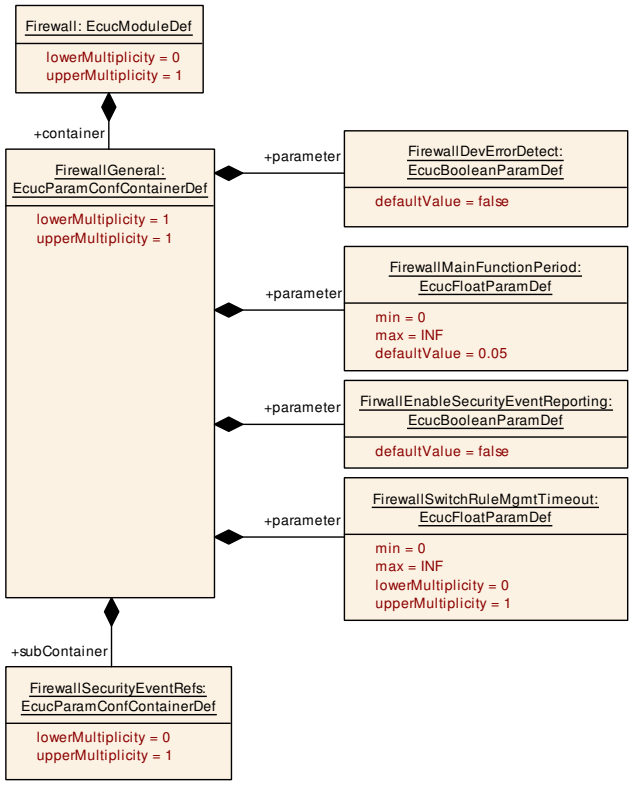



**Figure 10.1: General Firewall Configuration**

## [ECUC_Fw_00001] Definition of EcucModuleDef Firewall

*Status:* DRAFT

⌈

| | |
|---|---|
| **Module Name** | Firewall |
| **Description** | Configuration of the Firewall module. |
| **Post-Build Selectable Support** | true |
| **Post-Build Loadable Support** | true |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallConfig | 1 | This container contains the configuration parameters and sub containers of the Firewall module.<br>**Tags:** atp.Status=draft |
| FirewallGeneral | 1 | Contains the general configuration parameters of the module.<br>**Tags:** atp.Status=draft |
| FirewallRule | 1..* | Firewall Rule that defines the control information in individual packets.<br>**Tags:** atp.Status=draft |

▽

△

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallState | 1..* | Collection of Firewall states in which the Firewall may be activated (via the FirewallStateRef).<br>**Tags:** atp.Status=draft |
| FirewallStateDependentRules | 1..* | Firewall rules that are defined in a firewall state<br>**Tags:** atp.Status=draft |
| FirewallSwitchBucketCounterStatistics | 0..* | Polling of switch bucket counter statistics<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00002] Definition of EcucParamConfContainerDef FirewallGeneral

*Status:* DRAFT

⌈

| **Container Name** | FirewallGeneral |
|---|---|
| **Parent Container** | Firewall |
| **Upstream mapping defined** | – |
| **Description** | Contains the general configuration parameters of the module.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDevErrorDetect | 1 | [ECUC_Fw_00003] |
| FirewallMainFunctionPeriod | 1 | [ECUC_Fw_00005] |
| FirewallSwitchRuleMgmtTimeout | 0..1 | [ECUC_Fw_00142] |
| FirwallEnableSecurityEventReporting | 1 | [ECUC_Fw_00116] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallSecurityEventRefs | 0..1 | Container for the references to IdsMEvent elements representing the security events that the Firwall module shall report to the Ids M in case the coresponding security related event occurs (and if FirewallEnableSecurityEventReporting is set to "true"). The standardized security events in this container can be extended by vendor-specific security events.<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_Fw_00003] Definition of EcucBooleanParamDef FirewallDevErrorDetect

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallDevErrorDetect |
| **Parent Container** | FirewallGeneral |
| **Upstream mapping defined** | – |
| **Description** | Switches the development error detection and notification on or off.<br>• true: detection and notification is enabled.<br>• false: detection and notification is disabled.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucBooleanParamDef |
| **Default value** | false |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00005] Definition of EcucFloatParamDef FirewallMainFunctionPeriod

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallMainFunctionPeriod |
| **Parent Container** | FirewallGeneral |
| **Upstream mapping defined** | – |
| **Description** | Execution cycle of the respective Firewall_MainFunction instance in seconds.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucFloatParamDef |
| **Range** | ]0 .. INF[ |
| **Default value** | 0.05 |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00142] Definition of EcucFloatParamDef FirewallSwitchRuleMgmtTimeout

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallSwitchRuleMgmtTimeout |
| **Parent Container** | FirewallGeneral |
| **Upstream mapping defined** | – |
| **Description** | Timeout to wait for a confirmation of a switch core configuration request.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | EcucFloatParamDef |

▽

△

| Range | ]0 .. INF[ | |
|---|---|---|
| Default value | – | |
| Pre-Compile only | true | |

⌋

### [ECUC_Fw_00116] Definition of EcucBooleanParamDef FirwallEnableSecurityEventReporting

*Status:* DRAFT

⌈

| Parameter Name | FirwallEnableSecurityEventReporting |
|---|---|
| Parent Container | FirewallGeneral |
| Upstream mapping defined | – |
| Description | Switches the reporting of security events to the IdsM: - true: reporting is enabled. - false: reporting is disabled.<br>**Tags:** atp.Status=draft |
| Multiplicity | 1 |
| Type | EcucBooleanParamDef |
| Default value | false |
| Pre-Compile only | true |

⌋

## 10.2.2 Firewall Pdu Routing

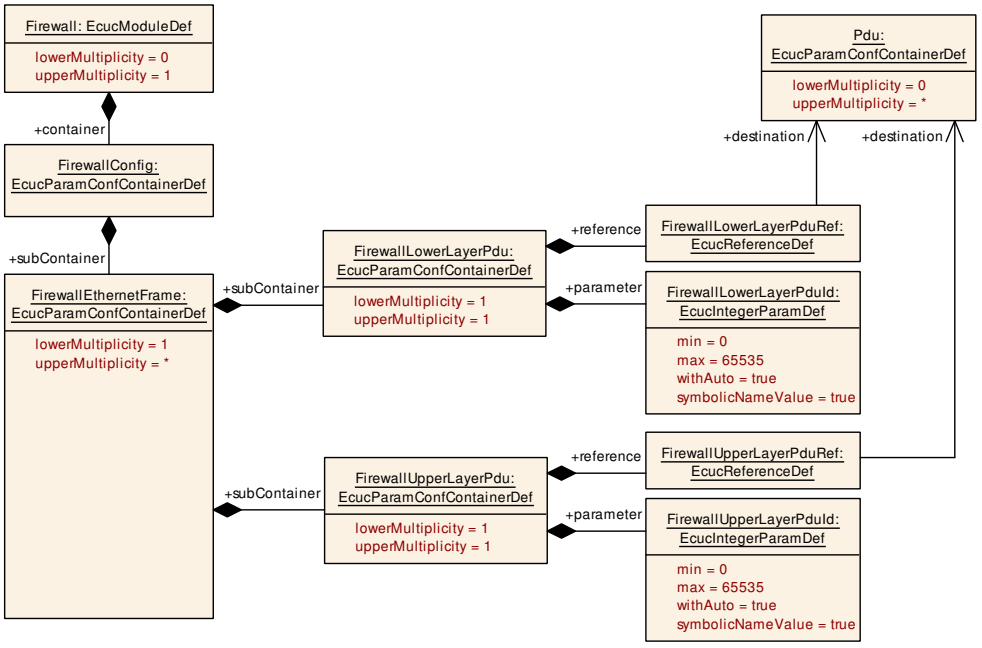



**Figure 10.2: Firewall Configuration Pdu Routing**

## [ECUC_Firewall_00146] Definition of EcucParamConfContainerDef FirewallEthernetFrame

*Status:* DRAFT

⌈

| Container Name | FirewallEthernetFrame | |
|---|---|---|
| Parent Container | FirewallConfig | |
| Upstream mapping defined | – | |
| Description | An ethernet frame to be filtered.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1..* | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| No Included Parameters |
|---|

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallLowerLayerPdu | 1 | Represents the lower layer PDU associated with an Ethernet Frame. This PDU is usually linked to the EthIf via LSduR. It may accept any meta data item types for forwarding to the upper layer, but may consume meta data items of the types BROADCAST_8 and FILTER_RULE_ID_16.<br>**Tags:** atp.Status=draft |
| FirewallUpperLayerPdu | 1 | Represents the upper layer PDU associated with an Ethernet Frame. This PDU is linked to the LSduR and its upper layer modules, e.g. the TcpIp. It may provide any meta data item types forwarded from the lower layer.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Firewall_00147] Definition of EcucParamConfContainerDef FirewallLowerLayerPdu

*Status:* DRAFT

⌈

| Container Name | FirewallLowerLayerPdu |
|---|---|
| Parent Container | FirewallEthernetFrame |
| Upstream mapping defined | – |
| Description | Represents the lower layer PDU associated with an EthernetFrame. This PDU is usually linked to the EthIf via LSduR. It may accept any meta data item types for forwarding to the upper layer, but may consume meta data items of the types BROADCAST_8 and FILTER_RULE_ID_16.<br>**Tags:** atp.Status=draft |
| Multiplicity | 1 |
| Pre-Compile only | true |
| Configuration Parameters | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallLowerLayerPduId | 1 | [ECUC_Firewall_00149] |
| FirewallLowerLayerPduRef | 1 | [ECUC_Firewall_00148] |

| No Included Containers |
|---|

⌋

### [ECUC_Firewall_00149] Definition of EcucIntegerParamDef FirewallLowerLayerPduId

*Status:* DRAFT

⌈

| **Parameter Name** | FirewallLowerLayerPduId | |
|---|---|---|
| **Parent Container** | FirewallLowerLayerPdu | |
| **Upstream mapping defined** | – | |
| **Description** | PDU identifier used for RxIndication from LSduR.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **With Auto** | true | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_Firewall_00148] Definition of EcucReferenceDef FirewallLowerLayerPduRef

*Status:* DRAFT

⌈

| **Parameter Name** | FirewallLowerLayerPduRef | |
|---|---|---|
| **Parent Container** | FirewallLowerLayerPdu | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to the global PDU.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Firewall_00150] Definition of EcucParamConfContainerDef FirewallUpperLayerPdu

*Status:* DRAFT

⌈

| Container Name | FirewallUpperLayerPdu |
|---|---|
| Parent Container | FirewallEthernetFrame |
| Upstream mapping defined | – |
| Description | Represents the upper layer PDU associated with an EthernetFrame. This PDU is linked to the LSduR and its upper layer modules, e.g. the TcpIp. It may provide any meta data item types forwarded from the lower layer.<br>**Tags:** atp.Status=draft |
| Multiplicity | 1 |
| Pre-Compile only | true |
| Configuration Parameters | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallUpperLayerPduId | 1 | [ECUC_Firewall_00152] |
| FirewallUpperLayerPduRef | 1 | [ECUC_Firewall_00151] |

| No Included Containers |
|---|

⌋

### [ECUC_Firewall_00152] Definition of EcucIntegerParamDef FirewallUpperLayerPduId

*Status:* DRAFT

⌈

| Parameter Name | FirewallUpperLayerPduId |
|---|---|
| Parent Container | FirewallUpperLayerPdu |
| Upstream mapping defined | – |
| Description | PDU identifier used for ReleaseRxBuffer from LSduR.<br>**Tags:** atp.Status=draft |
| Multiplicity | 1 |
| Type | EcucIntegerParamDef (Symbolic Name generated for this parameter) |
| Range | 0 .. 65535 |
| Default value | – |
| With Auto | true |
| Pre-Compile only | true |

⌋

### [ECUC_Firewall_00151] Definition of EcucReferenceDef FirewallUpperLayerPduRef

*Status:* DRAFT

⌈

| Parameter Name | FirewallUpperLayerPduRef |
|---|---|
| Parent Container | FirewallUpperLayerPdu |
| Upstream mapping defined | – |

▽

△

| | | |
|---|---|---|
| **Description** | Reference to the global PDU.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to Pdu | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.3 Connection to BswM

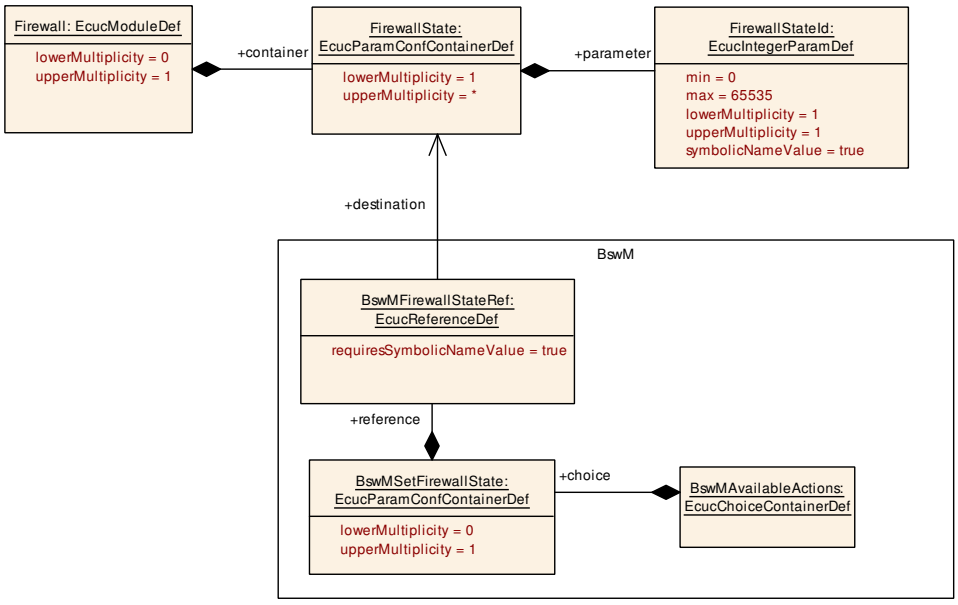



**Figure 10.3: Connection to the BswM**

## [ECUC_Fw_00141] Definition of EcucParamConfContainerDef FirewallConfig

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallConfig |
| **Parent Container** | Firewall |
| **Upstream mapping defined** | – |
| **Description** | This container contains the configuration parameters and sub containers of the Firewall module.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallInitialStateRef | 1 | [ECUC_Fw_00106] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallEthernetFrame | 1..* | An ethernet frame to be filtered.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00106] Definition of EcucReferenceDef FirewallInitialStateRef

*Status:* DRAFT

⌈

| **Parameter Name** | FirewallInitialStateRef | |
|---|---|---|
| **Parent Container** | FirewallConfig | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to the Firewall State that is defined as the default state.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | Reference to FirewallState | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00107] Definition of EcucParamConfContainerDef FirewallState

*Status:* DRAFT

⌈

| **Container Name** | FirewallState | |
|---|---|---|
| **Parent Container** | Firewall | |
| **Upstream mapping defined** | – | |
| **Description** | Collection of Firewall states in which the Firewall may be activated (via the Firewall StateRef).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallStateId | 1 | [ECUC_Fw_00108] |

| No Included Containers |
|---|

⌋

## [ECUC_Fw_00108] Definition of EcucIntegerParamDef FirewallStateId

*Status:* DRAFT

⌈

| Parameter Name | FirewallStateId | |
|---|---|---|
| Parent Container | FirewallState | |
| Upstream mapping defined | – | |
| Description | Parameter that identifies the Firewall State.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef (Symbolic Name generated for this parameter) | |
| Range | 0 .. 65535 | |
| Default value | – | |
| Pre-Compile only | true | |

⌋

## [ECUC_Fw_00006] Definition of EcucParamConfContainerDef FirewallStateDependentRules

*Status:* DRAFT

⌈

| Container Name | FirewallStateDependentRules | |
|---|---|---|
| Parent Container | Firewall | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Firewall rules that are defined in a firewall state<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1..* | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDefaultAction | 0..1 | [ECUC_Fw_00007] |
| FirewallStateRef | 1..* | [ECUC_Fw_00109] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallActionForMatchingRules | 1..* | Firewall action that is performed if the referenced pattern matches.<br>**Tags:** atp.Status=draft |

⌋

### [ECUC_Fw_00007] Definition of EcucEnumerationParamDef FirewallDefaultAction

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDefaultAction | |
| **Parent Container** | FirewallStateDependentRules | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This attribute defines a defaultAction.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | ALLOW | Allow<br>**Tags:** atp.Status=draft |
| | BLOCK | Block<br>**Tags:** atp.Status=draft |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00109] Definition of EcucReferenceDef FirewallStateRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallStateRef | |
| **Parent Container** | FirewallStateDependentRules | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to firewall states in which the Firewall is active. If one of the referenced Firewall States is active then the firewall rule shall be considered active as well.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Type** | Reference to FirewallState | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4 Filter Rules

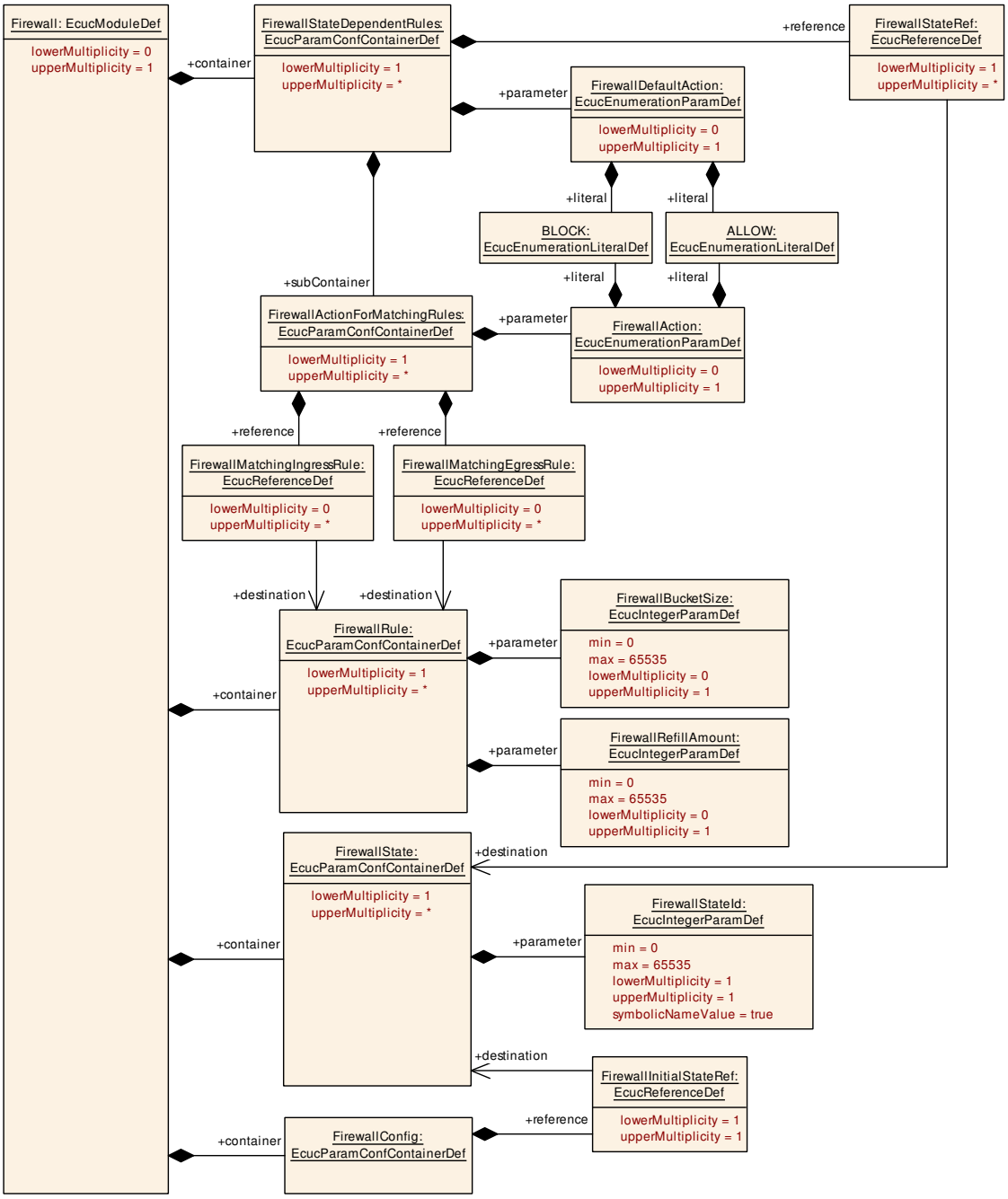



**Figure 10.4: Firewall Filter Rules**

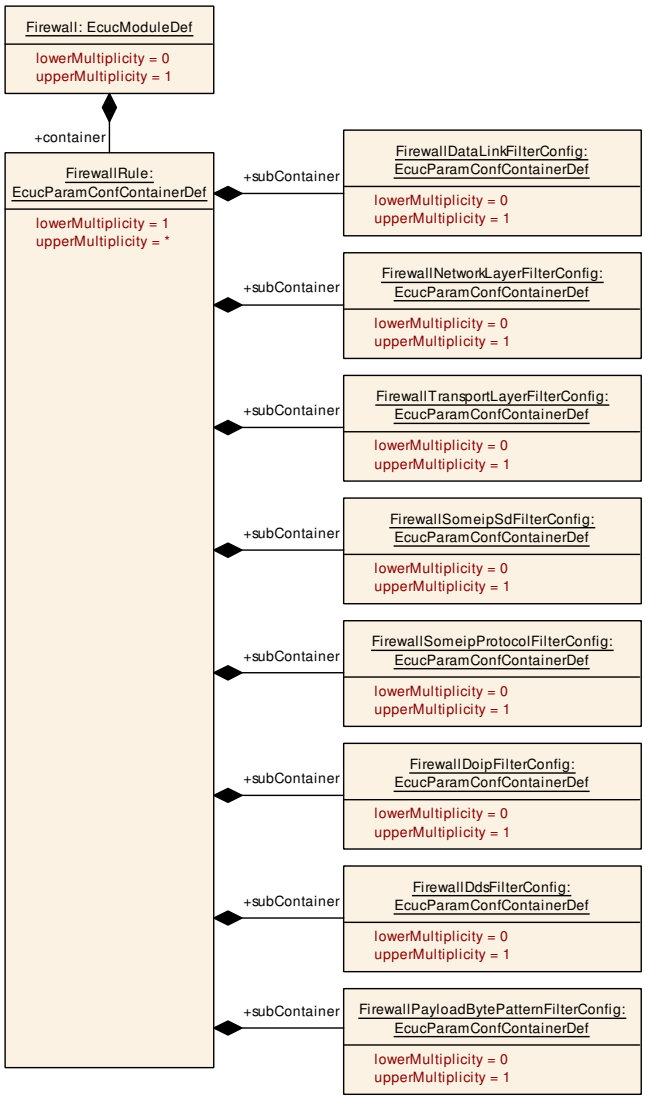



**Figure 10.5: Firewall Filter Rule with respective subrules**

## [ECUC_Fw_00011] Definition of EcucParamConfContainerDef FirewallRule

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallRule | |
| **Parent Container** | Firewall | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Firewall Rule that defines the control information in individual packets.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallBucketSize | 0..1 | [ECUC_Fw_00027] |
| FirewallRefillAmount | 0..1 | [ECUC_Fw_00026] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallDataLinkFilterConfig | 0..1 | Configuration of filter rules on the DataLink layer<br>**Tags:** atp.Status=draft |
| FirewallDdsFilterConfig | 0..1 | Configuration of filter rules for Dds<br>**Tags:** atp.Status=draft |
| FirewallDoipFilterConfig | 0..1 | Configuration of filter rules for DoIP<br>**Tags:** atp.Status=draft |
| FirewallNetworkLayerFilterConfig | 0..1 | Configuration of filter rules on the Network layer<br>**Tags:** atp.Status=draft |
| FirewallPayloadBytePatternFilterConfig | 0..1 | Configuration of a generic firewall rule that defines the individual bytes of a message that shall match.<br>**Tags:** atp.Status=draft |
| FirewallSomeipProtocolFilterConfig | 0..1 | Configuration of SOME/IP Protocol firewall rules<br>**Tags:** atp.Status=draft |
| FirewallSomeipSdFilterConfig | 0..1 | Configuration of SOME/IP Service Discovery firewall rules<br>**Tags:** atp.Status=draft |
| FirewallTransportLayerFilterConfig | 0..1 | Configuration of filter rules on Transport Layer level.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00027] Definition of EcucIntegerParamDef FirewallBucketSize

*Status:* DRAFT

⌈

| **Parameter Name** | FirewallBucketSize | |
|---|---|---|
| **Parent Container** | FirewallRule | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This attribute defines the capacity of the queue for rate limitation (leaky-bucket Algorithm).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | - | Selectable |

⌋

## [ECUC_Fw_00026] Definition of EcucIntegerParamDef FirewallRefillAmount

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallRefillAmount | |
| **Parent Container** | FirewallRule | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This attribute defines the output rate that describes how many packets leave the queue per second (leaky-bucket Algorithm).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00008] Definition of EcucParamConfContainerDef FirewallActionForMatchingRules

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallActionForMatchingRules | |
| **Parent Container** | FirewallStateDependentRules | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Firewall action that is performed if the referenced pattern matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallAction | 0..1 | [ECUC_Fw_00009] |
| FirewallMatchingEgressRule | 0..* | [ECUC_Fw_00143] |
| FirewallMatchingIngressRule | 0..* | [ECUC_Fw_00010] |

| **No Included Containers** |
|---|

⌋

## [ECUC_Fw_00009] Definition of EcucEnumerationParamDef FirewallAction

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallAction | |
| **Parent Container** | FirewallActionForMatchingRules | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Action that is performed by the firewall if the matchingRule is fulfilled.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucEnumerationParamDef | |
| **Range** | ALLOW | Allow<br>**Tags:** atp.Status=draft |
| | BLOCK | Block<br>**Tags:** atp.Status=draft |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00143] Definition of EcucReferenceDef FirewallMatchingEgressRule

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallMatchingEgressRule | |
| **Parent Container** | FirewallActionForMatchingRules | |
| **Upstream mapping defined** | – | |
| **Description** | Firewall rule expression against which the egress network traffic is matched.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..* | |
| **Type** | Reference to FirewallRule | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00010] Definition of EcucReferenceDef FirewallMatchingIngressRule

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallMatchingIngressRule |
| **Parent Container** | FirewallActionForMatchingRules |
| **Upstream mapping defined** | – |
| **Description** | Firewall rule expression against which the ingress network traffic is matched.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..* |
| **Type** | Reference to FirewallRule |

▽

△

| | | |
|---|---|---|
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.1 Data link layer configuration

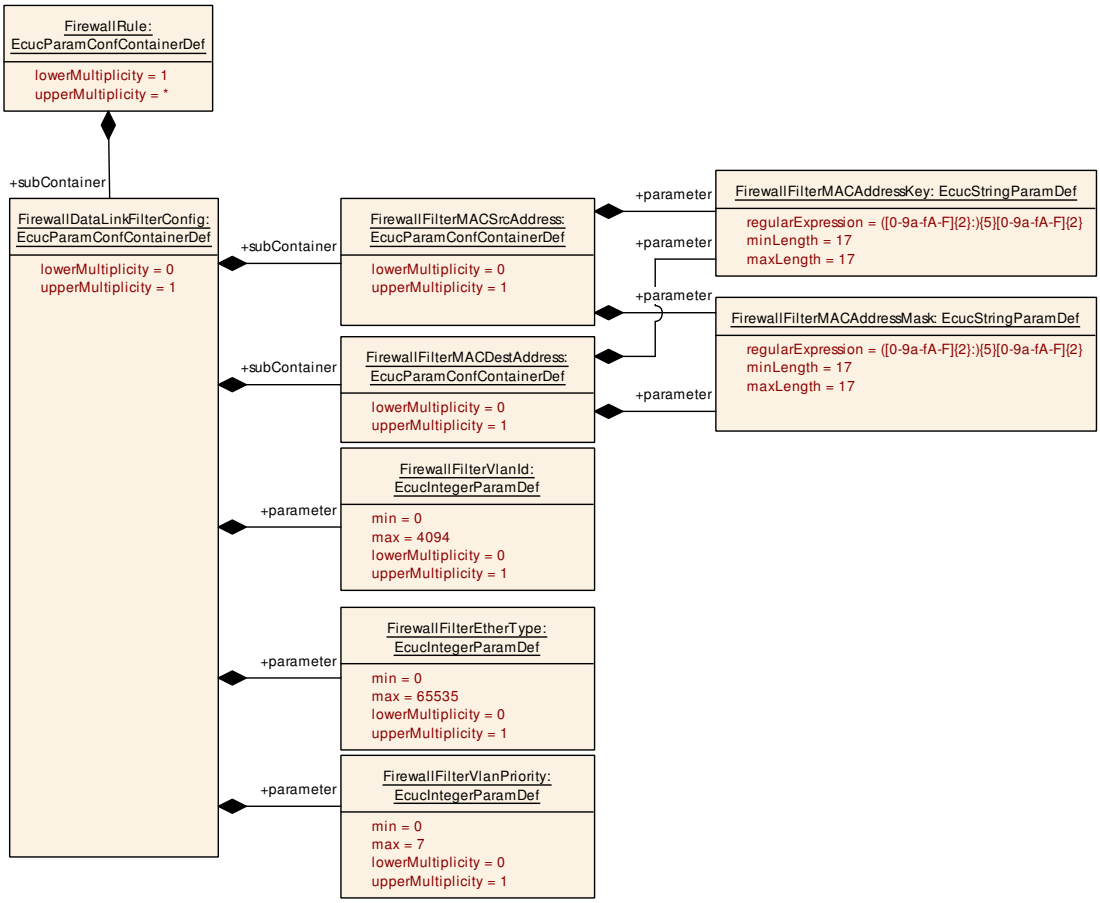



**Figure 10.6: Data link layer configuration**

## [ECUC_Fw_00139] Definition of EcucParamConfContainerDef FirewallDataLink FilterConfig

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallDataLinkFilterConfig |
| **Parent Container** | FirewallRule |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Configuration of filter rules on the DataLink layer<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterEtherType | 0..1 | [ECUC_Fw_00017] |
| FirewallFilterVlanId | 0..1 | [ECUC_Fw_00016] |
| FirewallFilterVlanPriority | 0..1 | [ECUC_Fw_00018] |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallFilterMACDestAddress | 0..1 | Configuration of one MAC destination filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterMACSrcAddress | 0..1 | Configuration of one MAC source filter.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00017] Definition of EcucIntegerParamDef FirewallFilterEtherType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterEtherType | |
| **Parent Container** | FirewallDataLinkFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the filter Ether Type.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

## [ECUC_Fw_00016] Definition of EcucIntegerParamDef FirewallFilterVlanId

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallFilterVlanId |
| **Parent Container** | FirewallDataLinkFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Definition of the filter VLAN ID.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4094 | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_Fw_00018] Definition of EcucIntegerParamDef FirewallFilterVlanPriority

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterVlanPriority | |
| **Parent Container** | FirewallDataLinkFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the filter VLAN Priority.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 7 | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_Fw_00013] Definition of EcucParamConfContainerDef FirewallFilterMACDestAddress

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallFilterMACDestAddress | |
| **Parent Container** | FirewallDataLinkFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Configuration of one MAC destination filter.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterMACAddressKey | 1 | [ECUC_Fw_00014] |
| FirewallFilterMACAddressMask | 1 | [ECUC_Fw_00015] |

| **No Included Containers** |
|---|

⌋

### [ECUC_Fw_00014] Definition of EcucStringParamDef FirewallFilterMACAddressKey

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterMACAddressKey | |
| **Parent Container** | FirewallFilterMACDestAddress, FirewallFilterMACSrcAddress | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Specifies the 48-bit physical address (MAC address) key value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucStringParamDef | |
| **Default value** | – | |
| **Length** | 17-17 | |
| **Regular Expression** | ([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2} | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00015] Definition of EcucStringParamDef FirewallFilterMACAddressMask

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterMACAddressMask | |
| **Parent Container** | FirewallFilterMACDestAddress, FirewallFilterMACSrcAddress | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Specifies the 48-bit physical address (MAC address) mask value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucStringParamDef | |
| **Default value** | – | |
| **Length** | 17-17 | |
| **Regular Expression** | ([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2} | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00012] Definition of EcucParamConfContainerDef FirewallFilterMAC-SrcAddress

*Status:* DRAFT

⌈

| Container Name | FirewallFilterMACSrcAddress | |
|---|---|---|
| Parent Container | FirewallDataLinkFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of one MAC source filter.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterMACAddressKey | 1 | [ECUC_Fw_00014] |
| FirewallFilterMACAddressMask | 1 | [ECUC_Fw_00015] |

| No Included Containers |
|---|

⌋

For parameter table [ECUC_Fw_00014] FirewallFilterMACAddressKey, see definition below container FirewallFilterMACDestAddress.

For parameter table [ECUC_Fw_00015] FirewallFilterMACAddressMask, see definition below container FirewallFilterMACDestAddress.

### 10.2.4.2 IPv4 configuration

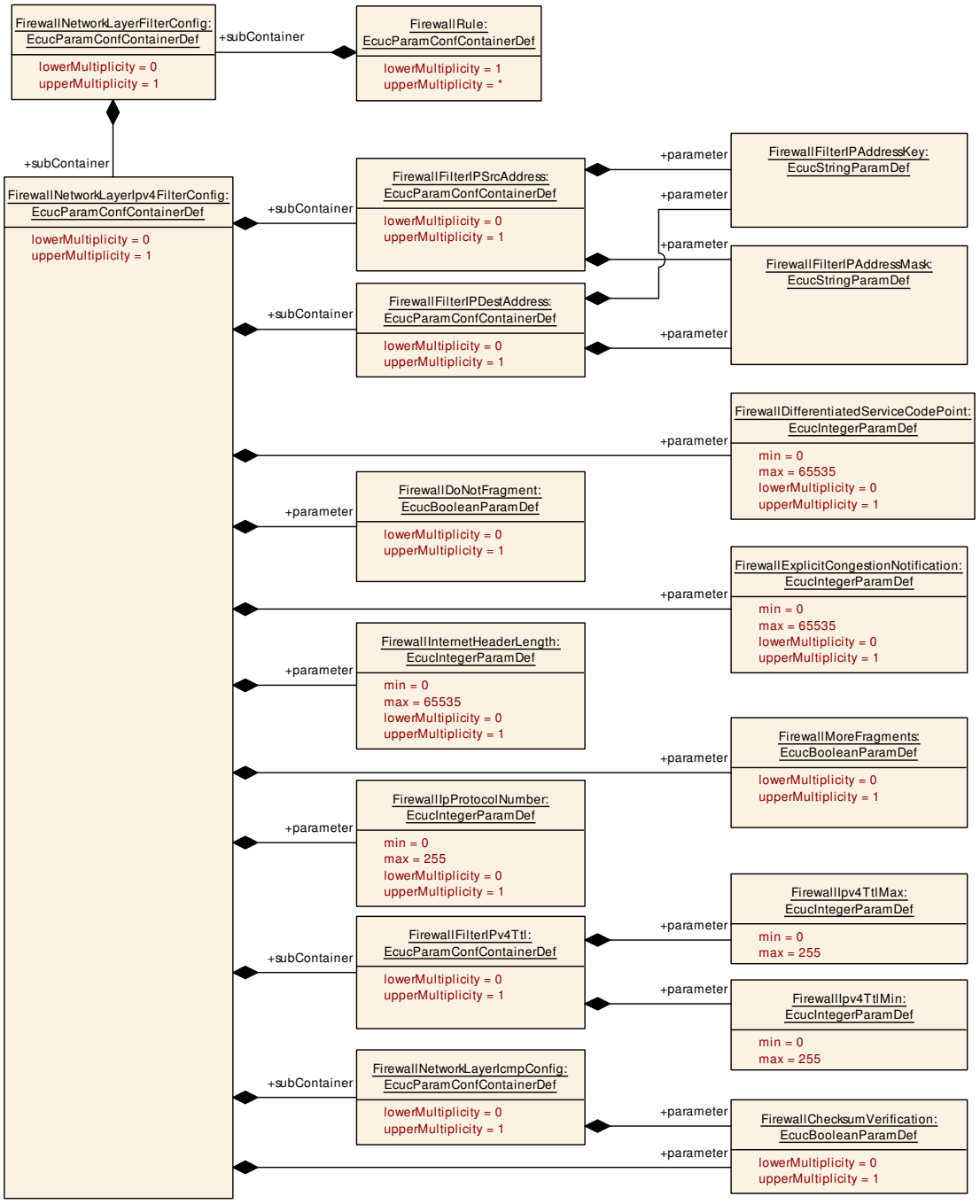



**Figure 10.7: IPv4 configuration**

## [ECUC_Fw_00030] Definition of EcucParamConfContainerDef FirewallNetwork LayerFilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallNetworkLayerFilterConfig | |
|---|---|---|
| Parent Container | FirewallRule | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules on the Network layer<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| No Included Parameters |
|---|

| Included Containers | | |
|---|---|---|
| Container Name | Multiplicity | Description |
| FirewallNetworkLayerIpv4FilterConfig | 0..1 | Configuration of filter rules for IPv6 on the Network layer<br>**Tags:** atp.Status=draft |
| FirewallNetworkLayerIpv6FilterConfig | 0..1 | Configuration of filter rules on the Network layer<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00140] Definition of EcucParamConfContainerDef FirewallNetworkLayerIpv4FilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallNetworkLayerIpv4FilterConfig | |
|---|---|---|
| Parent Container | FirewallNetworkLayerFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules for IPv6 on the Network layer<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| Parameter Name | Multiplicity | ECUC ID |
| FirewallChecksumVerification | 0..1 | [ECUC_Fw_00025] |
| FirewallDifferentiatedServiceCodePoint | 0..1 | [ECUC_Fw_00040] |
| FirewallDoNotFragment | 0..1 | [ECUC_Fw_00041] |
| FirewallExplicitCongestionNotification | 0..1 | [ECUC_Fw_00045] |
| FirewallInternetHeaderLength | 0..1 | [ECUC_Fw_00042] |
| FirewallIpProtocolNumber | 0..1 | [ECUC_Fw_00044] |
| FirewallMoreFragments | 0..1 | [ECUC_Fw_00043] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallFilterIPDestAddress | 0..1 | Configuration of one IP destination filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterIPSrcAddress | 0..1 | Configuration of one IP source filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterIPv4Ttl | 0..1 | Filter to match packets with a ttl value (TimeToLive defines the lifetime of data on the network).<br>**Tags:** atp.Status=draft |
| FirewallNetworkLayerIcmpConfig | 0..1 | Configuration of filter rules for ICMP (Internet Control Message Protocol).<br>**Tags:** atp.Status=draft |

⌋

For parameter table [ECUC_Fw_00025] FirewallChecksumVerification, see definition below container FirewallNetworkLayerIcmpConfig.

### [ECUC_Fw_00040] Definition of EcucIntegerParamDef FirewallDifferentiatedServiceCodePoint

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDifferentiatedServiceCodePoint | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a DSCP value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00041] Definition of EcucBooleanParamDef FirewallDoNotFragment

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallDoNotFragment |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |
| **Description** | Filter to match packets that have the doNotFragment bit in the Header set.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |

▽

△

| | | |
|---|---|---|
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00045] Definition of EcucIntegerParamDef FirewallExplicitCongestionNotification

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallExplicitCongestionNotification | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a ECN code point.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00042] Definition of EcucIntegerParamDef FirewallInternetHeaderLength

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallInternetHeaderLength | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a minimum ipv4 header length.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00044] Definition of EcucIntegerParamDef FirewallIpProtocolNumber

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallIpProtocolNumber | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a IP protocol number .<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00043] Definition of EcucBooleanParamDef FirewallMoreFragments

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallMoreFragments | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets that have the moreFragments flag in the Header set.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00032] Definition of EcucParamConfContainerDef FirewallFilterIPDestAddress

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallFilterIPDestAddress |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig, FirewallNetworkLayerIpv6FilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |
| **Description** | Configuration of one IP destination filter.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |

▽

△

| Post-Build Multiplicity | Loadable | Selectable |
|---|---|---|
| **Configuration Parameters** | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterIPAddressKey | 1 | [ECUC_Fw_00033] |
| FirewallFilterIPAddressMask | 1 | [ECUC_Fw_00034] |

| No Included Containers |
|---|

⌋

### [ECUC_Fw_00033] Definition of EcucStringParamDef FirewallFilterIPAddressKey

*Status:* DRAFT

⌈

| Parameter Name | FirewallFilterIPAddressKey | |
|---|---|---|
| Parent Container | FirewallFilterIPDestAddress, FirewallFilterIPSrcAddress | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | IP address key pattern.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucStringParamDef | |
| Default value | – | |
| Regular Expression | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00034] Definition of EcucStringParamDef FirewallFilterIPAddress Mask

*Status:* DRAFT

⌈

| Parameter Name | FirewallFilterIPAddressMask | |
|---|---|---|
| Parent Container | FirewallFilterIPDestAddress, FirewallFilterIPSrcAddress | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | IP address mask pattern.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucStringParamDef | |
| Default value | – | |
| Regular Expression | – | |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00031] Definition of EcucParamConfContainerDef FirewallFilterIPSrcAddress

*Status:* DRAFT

⌈

| Container Name | FirewallFilterIPSrcAddress | |
|---|---|---|
| Parent Container | FirewallNetworkLayerIpv4FilterConfig, FirewallNetworkLayerIpv6FilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of one IP source filter.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterIPAddressKey | 1 | [ECUC_Fw_00033] |
| FirewallFilterIPAddressMask | 1 | [ECUC_Fw_00034] |

| No Included Containers |
|---|

⌋

For parameter table [ECUC_Fw_00033] FirewallFilterIPAddressKey, see definition below container FirewallFilterIPDestAddress.

For parameter table [ECUC_Fw_00034] FirewallFilterIPAddressMask, see definition below container FirewallFilterIPDestAddress.

### [ECUC_Fw_00046] Definition of EcucParamConfContainerDef FirewallFilterIPv4Ttl

*Status:* DRAFT

⌈

| Container Name | FirewallFilterIPv4Ttl | |
|---|---|---|
| Parent Container | FirewallNetworkLayerIpv4FilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match packets with a ttl value (TimeToLive defines the lifetime of data on the network).<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallIpv4TtlMax | 1 | [ECUC_Fw_00047] |
| FirewallIpv4TtlMin | 1 | [ECUC_Fw_00048] |

| No Included Containers |
|---|

⌋

### [ECUC_Fw_00047] Definition of EcucIntegerParamDef FirewallIpv4TtlMax

*Status:* DRAFT

⌈

| Parameter Name | FirewallIpv4TtlMax | |
|---|---|---|
| Parent Container | FirewallFilterIPv4Ttl | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match packets with a max ttl value.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 255 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00048] Definition of EcucIntegerParamDef FirewallIpv4TtlMin

*Status:* DRAFT

⌈

| Parameter Name | FirewallIpv4TtlMin | |
|---|---|---|
| Parent Container | FirewallFilterIPv4Ttl | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match packets with a min ttl value.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 255 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### 10.2.4.3 IPv6 configuration

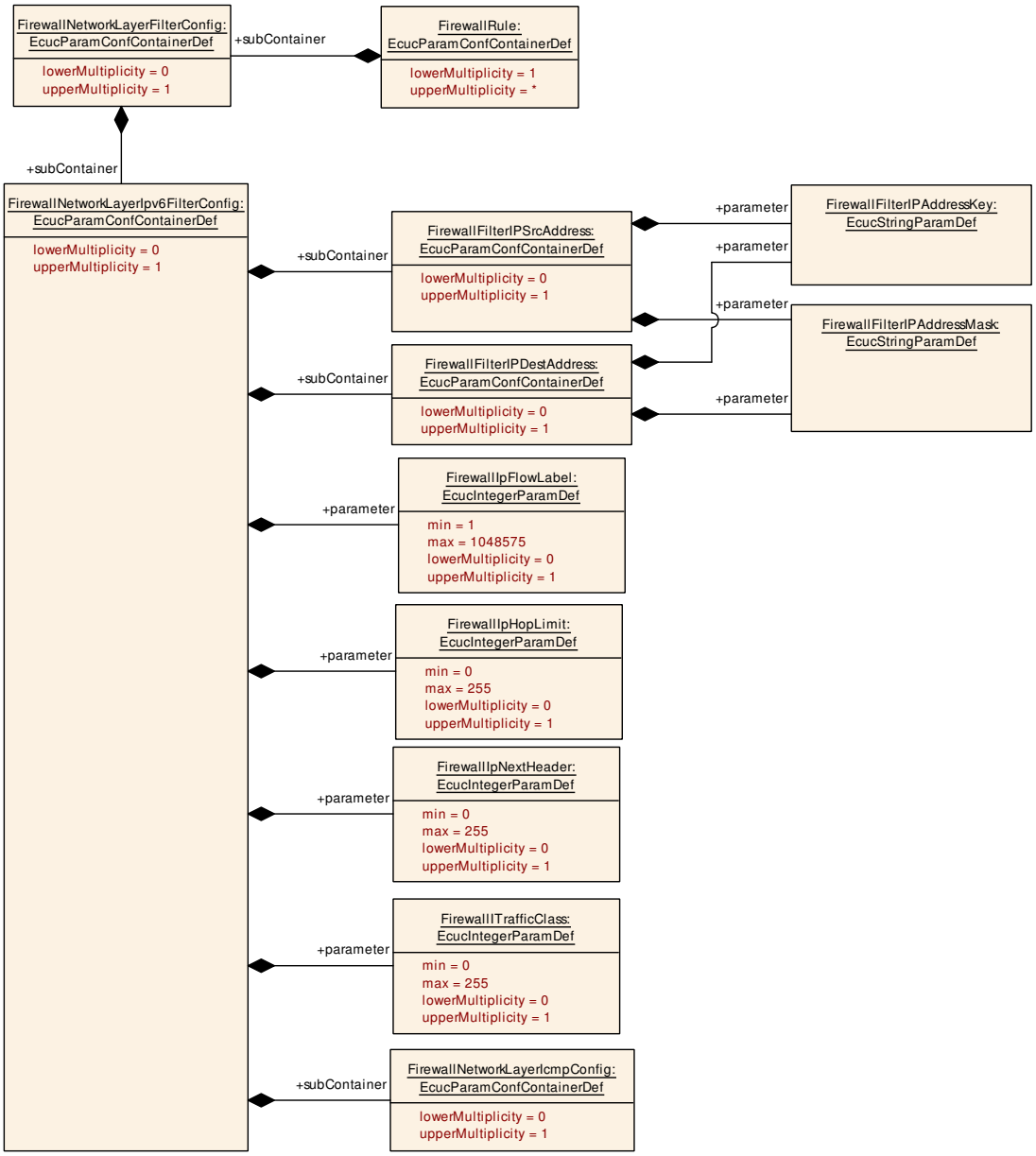



**Figure 10.8: IPv6 configuration**

## [ECUC_Fw_00049] Definition of EcucParamConfContainerDef FirewallNetworkLayerIpv6FilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallNetworkLayerIpv6FilterConfig | |
|---|---|---|
| Parent Container | FirewallNetworkLayerFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules on the Network layer<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallIpFlowLabel | 0..1 | [ECUC_Fw_00051] |
| FirewallIpHopLimit | 0..1 | [ECUC_Fw_00052] |
| FirewallIpNextHeader | 0..1 | [ECUC_Fw_00055] |
| FirewallITrafficClass | 0..1 | [ECUC_Fw_00056] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallFilterIPDestAddress | 0..1 | Configuration of one IP destination filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterIPSrcAddress | 0..1 | Configuration of one IP source filter.<br>**Tags:** atp.Status=draft |
| FirewallNetworkLayerIcmpConfig | 0..1 | Configuration of filter rules for ICMP (Internet Control Message Protocol).<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00051] Definition of EcucIntegerParamDef FirewallIpFlowLabel

*Status:* DRAFT

⌈

| Parameter Name | FirewallIpFlowLabel | |
|---|---|---|
| Parent Container | FirewallNetworkLayerIpv6FilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match packets with a defined flow label.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 1 .. 1048575 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00052] Definition of EcucIntegerParamDef FirewallIpHopLimit

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallIpHopLimit | |
| **Parent Container** | FirewallNetworkLayerIpv6FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a minimum hop limit.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00055] Definition of EcucIntegerParamDef FirewallIpNextHeader

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallIpNextHeader | |
| **Parent Container** | FirewallNetworkLayerIpv6FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a defined type of an extension header.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00056] Definition of EcucIntegerParamDef FirewallITrafficClass

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallITrafficClass | |
| **Parent Container** | FirewallNetworkLayerIpv6FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with a defined traffic class or priority.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |

▽

△

| | | |
|---|---|---|
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.4 ICMP configuration

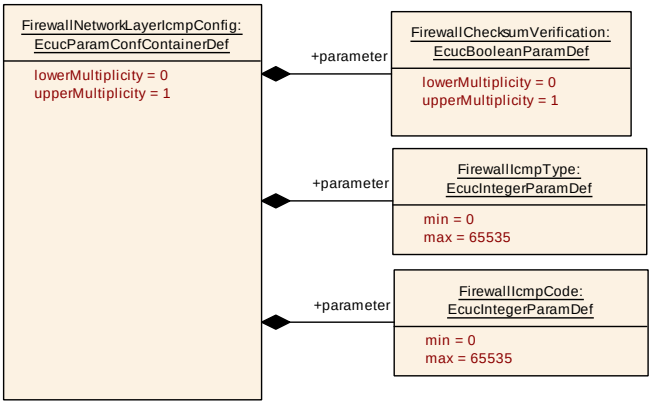



**Figure 10.9: ICMP configuration**

### [ECUC_Fw_00130] Definition of EcucParamConfContainerDef FirewallNetworkLayerIcmpConfig

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallNetworkLayerIcmpConfig | |
| **Parent Container** | FirewallNetworkLayerIpv4FilterConfig, FirewallNetworkLayerIpv6FilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Configuration of filter rules for ICMP (Internet Control Message Protocol).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallChecksumVerification | 0..1 | [ECUC_Fw_00025] |
| FirewallIcmpCode | 1 | [ECUC_Fw_00132] |
| FirewallIcmpType | 1 | [ECUC_Fw_00131] |

| **No Included Containers** |
|---|

⌋

### [ECUC_Fw_00025] Definition of EcucBooleanParamDef FirewallChecksumVerification

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallChecksumVerification | |
| **Parent Container** | FirewallNetworkLayerIcmpConfig, FirewallNetworkLayerIpv4FilterConfig, FirewallTransportLayerTcpFilterConfig, FirewallTransportLayerUdpFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Defines whether checksum verification is performed or not.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00132] Definition of EcucIntegerParamDef FirewallIcmpCode

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallIcmpCode | |
| **Parent Container** | FirewallNetworkLayerIcmpConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match packets with the Icmp code.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00131] Definition of EcucIntegerParamDef FirewallIcmpType

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallIcmpType |
| **Parent Container** | FirewallNetworkLayerIcmpConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |
| **Description** | Filter to match packets with the Icmp type.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 1 |
| **Type** | EcucIntegerParamDef |

▽

△

| | | |
|---|---|---|
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.5 TCP configuration

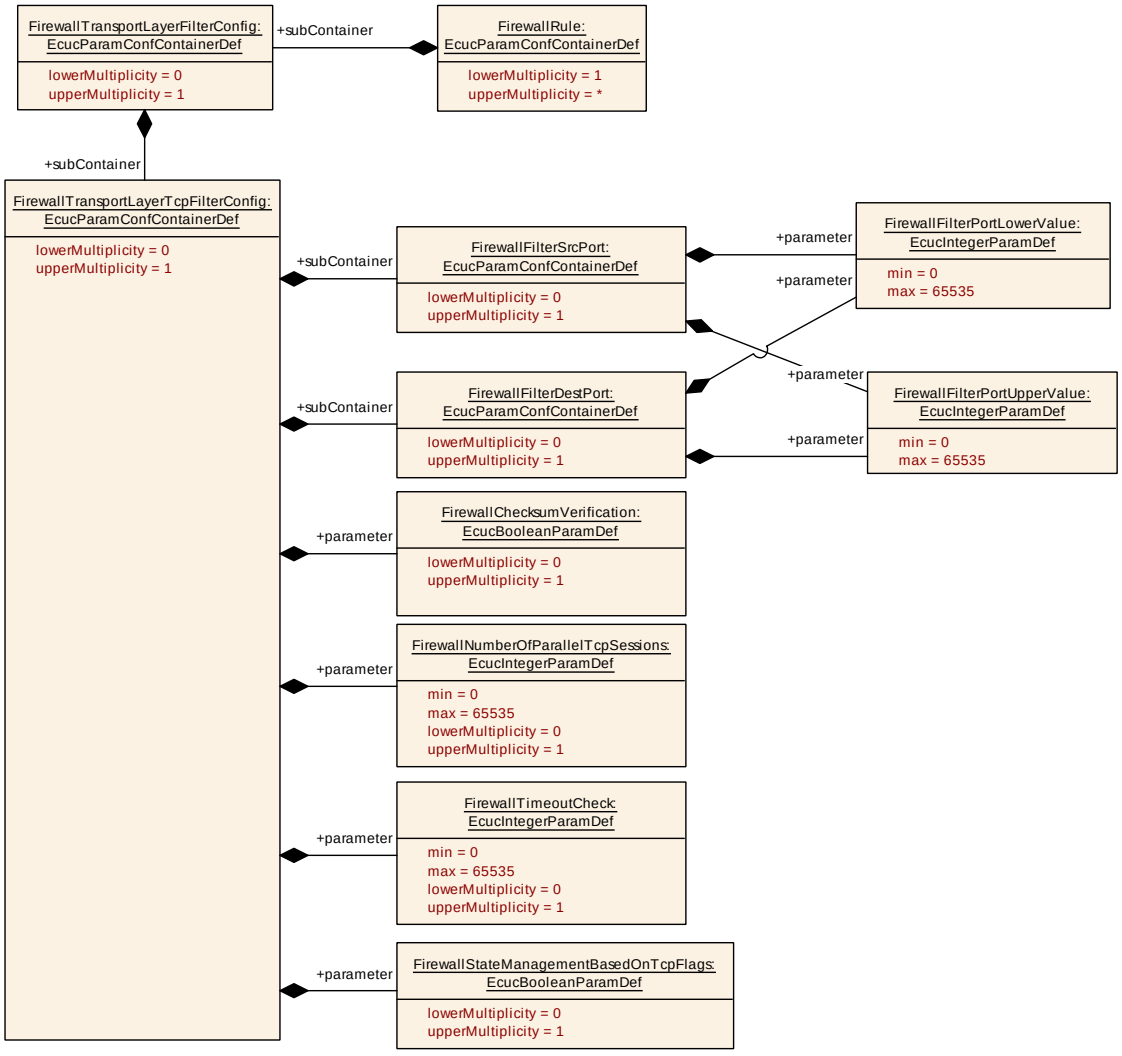



**Figure 10.10: TCP configuration**

### [ECUC_Fw_00138] Definition of EcucParamConfContainerDef FirewallTransportLayerFilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallTransportLayerFilterConfig | |
|---|---|---|
| Parent Container | FirewallRule | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules on Transport Layer level.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| No Included Parameters |
|---|

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallTransportLayerTcpFilterConfig | 0..1 | Configuration of filter rules for TCP on Transport Layer level.<br>**Tags:** atp.Status=draft |
| FirewallTransportLayerUdpFilterConfig | 0..1 | Configuration of filter rules for UDP on Transport Layer level.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00019] Definition of EcucParamConfContainerDef FirewallTransportLayerTcpFilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallTransportLayerTcpFilterConfig | |
|---|---|---|
| Parent Container | FirewallTransportLayerFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules for TCP on Transport Layer level.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallChecksumVerification | 0..1 | [ECUC_Fw_00025] |
| FirewallNumberOfParallelTcpSessions | 0..1 | [ECUC_Fw_00035] |
| FirewallStateManagementBasedOnTcpFlags | 0..1 | [ECUC_Fw_00037] |
| FirewallTimeoutCheck | 0..1 | [ECUC_Fw_00036] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallFilterDestPort | 0..1 | Configuration of a destination port filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterSrcPort | 0..1 | Configuration of a source port filter.<br>**Tags:** atp.Status=draft |

⌋

For parameter table [ECUC_Fw_00025] FirewallChecksumVerification, see definition below container FirewallNetworkLayerIcmpConfig.

### [ECUC_Fw_00035] Definition of EcucIntegerParamDef FirewallNumberOfParallelTcpSessions

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallNumberOfParallelTcpSessions | |
| **Parent Container** | FirewallTransportLayerTcpFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This parameter defines the maximal number of TCP Sessions that are allowed to be established.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00037] Definition of EcucBooleanParamDef FirewallStateManagementBasedOnTcpFlags

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallStateManagementBasedOnTcpFlags | |
| **Parent Container** | FirewallTransportLayerTcpFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This attribute defines whether the StateManagement is based on TCP flags or not.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00036] Definition of EcucIntegerParamDef FirewallTimeoutCheck

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallTimeoutCheck | |
| **Parent Container** | FirewallTransportLayerTcpFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This parameter defines the TCP Session timeout in seconds<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00022] Definition of EcucParamConfContainerDef FirewallFilterDestPort

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallFilterDestPort | |
| **Parent Container** | FirewallTransportLayerTcpFilterConfig, FirewallTransportLayerUdpFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Configuration of a destination port filter.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterPortLowerValue | 1 | [ECUC_Fw_00028] |
| FirewallFilterPortUpperValue | 1 | [ECUC_Fw_00029] |

| **No Included Containers** |
|---|

⌋

### [ECUC_Fw_00028] Definition of EcucIntegerParamDef FirewallFilterPortLowerValue

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterPortLowerValue | |
| **Parent Container** | FirewallFilterDestPort, FirewallFilterSrcPort | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the filter port lower value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00029] Definition of EcucIntegerParamDef FirewallFilterPortUpperValue

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallFilterPortUpperValue | |
| **Parent Container** | FirewallFilterDestPort, FirewallFilterSrcPort | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Definition of the filter port upper value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00020] Definition of EcucParamConfContainerDef FirewallFilterSrcPort

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallFilterSrcPort |
| **Parent Container** | FirewallTransportLayerTcpFilterConfig, FirewallTransportLayerUdpFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |
| **Description** | Configuration of a source port filter.<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallFilterPortLowerValue | 1 | [ECUC_Fw_00028] |
| FirewallFilterPortUpperValue | 1 | [ECUC_Fw_00029] |

| **No Included Containers** |
|---|

⌋

For parameter table [ECUC_Fw_00028] FirewallFilterPortLowerValue, see definition below container FirewallFilterDestPort.

For parameter table [ECUC_Fw_00029] FirewallFilterPortUpperValue, see definition below container FirewallFilterDestPort.

### 10.2.4.6 UDP configuration

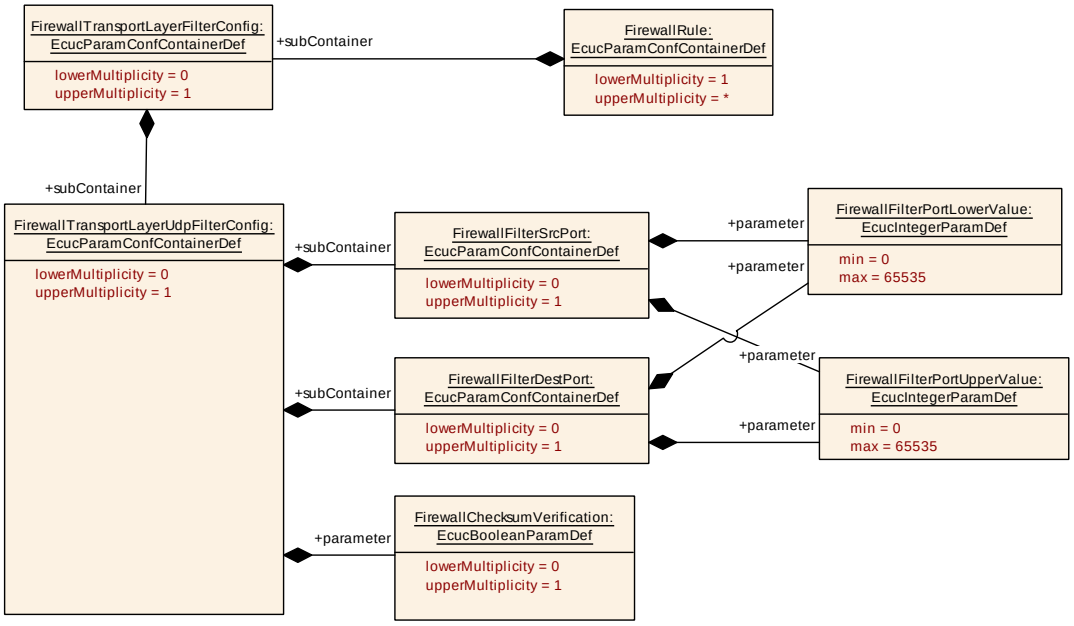



**Figure 10.11: UDP configuration**

**[ECUC_Fw_00038] Definition of EcucParamConfContainerDef FirewallTransportLayerUdpFilterConfig**

*Status:* DRAFT

⌈

| Container Name | FirewallTransportLayerUdpFilterConfig | |
|---|---|---|
| Parent Container | FirewallTransportLayerFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules for UDP on Transport Layer level.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallChecksumVerification | 0..1 | [ECUC_Fw_00025] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallFilterDestPort | 0..1 | Configuration of a destination port filter.<br>**Tags:** atp.Status=draft |
| FirewallFilterSrcPort | 0..1 | Configuration of a source port filter.<br>**Tags:** atp.Status=draft |

⌋

For parameter table [ECUC_Fw_00025] FirewallChecksumVerification, see definition below container FirewallNetworkLayerIcmpConfig.

#### 10.2.4.7 SOME/IP configuration

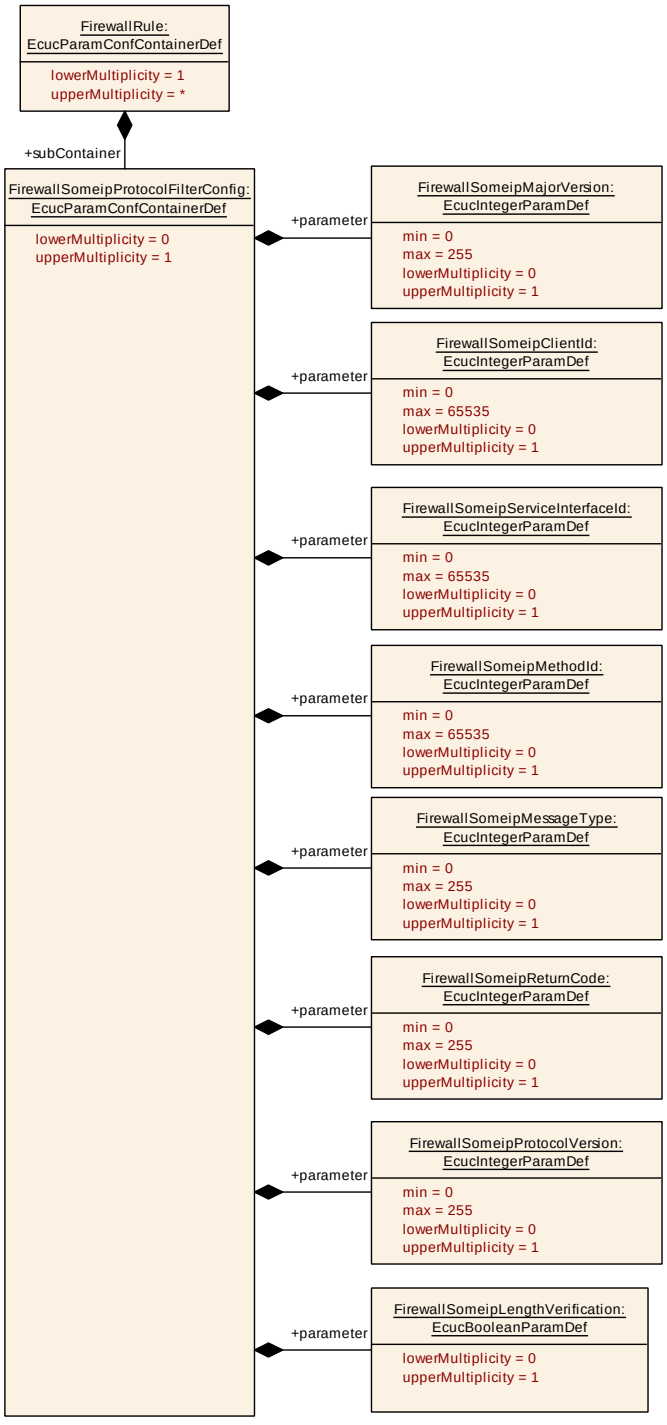



**Figure 10.12: SOME/IP configuration**

### [ECUC_Fw_00068] Definition of EcucParamConfContainerDef FirewallSomeip ProtocolFilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallSomeipProtocolFilterConfig | |
|---|---|---|
| Parent Container | FirewallRule | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of SOME/IP Protocol firewall rules<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallSomeipClientId | 0..1 | [ECUC_Fw_00070] |
| FirewallSomeipLengthVerification | 0..1 | [ECUC_Fw_00075] |
| FirewallSomeipMajorVersion | 0..1 | [ECUC_Fw_00069] |
| FirewallSomeipMessageType | 0..1 | [ECUC_Fw_00072] |
| FirewallSomeipMethodId | 0..1 | [ECUC_Fw_00071] |
| FirewallSomeipProtocolVersion | 0..1 | [ECUC_Fw_00074] |
| FirewallSomeipReturnCode | 0..1 | [ECUC_Fw_00073] |
| FirewallSomeipServiceInterfaceId | 0..1 | [ECUC_Fw_00065] |

| No Included Containers |
|---|

⌋

## [ECUC_Fw_00070] Definition of EcucIntegerParamDef FirewallSomeipClientId

*Status:* DRAFT

⌈

| Parameter Name | FirewallSomeipClientId | |
|---|---|---|
| Parent Container | FirewallSomeipProtocolFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter for SOME/IP messages in which the clientId in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 65535 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00075] Definition of EcucBooleanParamDef FirewallSomeipLength Verification

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipLengthVerification | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Defines whether length verification is performed or not.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucBooleanParamDef | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00069] Definition of EcucIntegerParamDef FirewallSomeipMajorVersion

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipMajorVersion | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP messages in which the majorVersion in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00072] Definition of EcucIntegerParamDef FirewallSomeipMessageType

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallSomeipMessageType |
| **Parent Container** | FirewallSomeipProtocolFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Filter for SOME/IP messages in which the message type in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00071] Definition of EcucIntegerParamDef FirewallSomeipMethodId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipMethodId | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP messages in which the methodId in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00074] Definition of EcucIntegerParamDef FirewallSomeipProtocolVersion

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipProtocolVersion | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP messages in which the protocol version in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |

▽

△

| | | |
|---|---|---|
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00073] Definition of EcucIntegerParamDef FirewallSomeipReturnCode

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipReturnCode | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP messages in which the return code in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00065] Definition of EcucIntegerParamDef FirewallSomeipServiceInterfaceId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipServiceInterfaceId | |
| **Parent Container** | FirewallSomeipProtocolFilterConfig, FirewallSomeipSdFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP messages in which the serviceInterfaceId in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00058] Definition of EcucParamConfContainerDef FirewallSomeipMajorVersion

*Status:* DRAFT

⌈

| Container Name | FirewallSomeipMajorVersion | |
|---|---|---|
| Parent Container | FirewallSomeipSdFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is in the configured max and min value.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallMajorVersionMaxValue | 0..1 | [ECUC_Fw_00060] |
| FirewallMajorVersionMinValue | 0..1 | [ECUC_Fw_00061] |

| No Included Containers |
|---|

⌋

## [ECUC_Fw_00060] Definition of EcucIntegerParamDef FirewallMajorVersionMaxValue

*Status:* DRAFT

⌈

| Parameter Name | FirewallMajorVersionMaxValue | |
|---|---|---|
| Parent Container | FirewallSomeipMajorVersion | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is smaller or equal than this value.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 255 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

## [ECUC_Fw_00061] Definition of EcucIntegerParamDef FirewallMajorVersionMinValue

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallMajorVersionMinValue | |
| **Parent Container** | FirewallSomeipMajorVersion | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is greater or equal than this value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00062] Definition of EcucParamConfContainerDef FirewallSomeipMinorVersion

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallSomeipMinorVersion | |
| **Parent Container** | FirewallSomeipSdFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is in the configured max and min value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallMinorVersionMaxValue | 0..1 | [ECUC_Fw_00063] |
| FirewallMinorVersionMinValue | 0..1 | [ECUC_Fw_00064] |

| **No Included Containers** |
|---|

⌋

### [ECUC_Fw_00063] Definition of EcucIntegerParamDef FirewallMinorVersionMax Value

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallMinorVersionMaxValue | |
| **Parent Container** | FirewallSomeipMinorVersion | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is smaller or equal than than this value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967294 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00064] Definition of EcucIntegerParamDef FirewallMinorVersionMin Value

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallMinorVersionMinValue | |
| **Parent Container** | FirewallSomeipMinorVersion | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is greater or equal than this value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967294 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.8 SOME/IP-SD configuration

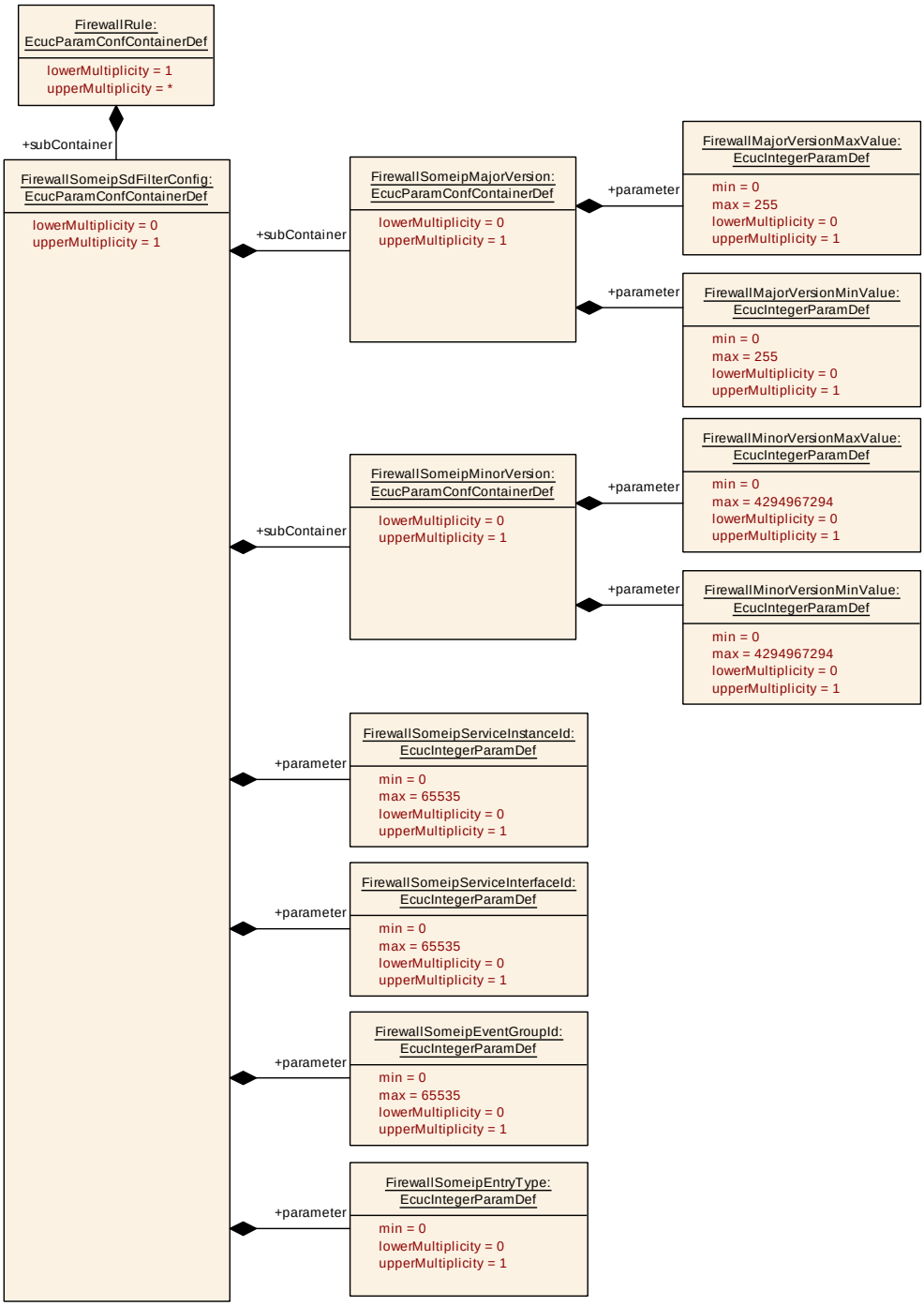



**Figure 10.13: SOME/IP-SD configuration**

## [ECUC_Fw_00057] Definition of EcucParamConfContainerDef FirewallSomeipSd FilterConfig

*Status:* DRAFT

⌈

| Container Name | FirewallSomeipSdFilterConfig | |
|---|---|---|
| Parent Container | FirewallRule | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of SOME/IP Service Discovery firewall rules<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallSomeipEntryType | 0..1 | [ECUC_Fw_00067] |
| FirewallSomeipEventGroupId | 0..1 | [ECUC_Fw_00066] |
| FirewallSomeipServiceInstanceId | 0..1 | [ECUC_Fw_00059] |
| FirewallSomeipServiceInterfaceId | 0..1 | [ECUC_Fw_00065] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallSomeipMajorVersion | 0..1 | Filter for SOME/IP SD messages in which the MajorVersion in the SOME/IP header is in the configured max and min value.<br>**Tags:** atp.Status=draft |
| FirewallSomeipMinorVersion | 0..1 | Filter for SOME/IP SD messages in which the MinorVersion in the SOME/IP header is in the configured max and min value.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00067] Definition of EcucIntegerParamDef FirewallSomeipEntryType

*Status:* DRAFT

⌈

| Parameter Name | FirewallSomeipEntryType | |
|---|---|---|
| Parent Container | FirewallSomeipSdFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter for SOME/IP SD messages in which the entryType in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 18446744073709551615 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00066] Definition of EcucIntegerParamDef FirewallSomeipEventGroupId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipEventGroupId | |
| **Parent Container** | FirewallSomeipSdFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the eventGroupId in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00059] Definition of EcucIntegerParamDef FirewallSomeipServiceInstanceId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallSomeipServiceInstanceId | |
| **Parent Container** | FirewallSomeipSdFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for SOME/IP SD messages in which the serviceInstanceId in the SOME/IP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

For parameter table [ECUC_Fw_00065] FirewallSomeipServiceInterfaceId, see definition below container FirewallSomeipProtocolFilterConfig.

### 10.2.4.9 DDS configuration

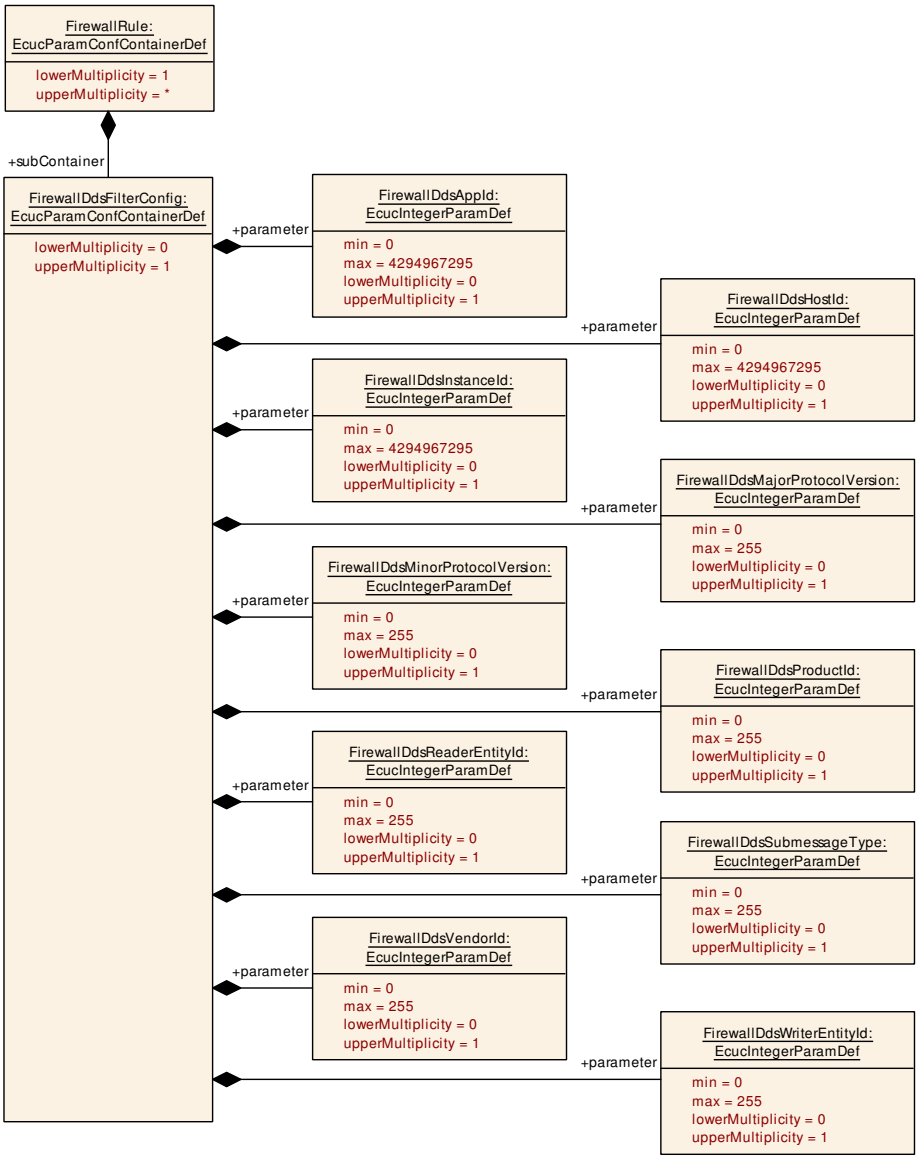



**Figure 10.14: DDS configuration**

## [ECUC_Fw_00092] Definition of EcucParamConfContainerDef FirewallDdsFilter Config

*Status:* DRAFT

⌈

| Container Name | FirewallDdsFilterConfig |
|---|---|
| Parent Container | FirewallRule |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Configuration of filter rules for Dds<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDdsAppId | 0..1 | [ECUC_Fw_00093] |
| FirewallDdsHostId | 0..1 | [ECUC_Fw_00094] |
| FirewallDdsInstanceId | 0..1 | [ECUC_Fw_00095] |
| FirewallDdsMajorProtocolVersion | 0..1 | [ECUC_Fw_00096] |
| FirewallDdsMinorProtocolVersion | 0..1 | [ECUC_Fw_00097] |
| FirewallDdsProductId | 0..1 | [ECUC_Fw_00098] |
| FirewallDdsReaderEntityId | 0..1 | [ECUC_Fw_00099] |
| FirewallDdsSubmessageType | 0..1 | [ECUC_Fw_00100] |
| FirewallDdsVendorId | 0..1 | [ECUC_Fw_00101] |
| FirewallDdsWriterEntityId | 0..1 | [ECUC_Fw_00102] |

| **No Included Containers** |
|---|

⌋

## [ECUC_Fw_00093] Definition of EcucIntegerParamDef FirewallDdsAppId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsAppId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the appId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967295 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00094] Definition of EcucIntegerParamDef FirewallDdsHostId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsHostId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the hostId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967295 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00095] Definition of EcucIntegerParamDef FirewallDdsInstanceId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsInstanceId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the instanceId in the DDSI-RTPS header and the INFO_DST (0x0E) submessage matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967295 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00096] Definition of EcucIntegerParamDef FirewallDdsMajorProtocolVersion

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallDdsMajorProtocolVersion |
| **Parent Container** | FirewallDdsFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Filter for DDSI-RTPS messages in which the majorProtocolVersion in the DDSI-RTPS header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00097] Definition of EcucIntegerParamDef FirewallDdsMinorProtocolVersion

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsMinorProtocolVersion | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the minorProtocolVersion in the DDSI-RTPS header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00098] Definition of EcucIntegerParamDef FirewallDdsProductId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsProductId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the productId in the DDSI-RTPS header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |

▽

△

| | | |
|---|---|---|
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00099] Definition of EcucIntegerParamDef FirewallDdsReaderEntityId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsReaderEntityId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the readerEntityID in a DDSI-RTPS submessage matches<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00100] Definition of EcucIntegerParamDef FirewallDdsSubmessageType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsSubmessageType | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Defines the allowed submessage type in the DDSI-RTPS message<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00101] Definition of EcucIntegerParamDef FirewallDdsVendorId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsVendorId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the vendorId in the DDSI-RTPS header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00102] Definition of EcucIntegerParamDef FirewallDdsWriterEntityId

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDdsWriterEntityId | |
| **Parent Container** | FirewallDdsFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter for DDSI-RTPS messages in which the writerEntityID in a DDSI-RTPS submessage matches<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.10 DoIP configuration

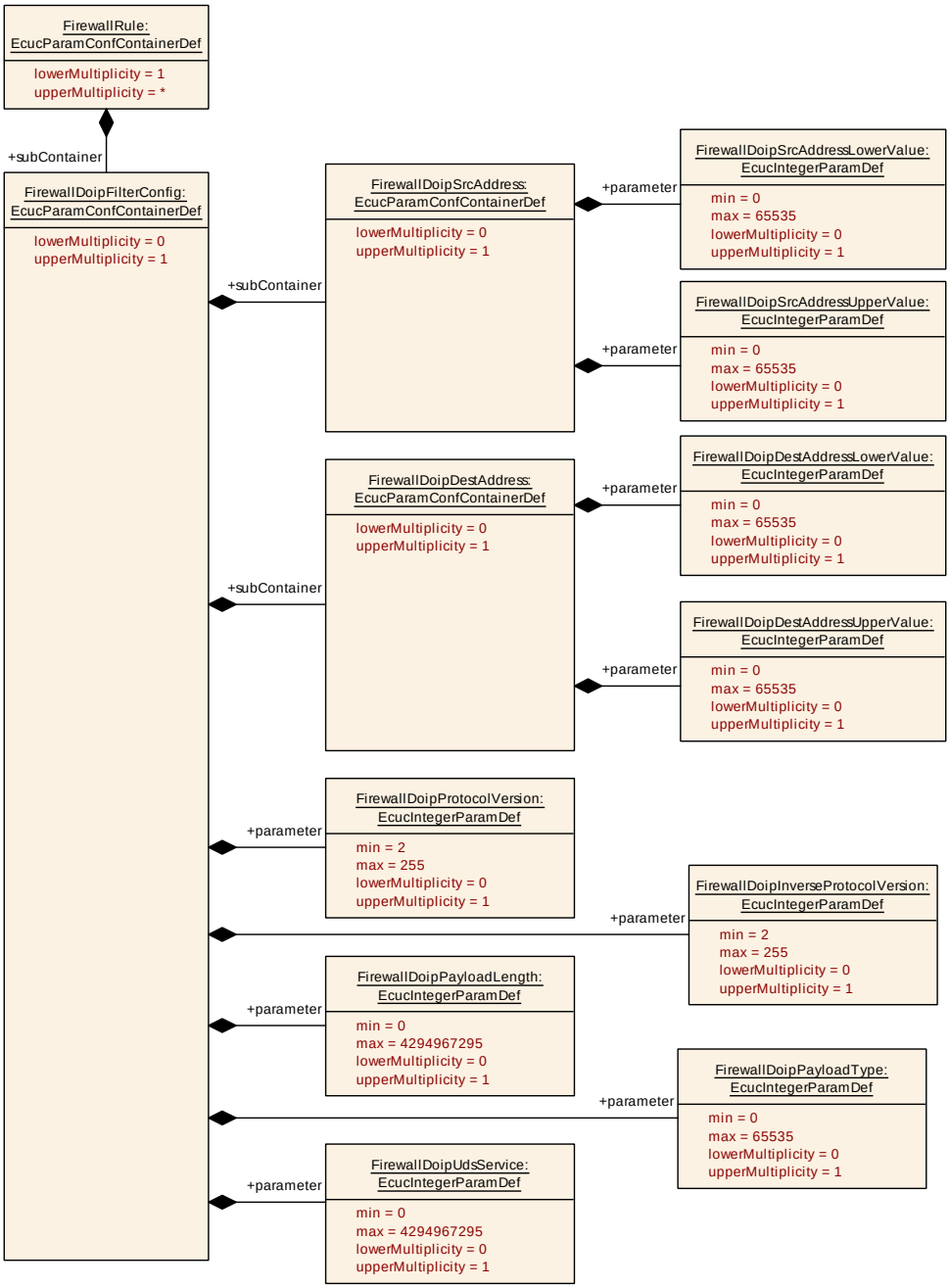



**Figure 10.15: DoIP configuration**

## [ECUC_Fw_00079] Definition of EcucParamConfContainerDef FirewallDoipFilter Config

*Status:* DRAFT

⌈

| Container Name | FirewallDoipFilterConfig | |
|---|---|---|
| Parent Container | FirewallRule | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of filter rules for DoIP<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDoipInverseProtocolVersion | 0..1 | [ECUC_Fw_00088] |
| FirewallDoipPayloadLength | 0..1 | [ECUC_Fw_00089] |
| FirewallDoipPayloadType | 0..1 | [ECUC_Fw_00090] |
| FirewallDoipProtocolVersion | 0..1 | [ECUC_Fw_00087] |
| FirewallDoipUdsService | 0..1 | [ECUC_Fw_00091] |

| Included Containers | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallDoipDestAddress | 0..1 | Configuration of a source port filter.<br>**Tags:** atp.Status=draft |
| FirewallDoipSrcAddress | 0..1 | Configuration of a source port filter.<br>**Tags:** atp.Status=draft |

⌋

## [ECUC_Fw_00088] Definition of EcucIntegerParamDef FirewallDoipInverseProtocolVersion

*Status:* DRAFT

⌈

| Parameter Name | FirewallDoipInverseProtocolVersion | |
|---|---|---|
| Parent Container | FirewallDoipFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match DoIP messages in which the inverseprotocolVersion in the DoIP header matches.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 2 .. 255 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00089] Definition of EcucIntegerParamDef FirewallDoipPayloadLength

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDoipPayloadLength | |
| **Parent Container** | FirewallDoipFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match DoIP messages in which the payloadLength in the DoIP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967295 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00090] Definition of EcucIntegerParamDef FirewallDoipPayloadType

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDoipPayloadType | |
| **Parent Container** | FirewallDoipFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match DoIP messages in which the payloadType in the DoIP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00087] Definition of EcucIntegerParamDef FirewallDoipProtocolVersion

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallDoipProtocolVersion |
| **Parent Container** | FirewallDoipFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Filter to match DoIP messages in which the protocolVersion in the DoIP header matches.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 2 .. 255 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00091] Definition of EcucIntegerParamDef FirewallDoipUdsService

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallDoipUdsService | |
| **Parent Container** | FirewallDoipFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Filter to match DoIP messages that contain the udsService.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 4294967295 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

## [ECUC_Fw_00082] Definition of EcucParamConfContainerDef FirewallDoipDestAddress

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallDoipDestAddress | |
| **Parent Container** | FirewallDoipFilterConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Configuration of a source port filter.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDoipDestAddressLowerValue | 0..1 | [ECUC_Fw_00085] |
| FirewallDoipDestAddressUpperValue | 0..1 | [ECUC_Fw_00086] |

| No Included Containers |
|---|

⌋

### [ECUC_Fw_00085] Definition of EcucIntegerParamDef FirewallDoipDestAddress LowerValue

*Status:* DRAFT

⌈

| Parameter Name | FirewallDoipDestAddressLowerValue | |
|---|---|---|
| Parent Container | FirewallDoipDestAddress | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match DoIP messages in which the destinationAddress is greater or equal than FirwallDoipDestAddressLowerValue<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 65535 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00086] Definition of EcucIntegerParamDef FirewallDoipDestAddress UpperValue

*Status:* DRAFT

⌈

| Parameter Name | FirewallDoipDestAddressUpperValue | |
|---|---|---|
| Parent Container | FirewallDoipDestAddress | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match DoIP messages in which the destinationAddress is smaller or equal than FirewallDoipDestAddressUpperValue<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 65535 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

### [ECUC_Fw_00081] Definition of EcucParamConfContainerDef FirewallDoipSrc Address

*Status:* DRAFT

⌈

| Container Name | FirewallDoipSrcAddress | |
|---|---|---|
| Parent Container | FirewallDoipFilterConfig | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Configuration of a source port filter.<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Post-Build Multiplicity | Loadable | Selectable |
| Configuration Parameters | | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallDoipSrcAddressLowerValue | 0..1 | [ECUC_Fw_00083] |
| FirewallDoipSrcAddressUpperValue | 0..1 | [ECUC_Fw_00084] |

| No Included Containers |
|---|

⌋

## [ECUC_Fw_00083] Definition of EcucIntegerParamDef FirewallDoipSrcAddress LowerValue

*Status:* DRAFT

⌈

| Parameter Name | FirewallDoipSrcAddressLowerValue | |
|---|---|---|
| Parent Container | FirewallDoipSrcAddress | |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] | |
| Description | Filter to match DoIP messages in which the sourceAddress is greater or equal than FirwallDoipDestAddressLowerValue<br>**Tags:** atp.Status=draft | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 0 .. 65535 | |
| Default value | – | |
| Post-Build Multiplicity | Loadable | Selectable |
| Post-Build Value | Loadable | Selectable |

⌋

## [ECUC_Fw_00084] Definition of EcucIntegerParamDef FirewallDoipSrcAddress UpperValue

*Status:* DRAFT

⌈

| Parameter Name | FirewallDoipSrcAddressUpperValue |
|---|---|
| Parent Container | FirewallDoipSrcAddress |
| Upstream mapping defined | [2, CP_TPS_SystemTemplate] |

▽

△

| | | |
|---|---|---|
| **Description** | Definition of the filter port upper value.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 65535 | |
| **Default value** | – | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### 10.2.4.11 Payload Byte Pattern configuration

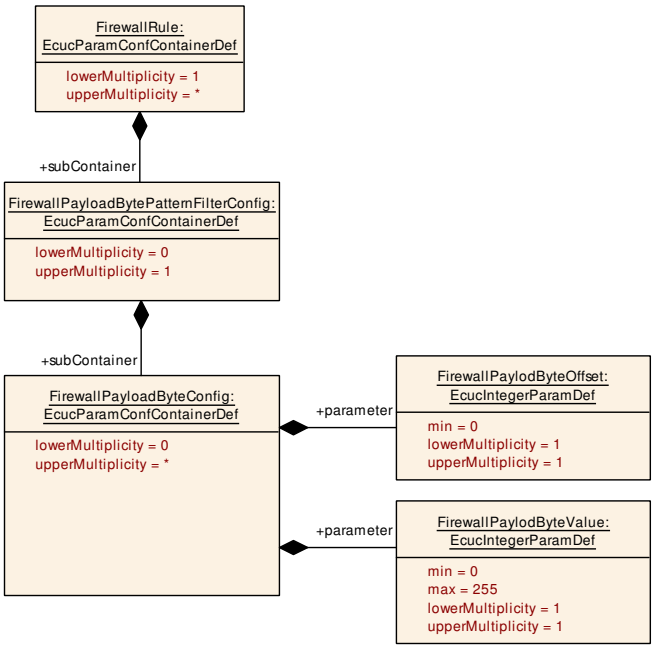



**Figure 10.16: Payload Byte Pattern configuration**

## [ECUC_Fw_00077] Definition of EcucParamConfContainerDef FirewallPayloadByteConfig

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallPayloadByteConfig |
| **Parent Container** | FirewallPayloadBytePatternFilterConfig |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] |
| **Description** | Configuration of a single byte in the datagram.<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| **Multiplicity** | 0..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallPaylodByteOffset | 1 | [ECUC_Fw_00078] |
| FirewallPaylodByteValue | 1 | [ECUC_Fw_00080] |

| **No Included Containers** |
|---|

⌋

## [ECUC_Fw_00078] Definition of EcucIntegerParamDef FirewallPaylodByteOffset

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallPaylodByteOffset | |
| **Parent Container** | FirewallPayloadByteConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This parameter defines the byte offset in the datagram (start byte of the Ethernet frame, i.e. offset 0 corresponds to the first byte of the destination MAC address).<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 18446744073709551615 | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

## [ECUC_Fw_00080] Definition of EcucIntegerParamDef FirewallPaylodByteValue

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallPaylodByteValue | |
| **Parent Container** | FirewallPayloadByteConfig | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | This attribute defines the byteValue in the datagram.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 255 | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

## [ECUC_Fw_00076] Definition of EcucParamConfContainerDef FirewallPayloadBytePatternFilterConfig

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallPayloadBytePatternFilterConfig | |
| **Parent Container** | FirewallRule | |
| **Upstream mapping defined** | [2, CP_TPS_SystemTemplate] | |
| **Description** | Configuration of a generic firewall rule that defines the individual bytes of a message that shall match.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..1 | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **No Included Parameters** |
|---|

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Description** |
| FirewallPayloadByteConfig | 0..* | Configuration of a single byte in the datagram.<br>**Tags:** atp.Status=draft |

⌋

### 10.2.5 Switch bucket counting mechanism

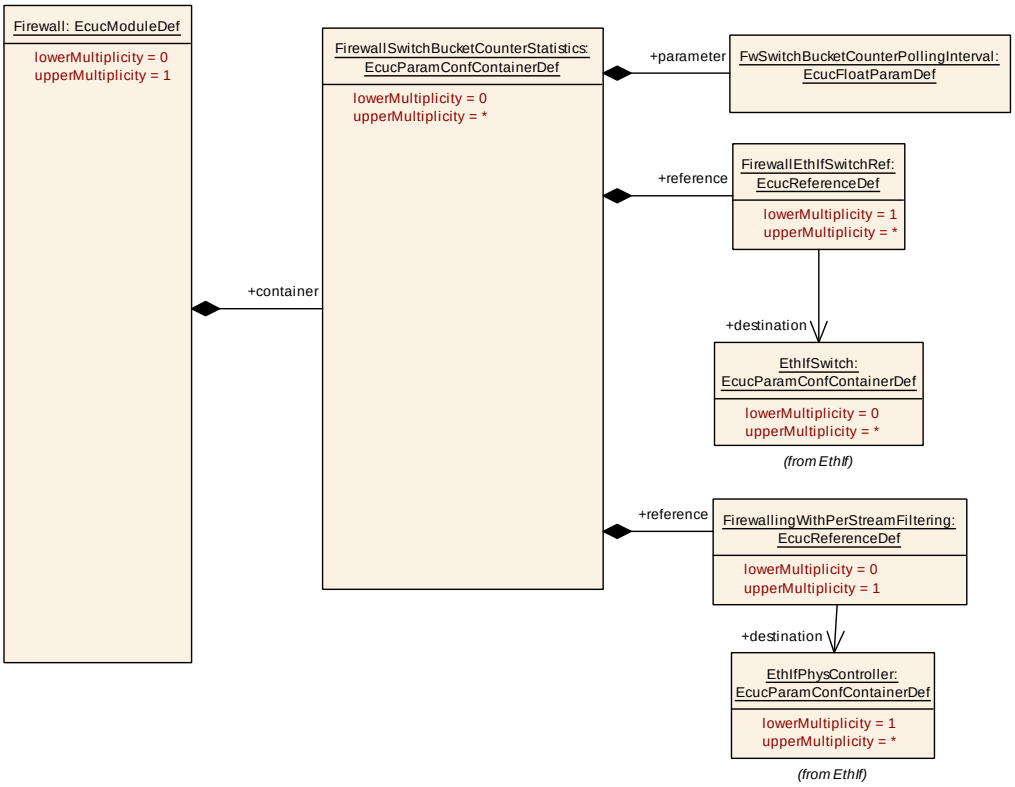



**Figure 10.17: Switch bucket counting mechanism**

## [ECUC_Fw_00134] Definition of EcucParamConfContainerDef FirewallSwitchBucketCounterStatistics

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Container Name** | FirewallSwitchBucketCounterStatistics | |
| **Parent Container** | Firewall | |
| **Upstream mapping defined** | – | |
| **Description** | Polling of switch bucket counter statistics<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 0..* | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Configuration Parameters** | | |

| **Included Parameters** | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FwSwitchBucketCounterPollingInterval | 1 | [ECUC_Fw_00135] |
| FirewallEthIfSwitchRef | 1..* | [ECUC_Fw_00136] |
| FirewallingWithPerStreamFiltering | 0..1 | [ECUC_Fw_00137] |

| **No Included Containers** |
|---|

⌋

### [ECUC_Fw_00135] Definition of EcucFloatParamDef FwSwitchBucketCounterPollingInterval

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FwSwitchBucketCounterPollingInterval | |
| **Parent Container** | FirewallSwitchBucketCounterStatistics | |
| **Upstream mapping defined** | – | |
| **Description** | Length of the switch bucket counter polling time interval (in seconds). Note: Shall be configured as a multiple of the IdsM main function period.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1 | |
| **Type** | EcucFloatParamDef | |
| **Range** | [-INF .. INF] | |
| **Default value** | – | |
| **Pre-Compile only** | true | |

⌋

### [ECUC_Fw_00136] Definition of EcucReferenceDef FirewallEthIfSwitchRef

*Status:* DRAFT

⌈

| | | |
|---|---|---|
| **Parameter Name** | FirewallEthIfSwitchRef | |
| **Parent Container** | FirewallSwitchBucketCounterStatistics | |
| **Upstream mapping defined** | – | |
| **Description** | Reference to EthIfSwitch for which the bucket counter statistics applies.<br>**Tags:** atp.Status=draft | |
| **Multiplicity** | 1..* | |
| **Type** | Reference to EthIfSwitch | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

⌋

### [ECUC_Fw_00137] Definition of EcucReferenceDef FirewallingWithPerStreamFiltering

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | FirewallingWithPerStreamFiltering |
| **Parent Container** | FirewallSwitchBucketCounterStatistics |
| **Upstream mapping defined** | – |
| **Description** | Reference to EthIfSwitch for which the bucket counter statistics applies.<br>**Tags:** atp.Status=draft |

▽

△

| | | |
|---|---|---|
| **Multiplicity** | 0..1 | |
| **Type** | Reference to EthIfPhysController | |
| **Post-Build Multiplicity** | Loadable | Selectable |
| **Post-Build Value** | Loadable | Selectable |

┘

### 10.2.6 Security Events

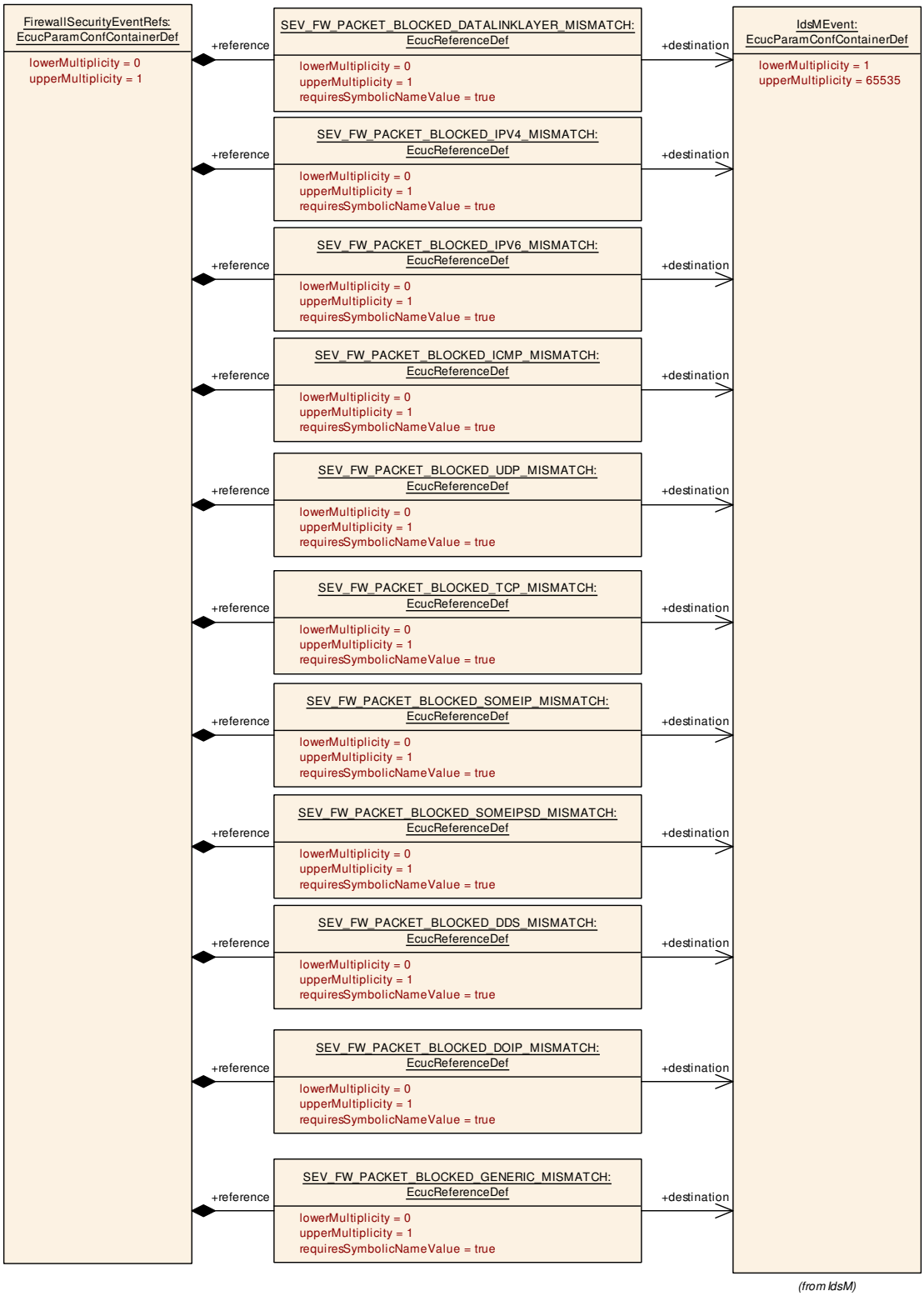



**Figure 10.18: Security Events**

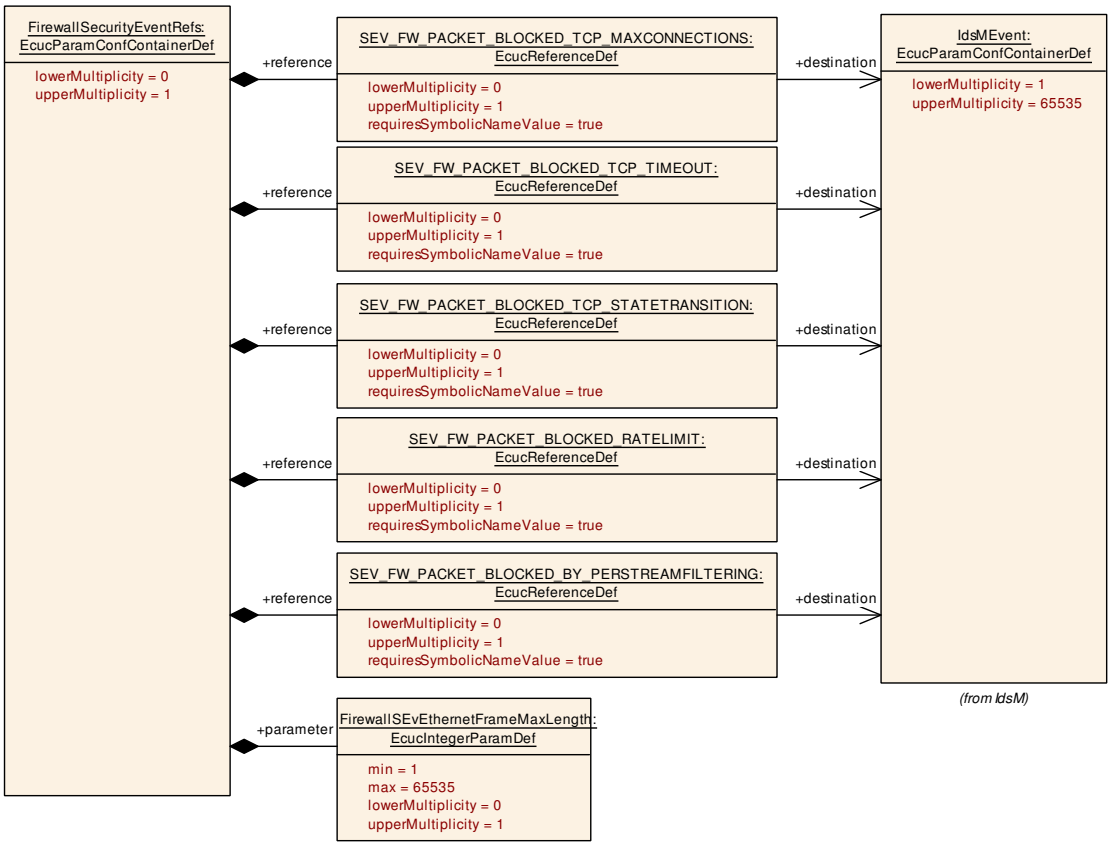



**Figure 10.19: Security Events (cont')**

## [ECUC_Fw_00110] Definition of EcucParamConfContainerDef FirewallSecurity EventRefs

*Status:* DRAFT

⌈

| | |
|---|---|
| **Container Name** | FirewallSecurityEventRefs |
| **Parent Container** | FirewallGeneral |
| **Upstream mapping defined** | – |
| **Description** | Container for the references to IdsMEvent elements representing the security events that the Firwall module shall report to the IdsM in case the coresponding security related event occurs (and if FirewallEnableSecurityEventReporting is set to "true"). The standardized security events in this container can be extended by vendor-specific security events.<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Pre-Compile only** | true |
| **Configuration Parameters** | |

| Included Parameters | | |
|---|---|---|
| **Parameter Name** | **Multiplicity** | **ECUC ID** |
| FirewallSEvEthernetFrameMaxLength | 0..1 | [ECUC_Fw_00153] |
| SEV_FW_PACKET_BLOCKED_BY_-PERSTREAMFILTERING | 0..1 | [ECUC_Fw_00129] |
| SEV_FW_PACKET_BLOCKED_DATALINKLAYER_-MISMATCH | 0..1 | [ECUC_Fw_00111] |
| SEV_FW_PACKET_BLOCKED_DDS_MISMATCH | 0..1 | [ECUC_Fw_00123] |
| SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH | 0..1 | [ECUC_Fw_00133] |
| SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH | 0..1 | [ECUC_Fw_00124] |
| SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH | 0..1 | [ECUC_Fw_00115] |
| SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH | 0..1 | [ECUC_Fw_00114] |
| SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH | 0..1 | [ECUC_Fw_00117] |
| SEV_FW_PACKET_BLOCKED_RATELIMIT | 0..1 | [ECUC_Fw_00128] |
| SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH | 0..1 | [ECUC_Fw_00121] |
| SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH | 0..1 | [ECUC_Fw_00122] |
| SEV_FW_PACKET_BLOCKED_TCP_-MAXCONNECTIONS | 0..1 | [ECUC_Fw_00125] |
| SEV_FW_PACKET_BLOCKED_TCP_MISMATCH | 0..1 | [ECUC_Fw_00120] |
| SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION | 0..1 | [ECUC_Fw_00126] |
| SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT | 0..1 | [ECUC_Fw_00127] |
| SEV_FW_PACKET_BLOCKED_UDP_MISMATCH | 0..1 | [ECUC_Fw_00119] |

| No Included Containers |
|---|

⌋

### [ECUC_Fw_00153] Definition of EcucIntegerParamDef FirewallSEvEthernetFrameMaxLength ⌈

| Parameter Name | FirewallSEvEthernetFrameMaxLength | |
|---|---|---|
| Parent Container | FirewallSecurityEventRefs | |
| Upstream mapping defined | [11, FO_TPS_SecurityExtractTemplate] | |
| Description | Specifies the maximum length in bytes of the Ethernet frame contained in the context data of a reported security event. | |
| Multiplicity | 0..1 | |
| Type | EcucIntegerParamDef | |
| Range | 1 .. 65535 | |
| Default value | – | |
| Pre-Compile only | true | |

⌋

### [ECUC_Fw_00129] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_BY_PERSTREAMFILTERING |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to per-stream filtering in the switch<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00111] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_DATALINKLAYER_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch on data link layer<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00123] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_DDS_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_DDS_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch in the DDS-RTPS protocol<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00133] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_DOIP_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch in the DoIP protocol<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00124] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_GENERIC_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch on generic inspection level<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00115] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_ICMP_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch within the ICMP protocol<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00114] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_IPV4_MISMATCH |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to a rule mismatch on IPv4 layer<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00117] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_IPV6_MISMATCH |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to a rule mismatch on IPv6 layer<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00128] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_RATELIMIT

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_RATELIMIT |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to the rate limit was reached<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00121] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_SOMEIP_MISMATCH |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to a rule mismatch in the SOME/IP protocol<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00122] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_SOMEIPSD_MISMATCH |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to a rule mismatch in the SOME/IP SD protocol<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00125] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_TCP_MAXCONNECTIONS |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to the maximal number of open TCP connections was reached<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

### [ECUC_Fw_00120] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_TCP_MISMATCH

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_TCP_MISMATCH |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to a rule mismatch on TCP layer<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00126] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_TCP_STATETRANSITION |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to an invalid TCP state transition<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

### [ECUC_Fw_00127] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT

*Status:* DRAFT

⌈

| Parameter Name | SEV_FW_PACKET_BLOCKED_TCP_TIMEOUT |
|---|---|
| Parent Container | FirewallSecurityEventRefs |
| Upstream mapping defined | – |
| Description | A network packet was blocked due to TCP timeout<br>**Tags:** atp.Status=draft |
| Multiplicity | 0..1 |
| Type | Symbolic name reference to IdsMEvent |
| Pre-Compile only | true |

⌋

**[ECUC_Fw_00119] Definition of EcucReferenceDef SEV_FW_PACKET_BLOCKED_UDP_MISMATCH**

*Status:* DRAFT

⌈

| | |
|---|---|
| **Parameter Name** | SEV_FW_PACKET_BLOCKED_UDP_MISMATCH |
| **Parent Container** | FirewallSecurityEventRefs |
| **Upstream mapping defined** | – |
| **Description** | A network packet was blocked due to a rule mismatch on UDP layer<br>**Tags:** atp.Status=draft |
| **Multiplicity** | 0..1 |
| **Type** | Symbolic name reference to IdsMEvent |
| **Pre-Compile only** | true |

⌋

## 10.3 Published Information

For details refer to [3] Chapter 10.3 *"Published Information"*.

# A Not applicable requirements

There are no not applicable requirements for the firewall.

# B Change history of AUTOSAR traceable items

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

## B.1 Traceable item history of this document according to AUTOSAR Release R26-11

### B.1.1 Added Specification Items in R26-11

none

### B.1.2 Changed Specification Items in R26-11

[CP_SWS_Fw_91010] [CP_SWS_Fw_91012] [ECUC_Fw_00002]

### B.1.3 Deleted Specification Items in R26-11

[CP_SWS_Fw_91004] [ECUC_Fw_00004]

## B.2 Traceable item history of this document according to AUTOSAR Release R25-11

### B.2.1 Added Specification Items in R25-11

[CP_SWS_Fw_60034] [ECUC_Fw_00153]

### B.2.2 Changed Specification Items in R25-11

[CP_SWS_Fw_60001] [CP_SWS_Fw_60002] [CP_SWS_Fw_60003] [CP_SWS_Fw_60020] [CP_SWS_Fw_60021] [CP_SWS_Fw_60022] [CP_SWS_Fw_60023] [CP_SWS_Fw_60024] [CP_SWS_Fw_60025] [CP_SWS_Fw_60026] [CP_SWS_Fw_60027] [CP_SWS_Fw_60028] [CP_SWS_Fw_60029] [CP_SWS_Fw_60030] [CP_SWS_Fw_60031] [CP_SWS_Fw_91010] [CP_SWS_Fw_91012] [ECUC_Fw_00110]

### B.2.3 Deleted Specification Items in R25-11

none

## B.3 Traceable item history of this document according to AUTOSAR Release R24-11

### B.3.1 Added Specification Items in R24-11

[CP_SWS_Fw_30027] [CP_SWS_Fw_40100] [CP_SWS_Fw_40101] [CP_SWS_Fw_40102] [CP_SWS_Fw_40103] [CP_SWS_Fw_40104] [CP_SWS_Fw_40105] [CP_SWS_Fw_40106] [CP_SWS_Fw_50010] [CP_SWS_Fw_50011] [ECUC_Firewall_00146] [ECUC_Firewall_00147] [ECUC_Firewall_00148] [ECUC_Firewall_00149] [ECUC_Firewall_00150] [ECUC_Firewall_00151] [ECUC_Firewall_00152]

### B.3.2 Changed Specification Items in R24-11

[CP_SWS_Fw_50004] [CP_SWS_Fw_50006] [CP_SWS_Fw_50007] [CP_SWS_Fw_50008] [CP_SWS_Fw_60001] [CP_SWS_Fw_60002] [CP_SWS_Fw_60003] [CP_SWS_Fw_60019] [CP_SWS_Fw_60020] [CP_SWS_Fw_60021] [CP_SWS_Fw_60022] [CP_SWS_Fw_60023] [CP_SWS_Fw_60024] [CP_SWS_Fw_60025] [CP_SWS_Fw_60026] [CP_SWS_Fw_60027] [CP_SWS_Fw_60028] [CP_SWS_Fw_60029] [CP_SWS_Fw_60030] [CP_SWS_Fw_60031] [CP_SWS_Fw_60032] [CP_SWS_Fw_91006] [CP_SWS_Fw_91008] [CP_SWS_Fw_91009] [CP_SWS_Fw_91010] [CP_SWS_Fw_91011] [CP_SWS_Fw_91012]

### B.3.3 Deleted Specification Items in R24-11

[CP_SWS_Fw_30001] [CP_SWS_Fw_40001] [CP_SWS_Fw_40002] [CP_SWS_Fw_40003] [CP_SWS_Fw_40005] [CP_SWS_Fw_50001] [CP_SWS_Fw_50002] [CP_SWS_Fw_91002]

## B.4 Traceable item history of this document according to AUTOSAR Release R23-11

Document newly introduced in R23-11.

### B.4.1 Added Specification Items in R23-11

[CP_SWS_Fw_30001] [CP_SWS_Fw_30002] [CP_SWS_Fw_30003] [CP_SWS_Fw_30004] [CP_SWS_Fw_30005] [CP_SWS_Fw_30006] [CP_SWS_Fw_30007] [CP_SWS_Fw_30008] [CP_SWS_Fw_30009] [CP_SWS_Fw_30010] [CP_SWS_Fw_30011] [CP_SWS_Fw_30012] [CP_SWS_Fw_30013] [CP_SWS_Fw_30014] [CP_SWS_Fw_30015] [CP_SWS_Fw_30016] [CP_SWS_Fw_30017] [CP_SWS_Fw_30018] [CP_SWS_Fw_30019] [CP_SWS_Fw_30020] [CP_SWS_Fw_30021]

[CP_SWS_Fw_30022] [CP_SWS_Fw_30023] [CP_SWS_Fw_30024] [CP_SWS_Fw_30025] [CP_SWS_Fw_30026] [CP_SWS_Fw_40001] [CP_SWS_Fw_40002] [CP_SWS_Fw_40003] [CP_SWS_Fw_40004] [CP_SWS_Fw_40005] [CP_SWS_Fw_40007] [CP_SWS_Fw_40008] [CP_SWS_Fw_40009] [CP_SWS_Fw_40011] [CP_SWS_Fw_40012] [CP_SWS_Fw_50001] [CP_SWS_Fw_50002] [CP_SWS_Fw_50003] [CP_SWS_Fw_50004] [CP_SWS_Fw_50005] [CP_SWS_Fw_50006] [CP_SWS_Fw_50007] [CP_SWS_Fw_50008] [CP_SWS_Fw_50009] [CP_SWS_Fw_60001] [CP_SWS_Fw_60002] [CP_SWS_Fw_60003] [CP_SWS_Fw_60004] [CP_SWS_Fw_60005] [CP_SWS_Fw_60006] [CP_SWS_Fw_60007] [CP_SWS_Fw_60008] [CP_SWS_Fw_60009] [CP_SWS_Fw_60010] [CP_SWS_Fw_60011] [CP_SWS_Fw_60012] [CP_SWS_Fw_60013] [CP_SWS_Fw_60014] [CP_SWS_Fw_60015] [CP_SWS_Fw_60016] [CP_SWS_Fw_60017] [CP_SWS_Fw_60018] [CP_SWS_Fw_60019] [CP_SWS_Fw_60020] [CP_SWS_Fw_60021] [CP_SWS_Fw_60022] [CP_SWS_Fw_60023] [CP_SWS_Fw_60024] [CP_SWS_Fw_60025] [CP_SWS_Fw_60026] [CP_SWS_Fw_60027] [CP_SWS_Fw_60028] [CP_SWS_Fw_60029] [CP_SWS_Fw_60030] [CP_SWS_Fw_60031] [CP_SWS_Fw_60032] [CP_SWS_Fw_60033] [CP_SWS_Fw_61000] [CP_SWS_Fw_91000] [CP_SWS_Fw_91001] [CP_SWS_Fw_91002] [CP_SWS_Fw_91003] [CP_SWS_Fw_91004] [CP_SWS_Fw_91005] [CP_SWS_Fw_91006] [CP_SWS_Fw_91007] [CP_SWS_Fw_91008] [CP_SWS_Fw_91009] [CP_SWS_Fw_91010] [CP_SWS_Fw_91011] [CP_SWS_Fw_91012]

### B.4.2 Changed Specification Items in R23-11

none

### B.4.3 Deleted Specification Items in R23-11

none