| Document Title | Specification of Vehicle-2-X Geo Networking |
|---|---|
| **Document Owner** | AUTOSAR |
| **Document Responsibility** | AUTOSAR |
| **Document Identification No** | 793 |

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

## Document Change History

| Date | Release | Changed by | Change Description |
|---|---|---|---|
| 2022-11-24 | R22-11 | AUTOSAR Release Management | • Editorial changes to reflect the introduction of V2XDM |
| 2021-11-25 | R21-11 | AUTOSAR Release Management | • Correction in V2xGn_TxConfirmation prototype<br>• Scrubbing and editorial changes |
| 2020-11-30 | R20-11 | AUTOSAR Release Management | • Editorial changes |
| 2019-11-28 | R19-11 | AUTOSAR Release Management | • Added support for C2CCC BSP 1.3<br>• Changed Document Status from Final to published |
| 2018-10-31 | 4.4.0 | AUTOSAR Release Management | • Update ETSI EN 302 636-4-1 v1.3.1<br>• Corrections on init and configuration<br>• Corrections in Tx and Rx flows<br>• Editorial changes |
| 2017-12-08 | 4.3.1 | AUTOSAR Release Management | • Editorial changes |
| 2016-11-30 | 4.3.0 | 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 ..... 5
1.1 Architectural overview ..... 5
1.2 Functional overview ..... 6
2 Acronyms and abbreviations ..... 7
3 Related documentation ..... 8
3.1 Input documents ..... 8
3.2 Related standards and norms ..... 8
3.3 Related specification ..... 10
4 Constraints and assumptions ..... 11
4.1 Limitations ..... 11
4.2 Applicability to car domains ..... 11
4.3 Authorisation Tickets and Pseudonyms ..... 11
5 Dependencies to other modules ..... 12
5.1 AUTOSAR DET (Default Error Tracer) ..... 12
5.2 AUTOSAR EcuM (Ecu State Manager) ..... 12
5.3 AUTOSAR Ethernet Interface (EthIf) ..... 12
5.4 AUTOSAR Vehicle-2-X Basic Transport Protocol (V2xBtp) ..... 12
5.5 AUTOSAR Vehicle-2-X Management (V2xM) ..... 12
5.6 File structure ..... 12
5.6.1 Code file structure ..... 12
6 Requirements traceability ..... 13
7 Functional specification ..... 16
7.1 General Functionality ..... 16
7.2 GeoNetworking Packet Structure and Format ..... 16
7.3 GeoNetworking Protocol Operations ..... 17
7.3.1 Network Management ..... 17
7.3.2 Security Mechanisms ..... 17
7.4 Message Forwarding ..... 18
7.5 Message Transmission ..... 18
7.6 Message Reception ..... 19
7.7 Error classification ..... 20
7.7.1 Development Errors ..... 20
7.7.2 Runtime Errors ..... 20
7.7.3 Transient Faults ..... 21
7.7.4 Production Errors ..... 21
7.7.5 Extended Production Errors ..... 21
8 API specification ..... 22
8.1 Imported types ..... 22
8.2 Type definitions ..... 23
8.2.1 V2xGn_TxParamsType ..... 23
8.3 Function definitions ..... 24

8.3.1 V2xGn_Init ..... 24
8.3.2 V2xGn_GetVersionInfo ..... 24
8.3.3 V2xGn_V2xM_PreparePseudonymChange ..... 25
8.3.4 V2xGn_V2xM_CommitPseudonymChange ..... 26
8.3.5 V2xGn_V2xM_AbortPseudonymChange ..... 27
8.3.6 V2xGn_Transmit ..... 28
8.4 Call-back notifications ..... 29
8.4.1 This is a list of functions provided for other modules. V2xGn_V2xM_EncapConfirmation ..... 29
8.4.2 V2xGn_V2xM_DecapConfirmation ..... 30
8.4.3 V2xGn_RxIndication ..... 31
8.4.4 V2xGn_TxConfirmation ..... 32
8.5 Scheduled functions ..... 33
8.5.1 V2xGn_MainFunction ..... 33
8.6 Expected Interfaces ..... 33
8.6.1 Mandatory Interfaces ..... 33
8.6.2 Optional Interfaces ..... 34
9 Sequence diagrams ..... 36
9.1 V2xGn_RxIndication ..... 36
9.2 V2xGn_Transmit ..... 37
9.3 V2xGn_V2xM_UpdatePseudonym ..... 38
9.4 V2xGn_MainFunction ..... 38
10 Configuration specification ..... 39
10.1 Containers and configuration parameters ..... 39
10.1.1 Variants ..... 39
10.1.2 V2xGn ..... 39
10.1.3 V2xGnGeneral ..... 40
10.1.4 V2xGnBeaconService ..... 49
10.1.5 V2xGnPacketForwarding ..... 50
10.2 Published Information ..... 59
11 Not applicable requirements ..... 60

# 1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Vehicle-2-X GeoNetworking (V2xGn).

V2xGn together with Vehicle-2-X Facilities (V2xFac) [8], Vehicle-2-X Basic Transport (V2xBtp) [9], Vehicle-2-X Management (V2xM) [10] , Vehicle-2-X Data Manager [26] and AUTOSAR BSW modules Ethernet Interface (EthIf) [5], Wireless Ethernet Driver (WEth) [11] and Wireless Ethernet Transceiver Driver (WEthTrcv) [12] forms the V2X stack within the AUTOSAR architecture.

The base for this document is the GeoNetworking specification [19][20]. It is assumed that the reader is familiar with this specification.

## 1.1 Architectural overview

V2xGn provides services to and is dependent on the upper V2xBtp module and uses the services of and gets services from the lower EthIf module to realize its functions explained in sections 1.2 and chapter 7 of this document.

Positioning of the V2xGn module within the AUTOSAR BSW and the Layered Software architecture [1] is shown in Figure 1.

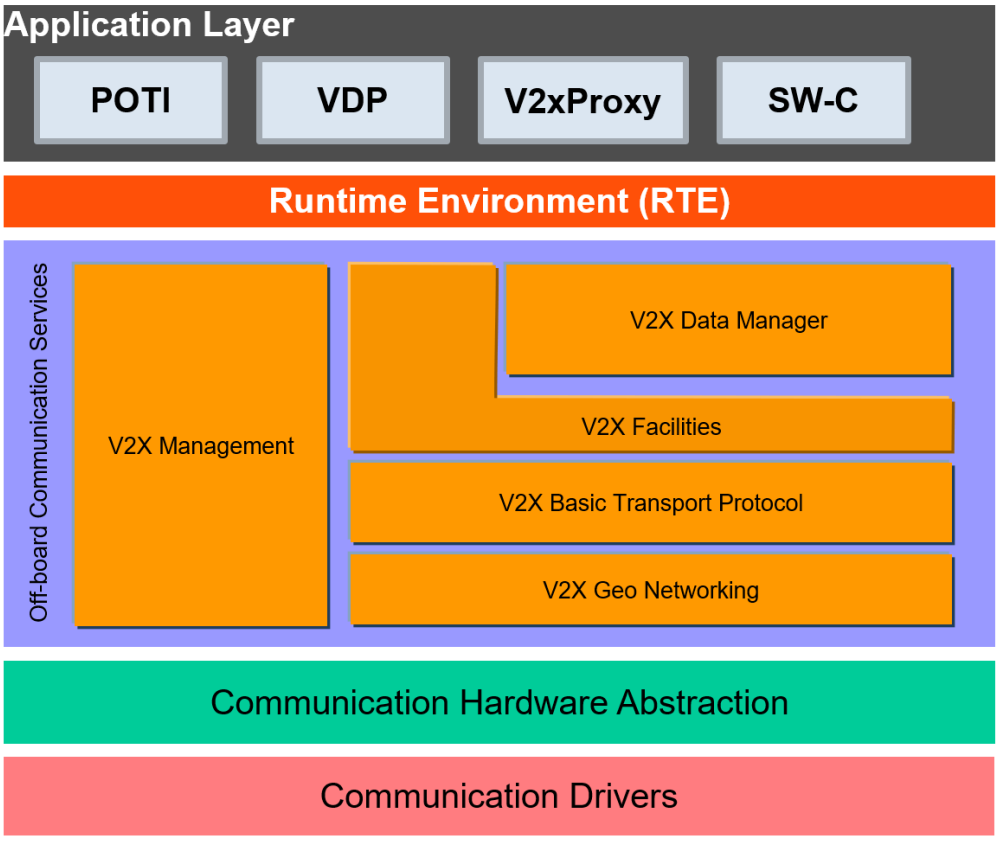



**Figure 1 – AUTOSAR BSW software architecture – V2xGn module scope**

## 1.2 Functional overview

The internal functionality of the V2xGn module should comply to the medium independent specification of the GeoNetworking protocol [19] and the medium dependent specification of the GeoNetworking protocol [20], relying on ETSI ITS-G5 technology as medium. The module provides services to the upper V2xBtp module specified in [9] and in order to provide its packet transport services, it relies on the lower EthIf module [5]. Vehicle-2-X specific data is also exchanged with the V2xM module.

GeoNetworking protocol is a set of network layer functionalities that enables ad hoc communication without infrastructure support using geographical positions of the communicating entities. It supports communication among individual Intelligent Transport System (ITS) station and distribution of packets in geographical areas. As GeoNetworking can be executed over different ITS technologies such as ITS-G5 and infrared, GeoNetworking specification consists of a standard for media-independent functionality [19] which specifies all functions that are common to all ITS access technologies and one or more media-dependent specifications [20] which includes extensions for a specific ITS technology.

# 2 Acronyms and abbreviations

The following acronyms and abbreviations have a local scope and are therefore not contained in the AUTOSAR glossary [4].

| ***Abbreviation / Acronym:*** | ***Description:*** |
|---|---|
| BTP | Basic Transport Protocol |
| CBF | Contention-Based Forwarding |
| DET | Default Error Tracer |
| GAC | GeoAnycast |
| GBC | GeoBroadcast |
| GN | GeoNetworking |
| GN-SDU | GeoNetworking Service Data Unit |
| ITS | Intelligent Transport System |
| MAC | Medium Access Control |
| SHB | Single Hop Broadcast |
| TC | Traffic Class |
| TSB | Topologically Scoped Broadcast |

# 3 Related documentation

## 3.1 Input documents

[1] AUTOSAR Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

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

[3] AUTOSAR General Specification for Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

[4] Glossary
AUTOSAR_TR_Glossary

[5] Specification of Ethernet Interface
AUTOSAR_SWS_EthernetInterface.pdf

[6] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager.pdf

[7] Specification of Default Error Tracer
AUTOSAR_SWS_DefaultErrorTracer.pdf

[8] Specification of Vehicle-2-X Facilities
AUTOSAR_SWS_V2XFacilities.pdf

[9] Specification of Vehicle-2-X Basic Transport
AUTOSAR_SWS_V2XBasicTransport.pdf

[10] Specification of Vehicle-2-X Management
AUTOSAR_SWS_V2XManagement.pdf

[11] Specification of Wireless Ethernet Driver
AUTOSAR_SWS_WirelessEthernetDriver.pdf

[12] Specification of Wireless Ethernet Transceiver Driver
AUTOSAR_SWS_WirelessEthernetTransceiverDriver.pdf

[13] Requirements on Vehicle-2-X communication
AUTOSAR_SRS_V2XCommunications.pdf

[26] Specification of Vehicle-2-X Data Manager
AUTOSAR_SWS_V2XDataManager.pdf

## 3.2 Related standards and norms

[14] Intelligent Transport Systems (ITS); Communications Architecture
ETSI EN 302 665 V1.1.1 (2010-09)

[15] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 1: Requirements
ETSI EN 302 636-1 V1.2.1 (2014-04)

[16] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 2: Scenarios
ETSI EN 302 636-2 V1.2.1 (2013-11)

[17] Intelligent Transport Systems (ITS); Vehicular Communications GeoNetworking Part 3: Network Architecture
ETSI EN 302 636-3 V1.2.1 (2014-12)

[18] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol
ETSI EN 302 636-5-1 V2.1.1 (2017-08)

[19] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality
ETSI EN 302 636-4-1 V1.3.1 (2017-08)

[20] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent functionalities for ITS-G5
ETSI TS 102 636-4-2 V1.1.1 (2013-10)

[21] Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 8: Interface between security entity and network and transport layer
ETSI TS 102 723-8

[22] Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 10: Interface between access layer and network and transport layer
ETSI TS 102 723-10 V1.1.1 (2012-11)

[23] C2C-CC BSP Requirement
C2CCC_RS_2037_BSP_Requirements.docx

[24] Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band between access layer and network and transport layer
ETSI TS 102 724 V1.1.1 (2012-10)

[25] List of EtherTypes by IEEE
http://standards.ieee.org/develop/regauth/ethertype/eth.txt

## 3.3 Related specification

AUTOSAR provides a General Specification on Basic Software (SWS BSW General) [3] which is also valid for V2xGn.

Thus, the specification SWS BSW General [3] shall be considered as additional and required specification for V2xGn.

# 4 Constraints and assumptions

## 4.1 Limitations

- The GeoNetworking protocol and therefore the V2xGn module requires a broadcast capable access layer in order to provide transmit services.
- Wireless Communication supports IEEE 802.11p only. Other 802.11 standards (e.g. for infrastructure networks and integration with TCP/IP) can be extended in future releases of the AUTOSAR standard.
- The V2X modules follow the guidance regarding the Day-1 scenarios defined by the Basic System Standards Profile from Car-2-Car-Consortium.
- AUTOSAR R19-11 only focuses on the European version of car-to-car communication as defined by ETSI. Extension to other regions are planned for future releases of the AUTOSAR standard.

## 4.2 Applicability to car domains

This specification is applicable to all car domains.

## 4.3 Authorisation Tickets and Pseudonyms

The Authorisation Ticket (AT) is referred to as Pseudonym in this document.

# 5 Dependencies to other modules

This section describes the relations of the V2xGn module to other modules within the AUTOSAR basic software architecture. It outlines the modules that are required or optional for the realization of the V2xGn module and the V2xGn services that these modules use.

## 5.1 AUTOSAR DET (Default Error Tracer)

In development mode, the V2xGn module reports errors through DET [7].

## 5.2 AUTOSAR EcuM (Ecu State Manager)

The EcuM [6] is responsible for the initialization of V2xGn.

## 5.3 AUTOSAR Ethernet Interface (EthIf)

The Ethernet Interface is the lower layer module of the V2xGn module.

## 5.4 AUTOSAR Vehicle-2-X Basic Transport Protocol (V2xBtp)

The V2xBtp is the upper layer module of the V2xGn module.

## 5.5 AUTOSAR Vehicle-2-X Management (V2xM)

V2xM is used for interchange of Data with other V2X-Modules. Security mechanisms are configured by the V2xM and are used by V2xGn.

## 5.6 File structure

### 5.6.1 Code file structure

For details refer to the chapter 5.1.6 “Code file structure” in *SWS_BSWGeneral* [3].

# 6 Requirements traceability

| Requirement | Description | Satisfied by |
|---|---|---|
| SRS_BSW_00345 | BSW Modules shall support pre-compile configuration | SWS_V2xGn_00078 |
| SRS_V2X_00010 | The implementation of the V2X system shall follow additional guidance given by C2C-CC requirements | SWS_V2xGn_00269, SWS_V2xGn_20169, SWS_V2xGn_20181, SWS_V2xGn_20260, SWS_V2xGn_20262, SWS_V2xGn_20263, SWS_V2xGn_20264, SWS_V2xGn_20265, SWS_V2xGn_20266, SWS_V2xGn_20267, SWS_V2xGn_20268, SWS_V2xGn_20270, SWS_V2xGn_20401 |
| SRS_V2X_00160 | The V2X system shall use end-to-end security for communication to external entities | SWS_V2xGn_00026, SWS_V2xGn_20251 |
| SRS_V2X_00161 | The V2X system shall employ the security envelope on its Network layer | SWS_V2xGn_00012, SWS_V2xGn_20251 |
| SRS_V2X_00163 | The "verification" of a message shall comprise at least cryptographic verification of the message's signature | SWS_V2xGn_NA_00003 |
| SRS_V2X_00164 | The V2X system shall only forward verified messages | SWS_V2xGn_00026 |
| SRS_V2X_00174 | The V2X system shall support key origin authentication for the new (long-term or pseudonym) public keys that are provided in certificate signing requests | SWS_V2xGn_NA_00003 |
| SRS_V2X_00176 | The V2X system shall change pseudonyms | SWS_V2xGn_00028, SWS_V2xGn_00091, SWS_V2xGn_00112, SWS_V2xGn_00115 |
| SRS_V2X_00184 | The V2X system shall allow applications to block the pseudonym change | SWS_V2xGn_NA_00003 |
| SRS_V2X_00189 | The V2X system shall be able to estimate vehicle states | SWS_V2xGn_NA_00004 |
| SRS_V2X_00190 | The V2X system shall handle vehicle states in a consistent manner | SWS_V2xGn_NA_00004 |
| SRS_V2X_00193 | The V2X system shall use ITS time as time base | SWS_V2xGn_NA_00004 |
| SRS_V2X_00207 | The difference between Station clock and time base shall be estimated | SWS_V2xGn_NA_00004 |
| SRS_V2X_00214 | The V2X system shall allow applications to deactivate transmission of CAMs | SWS_V2xGn_NA_00004 |

| | | |
|---|---|---|
| SRS_V2X_00232 | The V2X system shall cooperate with tolling zone stations in vicinity | SWS_V2xGn_NA_00001 |
| SRS_V2X_00242 | The V2Xsystem shall manage CAM transmission in such a way, that no outdated CAM will be transmitted | SWS_V2xGn_NA_00001 |
| SRS_V2X_00245 | The V2X system shall support per-packet transmission power control | SWS_V2xGn_NA_00001 |
| SRS_V2X_00259 | The V2X system shall manage the life time of all DENM packets | SWS_V2xGn_20259 |
| SRS_V2X_00279 | The V2X system shall support circular, rectangular and ellipsoidal geographical areas | SWS_V2xGn_20266 |
| SRS_V2X_00291 | The V2X system shall only send messages with valid postion and time | SWS_V2xGn_NA_00002 |
| SRS_V2X_00301 | The V2X system's Facility Layer shall handle DENM repetition | SWS_V2xGn_NA_00002 |
| SRS_V2X_00318 | The V2X system's Facility Layer shall generate traces and path histories | SWS_V2xGn_NA_00002 |
| SRS_V2X_00322 | The V2X system shall provide services to avoid channel congestion of the shared media | SWS_V2xGn_NA_00001 |
| SRS_V2X_00323 | The V2X system shall provide mitigation techniques to avoid disturbing other services operating at nearby frequencies | SWS_V2xGn_NA_00004 |
| SRS_V2X_00391 | The V2X system's access layer shall be ITS-G5 compliant | SWS_V2xGn_20414, SWS_V2xGn_NA_00001 |
| SRS_V2X_00405 | The V2X basic system shall support services for confidentiality | SWS_V2xGn_NA_00003 |
| SRS_V2X_00406 | The end-to-end security envelope shall be generated depending on the message type | SWS_V2xGn_NA_00003 |
| SRS_V2X_00407 | The signature in the end-to-end security envelope shall be generated using a private key corresponding to a valid authorization ticket (pseudonym certificate) | SWS_V2xGn_NA_00003 |
| SRS_V2X_00412 | The V2X system shall inform | SWS_V2xGn_NA_00003 |

| | | |
|---|---|---|
| | the driver about the expiration of the pseudonym certificates | |
| SRS_V2X_00413 | The V2X system shall inform the driver about the expiration of the Long Term Certificates | SWS_V2xGn_NA_00003 |
| SRS_V2X_00451 | The V2X system's access layer shall be compliant to the ETSI Harmonized Channel Specifications | SWS_V2xGn_NA_00001 |
| SRS_V2X_00511 | The V2X system shall provide services for communication to multiple, geographically scattered and movable entities | SWS_V2xGn_NA_00004 |
| SRS_V2X_00531 | The V2X system's Networking Layer shall support addressing based on geographic coordinates | SWS_V2xGn_20250, SWS_V2xGn_20251, SWS_V2xGn_20252, SWS_V2xGn_20255, SWS_V2xGn_20258, SWS_V2xGn_20414, SWS_V2xGn_20416 |
| SRS_V2X_00693 | The V2X system shall provide functionality for generating traces and path histories | SWS_V2xGn_NA_00004 |
| SRS_V2X_00711 | The V2X system's CA basic service shall be compliant to ETSI Specification of Cooperative Awareness Basic Service | SWS_V2xGn_NA_00002 |
| SRS_V2X_00741 | The V2X system's DEN basic service shall be compliant to ETSI Specifications of Decentralized Environmental Notification Basic Service | SWS_V2xGn_NA_00002 |

**Note:**
Requirement IDs within this document have an encoding to state where each requirement has its origin:

- SWS items starting with a leading 0 (SWS_V2xGn_0xxxx) are module specific and not inherited.
- SWS items starting with a leading 2 (SWS_V2xGn_2xxxx) are inherited from C2C-CC Basic System Profile

# 7 Functional specification

## 7.1 General Functionality

**[SWS_V2xGn_00012]** [ The V2xGn Module shall implement the GeoNetworking Protocol as defined in [19], [20] and [23] unless specified otherwise in this document.] (SRS_V2X_00161)

**[SWS_V2xGn_00013]** [ The GeoNetworking Protocol shall support the GeoNetworking related requirements specified in [23] ] ( )

**[SWS_V2xGn_20250]** [ All default constants and parameters of the V2xGn module not defined or overwritten in the current document shall be set as specified in Annex H of [19]. ] (SRS_V2X_00531)

**[SWS_V2xGn_20251]** [ The V2xGn module shall be implemented assuming the ETSI parameter itsGnSecurity is constantly set to ENABLED. ] (SRS_V2X_00531, SRS_V2X_00160, SRS_V2X_00161)

**[SWS_V2xGn_20252]** [ The V2xGn module shall only support anonymous address configuration mode. ] (SRS_V2X_00531)

**[SWS_V2xGn_20255]** [ The V2xGn module shall support geo-areas areas of up to 80 km2. In consequence, the itsGnMaxGeoAreaSize shall have a value of 80. It is configurable by the configuration option V2xGnItsGnMaxGeoAreaSize. ] (SRS_V2X_00531)

**[SWS_V2xGn_20414]** [ The V2xGn module shall be implemented with respect to the ETSI parameter itsGnIfType constantly set to ITS-G5. ] (SRS_V2X_00531, SRS_V2X_00391)

**[SWS_V2xGn_00130]** [ The V2xGn module shall get the pointer to the current time information via V2xM_GetRefTimePtr within the V2xGn_Init function. ] ( )

**[SWS_V2xGn_20416]** [ Packet repetition shall not be performed by V2xGn module and the corresponding steps in the packet handling procedures in [19] clause 10.3 shall not be executed.
The parameter 'Maximum repetition time' of the service primitive GN-DATA.request is not applicable. Also, the GN protocol constant itsGnMinPacketRepetitionInterval is not applicable.] (SRS_V2X_00531)

## 7.2 GeoNetworking Packet Structure and Format

**[SWS_V2xGn_00020]** [ The GeoNetworking protocol shall only support the packet header types Single Hop Broadcast packet header, GeoBroadcast packet headers and Beacon packet header.
] ( )

**[SWS_V2xGn_20258]** [ The V2xGn module shall set the LifeTime field of all SHB packets to 1 second. Consequently, the multiplier bit of the LT field shall be set to 1 and the base bit of the LT field shall be set to 1. ] (SRS_V2X_00531)

**[SWS_V2xGn_20259]** [ The V2xGn module shall set the LifeTime field of all GBC packets to the value of the maxPacketLifetime from the transmit parameters TxParams.The value of the LifeTime field shall not exceed the itsGnMaxPacketLifetime, specified in [19], Annex H. ] (SRS_V2X_00259)

## 7.3 GeoNetworking Protocol Operations

### 7.3.1 Network Management

**[SWS_V2xGn_00022]** [ The V2xGn module shall update the local position and time information. The minimum update frequency is configured by the configuration parameter `V2xGnItsGnMinUpdateFrequencyEPV`. The scheduled function `V2xGn_MainFunction()` shall be used for the cyclic update. ] ( )

**[SWS_V2xGn_00023]** [ The V2xGn module shall support GeoNetworking beaconing. The scheduled function `V2xGn_MainFunction()` shall be used for the cyclic beaconing. ] ( )

**[SWS_V2xGn_00269]** [ The V2xGn module shall only send beacons if ego position is accurate enough to set the Position Accuracy Indicator (PAI) to 1. ] (SRS_V2X_00010)

**[SWS_V2xGn_00081]** [ The V2xGn module shall support Location Table Maintainance. The scheduled function `V2xGn_MainFunction()` shall be used for the cyclic maintainance of the Location Table. ] ( )

**[SWS_V2xGn_00129]** [ The V2xGn module shall get the current position and time information via V2xM_GetPositionAndTime within the MainFunction. ] ( )

### 7.3.2 Security Mechanisms

**[SWS_V2xGn_00026]** [ The V2xGn module shall use security services provided by V2xM V2xM_V2xGn_ReqEncap and V2xM_V2xGn_ReqDecap.] (SRS_V2X_00160, SRS_V2X_00164)

**[SWS_V2xGn_00028]** [ The V2xGn shall suspend transmission of messages and clear transmit buffers when a pseudonym changes is in preparation. ] (SRS_V2X_00176)

**Note:** The V2xM will notify the V2xGn about pseudonym changes via V2xGn_V2xM_PreparePseudonymChange, V2xGn_V2xM_CommitPseudonymChange and V2xGn_V2xM_AbortPseudonymChange.

## 7.4 Message Forwarding

**[SWS_V2xGn_20266]** ⌈ The V2xGn module shall only support Area forwarding algorithms specified in [19] Annex E.3. ⌋ (SRS_V2X_00010, SRS_V2X_00279)

**[SWS_V2xGn_20267]** ⌈ When forwarding packets, the V2xGn module shall use the DCC profile DP3 as defined in [24]. ⌋ (SRS_V2X_00010)

**[SWS_V2xGn_20169]** ⌈ The V2xGn module shall check the distance from the sender position - in the security envelope, if available - and forward only messages with a distance from the sender of equal or less than 6 km. ⌋ (SRS_V2X_00010)

## 7.5 Message Transmission

**[SWS_V2xGn_00034]** ⌈ The V2xGn module shall provide the API `V2xGn_Transmit()` to enable transmit requests from the V2xBtp Module. ⌋ ( )

**[SWS_V2xGn_00082]** ⌈
The V2xGn module shall use EthIf_ProvideTxBuffer to aquire a buffer within the Wireless Ethernet Driver for a V2X Packet to be transmitted. This has to be done during the V2xGn_Transmit context. ⌋()

**[SWS_V2xGn_00083]** ⌈
The V2xGn module shall provide transmission parameters to the Wireless Ethernet Driver for a V2X Packet to be transmitted via an API call to EthIf_SetBufWTxParams. This has to be done during the V2xGn_Transmit context. ⌋()

**[SWS_V2xGn_00035]** ⌈ The V2xGn module shall transmits packets using the `EthIf_Transmit()` API provided by the EthIf Module. This has to be done during the V2xGn_Transmit context. ⌋ ( )

**[SWS_V2xGn_00036]** ⌈ If the configuration parameter V2xGnTxConfirmation is enabled, the V2xGn module shall provide information about the status of the transmission with an associated ID (generated by the V2xFac module and handed down to track the status of the packet) to the V2xBtp Module via the `V2xBtp_TxConfirmation()` callback.⌋ ( )

**[SWS_V2xGn_20260]** ⌈ The V2xGn module shall buffer GBC packets when no neighbours are available (store-carry-forward) if the SCF bit of the TC (Traffic Class) field of GBC packets is set to 1. ⌋ (SRS_V2X_00010)

**[SWS_V2xGn_20262]** [ The V2xGn module is not required to offload packets to another channel. Consequently, the channel offload bit of the TC (Traffic Class) field in the BTP frames to be sent should be set to 0. Value 1 will be ignored anyway.] (SRS_V2X_00010)

**[SWS_V2xGn_20263]** [ The V2xGn module shall only use the DCC profiles specified in [SWS_WEth_20235]. Consequently, the DCC Profile ID bits of the TC (Traffic Class) field shall only use the DPID values defined in **[SWS_WEth_20235]**.
] (SRS_V2X_00010)

**[SWS_V2xGn_20264]** [ The V2xGn shall set the itsGnIsMobile bit of the Flags field to 1. ] (SRS_V2X_00010)

**[SWS_V2xGn_20265]** [ The V2xGn shall set the Maximum Hop Limit (MHL) field to 10. ] (SRS_V2X_00010)

**[SWS_V2xGn_20270]** [ All GeoNetworking frames sent by the V2xGn module shall use the EtherType value 0x8947 as listed by the IEEE Registration Authority at [25]. ] (SRS_V2X_00010)

**[SWS_V2xGn_20401]** [ The GN Source Address shall be constructed as follows:

- Set the field M (bit 0) to 0.
- Set the field ST (bits 1 to 5) to the station type of the ITS-S. The station type in the GN source address shall be identical to the station type in CAMs/DENMs.
- Set reserved bits 6 to 15 to 0.
- Set the field MID (bits 16 to 63) to the value of the MAC address.

] (SRS_V2X_00010)

**[SWS_V2xGn_00128]** [ The V2xGn module shall call V2xM_GetGlobalTxParams that delivers own channel CBR information set in the GeoNetworking header to be transmitted. ] ( )

## 7.6 Message Reception

**[SWS_V2xGn_00038]** [ The V2xGn module shall create a unique TransactionId for each received packet. This TransactionId is handed up to track the received packets and is used for verification on demand. ] ( )

**[SWS_V2xGn_00039]** [ The V2xGn module shall indicate received packets via the `V2xBtp_RxIndication()` callback to the BTP module. ] ( )

**[SWS_V2xGn_00084]** [ The V2xGn module shall get the reception status of a received packet during the `V2xGn_RxIndication()` from the EthIf module with a call to `EthIf_GetBufWRxParams()`. ] ( )

**[SWS_V2xGn_20268]** [ The V2xGn module shall only use duplicate packet detection as specified in [19] Annex A.2 and A.3. ] (SRS_V2X_00010)

**[SWS_V2xGn_20181]** [ If the V2xGn module detects a collision of the least significant 32 bit of the "Certificate digest" / "hashedId8" with the "Certificate digest" / "hashedId8" of another ITS station, it shall initiate a change of its authorization ticket (pseudonym) only if the certificate corresponding to the other "Certificate digest" / "hashedId8" is valid, and the current authorization ticket was selected according to rules defined in SWS_V2xM_00201 (that is to say no such collision has already triggered the change to the current authorization ticket). ] (SRS_V2X_00010)

**[SWS_V2xGn_00127]** [ The V2xGn module shall call V2xM_SetGlobalRxParams with CBR information extracted from the GeoNetworking header. ] ( )

**[SWS_V2xGn_00131]** [ The V2xGn module shall use V2xM_CalcDistance when calculations of geographical distances are necessary for the V2xGn protocol operations. ] ( )

## 7.7 Error classification

Section 7.x "Error Handling" of the document "General Specification of Basic Software Modules" 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

**[SWS_V2xGn_00134]** [
In case development error detection is enabled for the V2xGn module, the V2xGn module shall check API parameters for validity and report detected errors to the DET. ]()

**[SWS_V2xGn_00041]**[

| *Type of error* | *Related error code* | *Error value* |
|---|---|---|
| API service called with invalid parameter | V2XGN_E_PARAM | 0x01 |
| API service called with invalid pointer | V2XGN_E_PARAM_POINTER | 0x02 |
| API service used without module initialization | V2XGN_E_UNINIT | 0x03 |
| API service called with invalid configuration pointer | V2XGN_E_INIT_FAILED | 0x04 |

]()

### 7.7.2 Runtime Errors

There are no runtime errors.

### 7.7.3 Transient Faults

There are no transient faults.

### 7.7.4 Production Errors

There are no production errors.

### 7.7.5 Extended Production Errors

There are no extended production errors.

# 8 API specification

## 8.1 Imported types

In this chapter all types included from the following modules are listed:

**[SWS_V2xGn_00042]**[

| *Module* | *Header File* | *Imported Type* |
|---|---|---|
| ComStack_Types | ComStack_Types.h | BufReq_ReturnType |
| Eth | Eth_GeneralTypes.h | Eth_BufIdxType |
| | Eth_GeneralTypes.h | Eth_FrameType |
| Std | Std_Types.h | Std_ReturnType |
| | Std_Types.h | Std_VersionInfoType |
| V2x_GeneralTypes | Rte_V2xM_Type.h | V2xM_PositionAndTimeType |
| | Rte_V2xM_Type.h | V2xM_SecReportType |
| | V2x_GeneralTypes.h | V2x_ChanType |
| | V2x_GeneralTypes.h | V2x_GnAddressType |
| | V2x_GeneralTypes.h | V2x_GnAreaShapeType |
| | V2x_GeneralTypes.h | V2x_GnDestinationAreaType |
| | V2x_GeneralTypes.h | V2x_GnDestinationType |
| | V2x_GeneralTypes.h | V2x_GnLongPositionVectorType |
| | V2x_GeneralTypes.h | V2x_GnPacketTransportType |
| | V2x_GeneralTypes.h | V2x_GnTxResultType |
| | V2x_GeneralTypes.h | V2x_GnUpperProtocolType |
| | V2x_GeneralTypes.h | V2x_PseudonymType |
| | V2x_GeneralTypes.h | V2x_SecProfileType |
| | V2x_GeneralTypes.h | V2x_SecReturnType |
| | V2x_GeneralTypes.h | V2x_TrafficClassIdType |
| V2xBtp | V2xBtp.h | V2xBtp_RxParamsType |
| WEth | WEth_GeneralTypes.h | WEth_BufWRxParamIdType |
| | WEth_GeneralTypes.h | WEth_BufWTxParamIdType |

](

## 8.2 Type definitions

### 8.2.1 V2xGn_TxParamsType

**[SWS_V2xGn_00063]**[

| | | |
|---|---|---|
| ***Name*** | V2xGn_TxParamsType | |
| ***Kind*** | Structure | |
| ***Elements*** | upperProtocol | |
| | ***Type*** | V2x_GnUpperProtocolType |
| | ***Comment*** | The protocol which triggered the request. (e.g. BTP-A or BTP-B) |
| | transportType | |
| | ***Type*** | V2x_GnPacketTransportType |
| | ***Comment*** | Specifies the packet transportation type |
| | destinationAddress | |
| | ***Type*** | V2x_GnAddressType |
| | ***Comment*** | Destination address for GeoUnicast packet |
| | destinationArea | |
| | ***Type*** | V2x_GnDestinationAreaType |
| | ***Comment*** | Destination area for GeoBroadcast/GeoAnycast packet. |
| | destinationType | |
| | ***Type*** | V2x_GnDestinationType |
| | ***Comment*** | Select which destination type (destinationAddress or destinationArea is used for this packet). |
| | secProfile | |
| | ***Type*** | V2x_SecProfileType |
| | ***Comment*** | Parameters depending on the security service. |
| | maxPacketLifetime | |
| | ***Type*** | uint16 |
| | ***Comment*** | Time a packet can be buffered until it reaches the destination. |
| | trafficClassId | |
| | ***Type*** | V2x_TrafficClassIdType |
| | ***Comment*** | Set of parameter specifying the traffic class for the message. |
| ***Description*** | Structure containing parameters for the V2xGn_Transmit() API. | |

| | |
|---|---|
| ***Available via*** | V2xGn.h |

]()

## 8.3 Function definitions

### 8.3.1 V2xGn_Init

**[SWS_V2xGn_00068]**⌈

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_Init | |
| ***Syntax*** | `void V2xGn_Init (`<br>`  void* CfgPtr`<br>`)` | |
| ***Service ID [hex]*** | 0x01 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | CfgPtr | Pointer to configuration set |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Initializes the V2xGn module. | |
| ***Available via*** | V2xGn.h | |

]()**[SWS_V2xGn_00133]** ⌈
If development error detection is enabled: The function shall check the parameter CfgPtr for containing a valid configuration. If the check fails, the function shall raise the development error V2XGN_E_INIT_FAILED. ⌋ ()

### 8.3.2 V2xGn_GetVersionInfo

**[SWS_V2xGn_00069]**⌈

| | |
|---|---|
| ***Service Name*** | V2xGn_GetVersionInfo |
| ***Syntax*** | `void V2xGn_GetVersionInfo (`<br>`  Std_VersionInfoType* VersionInfoPtr`<br>`)` |
| ***Service ID [hex]*** | 0x02 |
| ***Sync/Async*** | Synchronous |

| | | |
|---|---|---|
| ***Reentrancy*** | Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | VersionInfo Ptr | Pointer to where to store the version information of this module. |
| ***Return value*** | None | |
| ***Description*** | Returns the version information of this module. | |
| ***Available via*** | V2xGn.h | |

](**[SWS_V2xGn_00090]** [
If development error detection is enabled: the function shall check the parameter VersionInfoPtr for being valid. If the check fails, the function shall raise the development error V2XGN_E_PARAM_POINTER. ]()

### 8.3.3 V2xGn_V2xM_PreparePseudonymChange

**[SWS_V2xGn_00072]**[

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_V2xM_PreparePseudonymChange | |
| ***Syntax*** | `Std_ReturnType V2xGn_V2xM_PreparePseudonymChange ( const V2x_PseudonymType* PseudonymPtr )` | |
| ***Service ID [hex]*** | 0x05 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | PseudonymPtr | The Pseudonym provided by V2xM |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | E_OK: operation successful<br>E_NOT_OK: operation failed |
| ***Description*** | This function is called by the V2xM when a Pseudonym Change occurs to prepare the change in every module using it. | |
| ***Available via*** | V2xGn_V2xM.h | |

](**[SWS_V2xGn_00091]** [

The function V2xGn_V2xM_PreparePseudonymChange shall prepare the setting of the pseudonym specific part of the GeoNetworking Address being used for packet transmission. ](SRS_V2X_00176)

**[SWS_V2xGn_00092]** [
If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ]()

**[SWS_V2xGn_00093]** [
If development error detection is enabled: the function shall check the parameter PseudonymPtr for being valid. If the check fails, the function shall raise the development error V2XGN_E_PARAM_POINTER. ]()

**Note**: This starts a module internal transaction for the pseudonym change. The actual pseudonym change becomes effective only after an API call to V2xGn_V2xM_CommitPseudonymChange.

### 8.3.4 V2xGn_V2xM_CommitPseudonymChange

**[SWS_V2xGn_00111]**[

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_V2xM_CommitPseudonymChange | |
| ***Syntax*** | `Std_ReturnType V2xGn_V2xM_CommitPseudonymChange ( void )` | |
| ***Service ID [hex]*** | 0x09 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | E_OK: operation successful<br>E_NOT_OK: operation failed |
| ***Description*** | This function is called by the V2xM when all modules are OK with the pseudonym change and the change is to be committed. | |
| ***Available via*** | V2xGn_V2xM.h | |

]()**[SWS_V2xGn_00112]** [
The function V2xGn_V2xM_CommitPseudonymChange shall update the pseudonym specific part of the module's GeoNetworking Address. ](SRS_V2X_00176)

**[SWS_V2xGn_00113]** ⌈
If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ⌋()

Note: The function requires previous preparation of the pseudonym via an API call to V2xGn_V2xM_PreparePseudonymChange.

### 8.3.5 V2xGn_V2xM_AbortPseudonymChange

**[SWS_V2xGn_00126]**⌈

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_V2xM_AbortPseudonymChange | |
| ***Syntax*** | `Std_ReturnType V2xGn_V2xM_AbortPseudonymChange ( void )` | |
| ***Service ID [hex]*** | 0x0a | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | None | |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | Std_ReturnType | E_OK: operation successful<br>E_NOT_OK: operation failed |
| ***Description*** | This function is called by the V2xM when not all modules are OK with the pseudonym change and the change is to be rolled back. | |
| ***Available via*** | V2xGn_V2xM.h | |

⌋()**[SWS_V2xGn_00115]** ⌈
The function V2xGn_V2xM_AbortPseudonymChange shall set the state of the module to the state before the pseudonym change has been prepared.
⌋(SRS_V2X_00176)

**[SWS_V2xGn_00116]** ⌈
If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ⌋()

Note: The function requires previous preparation of the pseudonym via an API call to V2xGn_V2xM_PreparePseudonymChange.

### 8.3.6 V2xGn_Transmit

**[SWS_V2xGn_00070]**⌈

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_Transmit | |
| ***Syntax*** | `V2x_GnTxResultType V2xGn_Transmit ( uint16 TransactionId16, const V2xGn_TxParamsType* TxParams, uint16 Length )` | |
| ***Service ID [hex]*** | 0x03 | |
| ***Sync/Async*** | Asynchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | TransactionId16 | Transaction Id for the Packet |
| | TxParams | Additional transmission parameters |
| | Length | Length of the user data |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | V2x_GnTx-ResultType | Values specified in the Type could be returned.<br>V2X_GNTX_ACCEPTED if no error occurred.<br>V2X_GNTX_ACCEPTED transmit has been accepted<br>V2X_GNTX_E_MAXSDUSIZEOVFL transmit has been rejected due to maximum length exceedance<br>V2X_GNTX_E_MAXPACKETLIFETIME transmit has been rejected due to maximum lifetime exceedance<br>V2X_GNTX_E_TCID transmit has been rejected due to unsupported Traffic Class ID<br>V2X_GNTX_E_MAXGEOAREASIZE transmit has been rejected due to GeoArea exceeds max size<br>V2X_GNTX_E_UNSPECIFIED transmit has been rejected due to unspecified reasons |
| ***Description*** | Is called by V2x_Btp to send a message. | |
| ***Available via*** | V2xGn.h | |

⌋()**[SWS_V2xGn_00095]** ⌈
The function V2xGn_Transmit shall transmit a V2X Packet. ⌋()

**[SWS_V2xGn_00096]** ⌈
If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ⌋()

**[SWS_V2xGn_00098]** ⌈

The function shall return V2X_GNTX_E_MAXSDUSIZEOVFL if the call to EthIf_ProvideTxBuffer returns BUFREQ_E_OVFL. ⌋()

**[SWS_V2xGn_00099]** ⌈
The function shall return V2X_GNTX_E_MAXPACKETLIFETIME if the parameter TxParams.maxPacketLifeTime is invalid. ⌋()

**[SWS_V2xGn_00100]** ⌈
The function shall return V2X_GNTX_E_TCID if the parameter TxParams.trafficClass is invalid. ⌋()

**[SWS_V2xGn_00101]** ⌈
The function shall return V2X_GNTX_E_MAXGEOAREASIZE if the parameter TxParams.destinationType is V2X_GNDESTINATION_AREA and the parameter TxParams.destinationArea is invalid. ⌋()

## 8.4 Call-back notifications

### 8.4.1 This is a list of functions provided for other modules. V2xGn_V2xM_EncapConfirmation

**[SWS_V2xGn_00118]**⌈

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_V2xM_EncapConfirmation | |
| ***Syntax*** | `void V2xGn_V2xM_EncapConfirmation ( uint16 TransactionId16 )` | |
| ***Service ID [hex]*** | 0x0b | |
| ***Sync/Async*** | Asynchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | TransactionId16 | The TransactionId of the encapsulated packet |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function is called by the V2xM when an encapsulation has been finished. | |
| ***Available via*** | V2xGn_V2xM.h | |

⌋()**[SWS_V2xGn_00119]** ⌈
The function V2xGn_V2xM_EncapConfirmation shall finalize the packet transmission by transmitting the packet to the lower layer. ⌋()

**[SWS_V2xGn_00120]** ⌈

If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ]()

Note: The function requires previous successful transmission request via the API V2xGn_Transmit.

### 8.4.2 V2xGn_V2xM_DecapConfirmation

**[SWS_V2xGn_00122]**[

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_V2xM_DecapConfirmation | |
| ***Syntax*** | `void V2xGn_V2xM_DecapConfirmation (`<br>`  uint32 TransactionId32,`<br>`  V2xM_SecReportType SecReport,`<br>`  uint64 CertificateId,`<br>`  uint32 ItsAid,`<br>`  uint8 SspLength,`<br>`  uint8* SspBits`<br>`)` | |
| ***Service ID [hex]*** | 0x0c | |
| ***Sync/Async*** | Asynchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | Transaction Id32 | The TransactionId32 of the decapsulated packet |
| | SecReport | The security report. |
| | CertificateId | The identification of the used for verification (by certificate hash) |
| | ItsAid | The numerical value of the ITS-AID |
| | SspLength | The length (in octets, up to 31) of the SSP bits |
| | SspBits | The SSP bits |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | This function is called by the V2xM when a decapsulation has been finished. | |
| ***Available via*** | V2xGn_V2xM.h | |

]()**[SWS_V2xGn_00123]** [
The function V2xGn_V2xM_DecapConfirmation shall continue the processing of a received packet by proceeding with V2xGn protocol operations. ]()

**[SWS_V2xGn_00124]** [

If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ]()

Note: The function requires previous successful reception of a packet via the API V2xGn_RxIndication.

### 8.4.3 V2xGn_RxIndication

**[SWS_V2xGn_00071]**[

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_RxIndication | |
| ***Syntax*** | `void V2xGn_RxIndication (`<br>`  uint8 CtrlIdx,`<br>`  Eth_FrameType FrameType,`<br>`  boolean IsBroadcast,`<br>`  const uint8* PhysAddrPtr,`<br>`  uint8* DataPtr,`<br>`  uint16 LenByte`<br>`)` | |
| ***Service ID [hex]*** | 0x04 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | CtrlIdx | Index of the Ethernet controller within the context of the Ethernet Interface |
| | FrameType | frame type of received Ethernet frame |
| | Is Broadcast | parameter to indicate a broadcast frame |
| | PhysAddr Ptr | pointer to Physical source address (MAC address in network byte order) of received Ethernet frame |
| | DataPtr | Pointer to payload of the received Ethernet frame (i.e. Ethernet header is not provided). |
| | LenByte | Length of received data. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Indicates the reception of an Ethernet frame | |
| ***Available via*** | V2xGn.h | |

]()**[SWS_V2xGn_00103]** [

The function V2xGn_RxIndication shall get reception parameters of the Wireless Ethernet Driver for a V2X Packet received via an API call to EthIf_GetBufWRxParams. ]()
This is done to get access to the wireless specific reception parameters (e.g. the RSSI or the TransactionId32) of the packet that is not available through the V2xGn_RxIndication API.

**[SWS_V2xGn_00104]** [
If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ]()

**[SWS_V2xGn_00105]** [
If development error detection is enabled: the function shall check the parameter DataPtr for being valid. If the check fails, the function shall raise the development error V2XGN_E_PARAM_POINTER. ]()

### 8.4.4 V2xGn_TxConfirmation

**[SWS_V2xGn_00074]**[

| | | |
|---|---|---|
| ***Service Name*** | V2xGn_TxConfirmation | |
| ***Syntax*** | `void V2xGn_TxConfirmation ( uint8 CtrlIdx, uint8 BufIdx, Std_ReturnType Result )` | |
| ***Service ID [hex]*** | 0x07 | |
| ***Sync/Async*** | Synchronous | |
| ***Reentrancy*** | Non Reentrant | |
| ***Parameters (in)*** | Ctrl Idx | Index of the Ethernet controller within the context of the Ethernet Interface |
| | BufIdx | Index of the buffer resource |
| | Result | Indicates if the PDU was transmitted successfully (E_OK) or not. |
| ***Parameters (inout)*** | None | |
| ***Parameters (out)*** | None | |
| ***Return value*** | None | |
| ***Description*** | Confirms the transmission of an Ethernet frame | |
| ***Available via*** | V2xGn.h | |

]()**[SWS_V2xGn_00107]** [

If development error detection is enabled: the function shall check that the service V2xGn_Init was previously called. If the check fails, the function shall raise the development error V2XGN_E_UNINIT. ]()

## 8.5 Scheduled functions

### 8.5.1 V2xGn_MainFunction

**[SWS_V2xGn_00075]**[

| | |
|---|---|
| ***Service Name*** | V2xGn_MainFunction |
| ***Syntax*** | `void V2xGn_MainFunction ( void )` |
| ***Service ID [hex]*** | 0x08 |
| ***Description*** | Main function of the V2xGn module for periodical execution of protocol operations. |
| ***Available via*** | SchM_V2xGn.h |

]()

## 8.6 Expected Interfaces

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

### 8.6.1 Mandatory Interfaces

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

**[SWS_V2xGn_00076]**[

| ***API Function*** | ***Header File*** | ***Description*** |
|---|---|---|
| EthIf_GetBufWRx-Params | EthIf.h | Read out values related to the receive direction of the transceiver for a received packet. For example, this could be RSSI or Channel belonging to one single packet. |
| EthIf_GetBufWTx-Params | EthIf.h | Read out values related to the transmit direction of the transceiver for a transmitted packet. For example, this could be transaction ID belonging to one single packet. |
| EthIf_ProvideTx-Buffer | EthIf.h | Provides access to a transmit buffer of the specified Ethernet controller. |
| EthIf_SetBufWTx-Params | EthIf.h | Set values related to the transmit direction of the transceiver for a specific buffer (packet to be sent). For example, this can be the desired transmit power or the channel belonging to one single |

| | | |
|---|---|---|
| | | packet. |
| EthIf_Transmit | EthIf.h | Triggers transmission of a previously filled transmit buffer |
| V2xBtp_CopyTx-Data | V2xBtp.h | This API is called by the V2xGn module to request the V2xBtp module to copy the transmission data to a specific location. |
| V2xBtp_Rx-Indication | V2xBtp.h | Via this API, the V2xBtp module gets the data (BTP-PDU), the Geo Networking parameters and the Transaction ID of a received Geo Networking packet. |
| V2xM_Calc-Distance | V2xM.h | Calculates the distance between two geographical points on earth with the assumption that they are on elevation 0. |
| V2xM_Get-PositionAndTime | V2xM.h | Provides the instantaneous position information. |
| V2xM_GetRef-TimePtr | V2xM.h | Provides a pointer to the time reference of the V2X-Stack. |
| V2xM_Trigger-Pseudonym-Change | V2xM.h | This function is called by the V2xFac, V2xGn or another entity to change the Pseudonym used by the V2X-Stack, e.g. due to a Geo Networking address conflict. |
| V2xM_V2xGn_-GetGlobalTx-Params | V2xM_ V2xGn.h | This function is called by V2xGn to get the current channel busy percentage for the specified channel |
| V2xM_V2xGn_-ReqDecap | V2xM_ V2xGn.h | This function is called by the V2xGn to decrypt and verify a message. An asynchronous V2xGn_V2xM_DecapConfirmation call will be used to notify V2xGn of the result. |
| V2xM_V2xGn_-ReqEncap | V2xM_ V2xGn.h | This function is called by the V2xGn to sign and/or encrypt a message. An asynchronous V2xGn_V2xM_EncapConfirmation call will be used to notify V2xGn of the result. |
| V2xM_V2xGn_-SetGlobalRx-Params | V2xM_ V2xGn.h | This function is called by V2xGn to set the current channel busy percentage for the specified channel |

]()

### 8.6.2 Optional Interfaces

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

**[SWS_V2xGn_00077]**[

| *API Function* | *Header File* | *Description* |
|---|---|---|
| Det_ReportError | Det.h | Service to report development errors. |
| V2xBtp_Tx-Confirmation | V2xBtp.h | By this API primitive, the V2xBtp module gets an indication from the V2xGn module about the status of the transmission of the data (FAC-PDU) with the associated ID. |

]()

# 9 Sequence diagrams

The following sequence diagrams show the interactions between the V2xGn module and its adjacent modules.

Please note that the sequence diagrams are an extension for illustrational purposes to ease understanding of the specification and to support the functional specification described in chapter 7 and API specification described in chapter 8.

Note that all parameters and return types are left out to make the diagrams easier to read and understand.

## 9.1 V2xGn_RxIndication

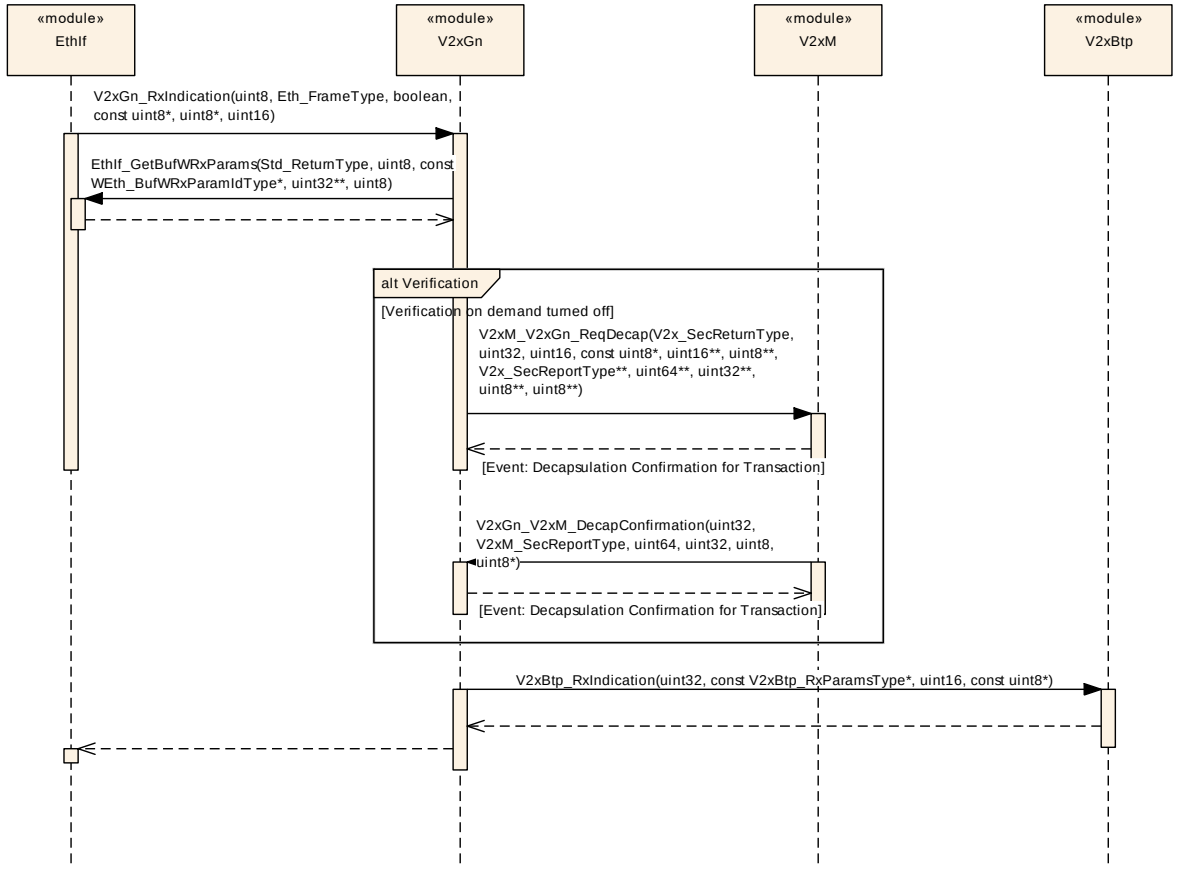



**Figure 2 –V2xGn_RxIndication**

## 9.2 V2xGn_Transmit

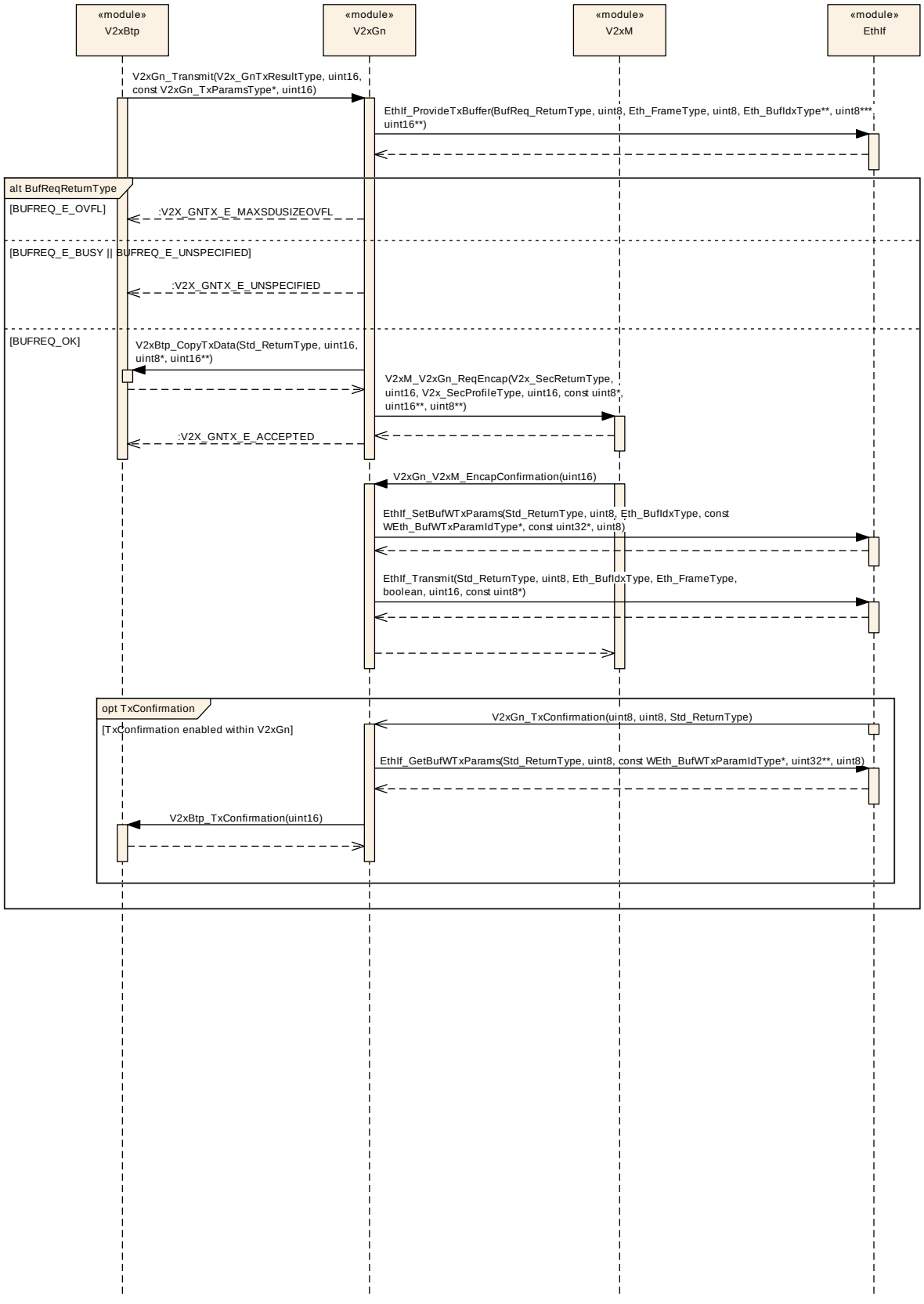



**Figure 3 – V2xGn_Transmit**

## 9.3 V2xGn_V2xM_UpdatePseudonym

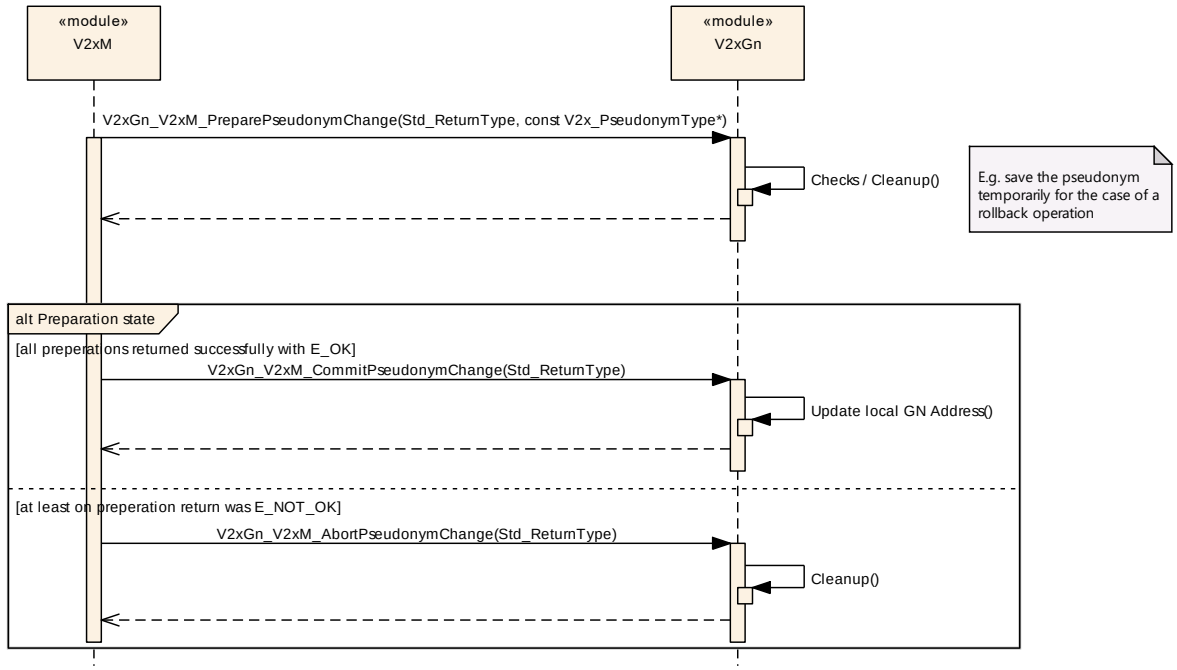



**Figure 4 – V2xGn_V2xM_UpdatePseudonym**

## 9.4 V2xGn_MainFunction

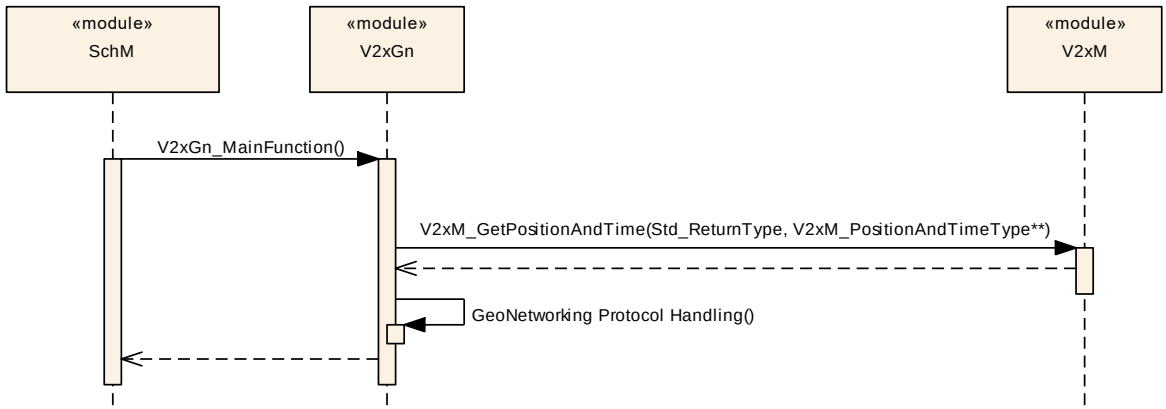



**Figure 5 – V2xGn_MainFunction**

# 10 Configuration specification

Chapter 10.1 specifies the structure (containers) and the parameters of the module V2xGn.

Chapter 0 specifies additionally published information of the module V2xGn.

## 10.1 Containers and configuration parameters

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

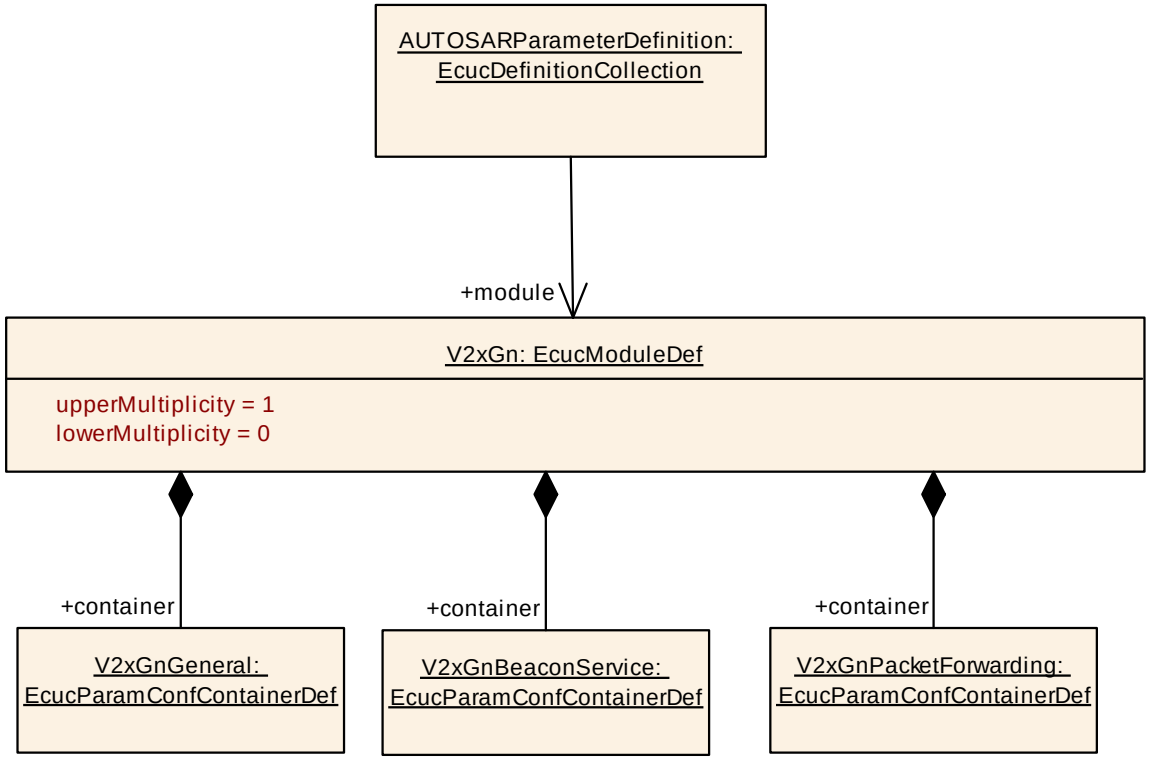
### 10.1.1 Variants

**[SWS_V2xGn_00078]** [ The V2xGn module only supports VARIANT-PRE-COMPILE ] (SRS_BSW_00345)

### 10.1.2 V2xGn

| **SWS Item** | [ECUC_V2xGn_00001] |
|---|---|
| **Module Name** | V2xGn |
| **Description** | Configuration of the V2xGn (Vehicle-2-X Geo Networking) module. |
| **Post-Build Variant Support** | false |
| **Supported Config Variants** | VARIANT-PRE-COMPILE |

| **Included Containers** | | |
|---|---|---|
| **Container Name** | **Multiplicity** | **Scope / Dependency** |
| V2xGnBeacon-Service | 1 | This container contains the GeoNetworking configuration parameters related to the beacon service. |
| V2xGnGeneral | 1 | This container specifies the general configuration parameters of the V2xGn module. |
| V2xGnPacket-Forwarding | 1 | This container contains the GeoNetworking configuration parameters related to packet forwarding. |




### 10.1.3 V2xGnGeneral

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00002] |
| **Container Name** | V2xGnGeneral |
| **Parent Container** | V2xGn |
| **Description** | This container specifies the general configuration parameters of the V2xGn module. |
| **Configuration Parameters** | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00006] |
| **Parameter Name** | V2xGnDevErrorDetect |
| **Parent Container** | V2xGnGeneral |
| **Description** | Switches the Default Error Tracer (Det) detection and notification ON or OFF.<br>• true: enabled (ON)<br>• false: disabled (OFF) |
| **Multiplicity** | 1 |

| | | | |
|---|---|---|---|
| **Type** | EcucBooleanParamDef | | |
| **Default value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00016] | | |
| **Parameter Name** | V2xGnItsGnLifetimeLocTE | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Location table maintenance: Lifetime of an entry in the location table in [s] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | [0 .. 65535] | | |
| **Default value** | 20 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local<br>dependency: shall be a multiple of the V2xGnMainFunctionPeriod | | |

| | | |
|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00009] | |
| **Parameter Name** | V2xGnItsGnLocalGnAddr | |
| **Parent Container** | V2xGnGeneral | |
| **Description** | 64bit GeoNetworking Address. | |
| **Multiplicity** | 1 | |
| **Type** | EcucIntegerParamDef | |
| **Range** | 0 .. 18446744073709551615 | |
| **Default value** | 1 | |
| **Post-Build Variant Value** | false | |

| | | | |
|---|---|---|---|
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00014] | | |
| **Parameter Name** | V2xGnItsGnMaxGeoNetworkingHeaderSize | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Maximum size of GeoNetworking header in [Byte]. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 65535 | | |
| **Default value** | 88 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00013] | | |
| **Parameter Name** | V2xGnItsGnMaxSduSize | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Maximum size of GN-SDU in [Byte]. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 65535 | | |
| **Default value** | 1398 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00011] | | |
| **Parameter Name** | V2xGnItsGnMinUpdateFrequencyEPV | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Minimum update frequency of ego position vector (EPV) in [s]. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | [0 .. 65535] | | |
| **Default value** | -- | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local<br>dependency: shall be a multiple of the V2xGnMainFunctionPeriod | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00012] | | |
| **Parameter Name** | V2xGnItsGnPaiInterval | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Distance related to the confidence interval for latitude and longitude [m]. Used to determine the PAI. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 100 | | |
| **Default value** | 80 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00008] |

| | | | |
|---|---|---|---|
| **Parameter Name** | V2xGnItsGnProtocolVersion | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | GeoNetworking protocol version as defined in Annex H of [14] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 255 | | |
| **Default value** | 1 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00017] | | |
| **Parameter Name** | V2xGnItsGnSnDecapResultHandling | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Indicates the handling of the V2xM_ReqDecap result code. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | V2XGN_NON_STRICT_SEC_HANDLING | GN packets that are not correctly verified and decrypted can be passed to the upper protocol entity for further processing. | |
| | V2XGN_STRICT_SEC_HANDLING | Received GN packets that are not correctly verified and decrypted are always dropped. | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00015] |
| **Parameter Name** | V2xGnItsGnStationType |

| | | | |
|---|---|---|---|
| **Parent Container** | V2xGnGeneral | | |
| **Description** | Station Type used in GeoNetworking protocol, RoadSideUnit (15) not supported by AUTOSAR. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucEnumerationParamDef | | |
| **Range** | V2XFAC_ST_BUS | -- | |
| | V2XFAC_ST_CYCLIST | -- | |
| | V2XFAC_ST_HEAVYTRUCK | -- | |
| | V2XFAC_ST_LIGHTTRUCK | -- | |
| | V2XFAC_ST_MOPED | -- | |
| | V2XFAC_ST_MOTORCYCLE | -- | |
| | V2XFAC_ST_PASSENGERCAR | -- | |
| | V2XFAC_ST_PEDESTRIAN | -- | |
| | V2XFAC_ST_SPECIALVEHICLES | -- | |
| | V2XFAC_ST_TRAILER | -- | |
| | V2XFAC_ST_TRAM | -- | |
| | V2XFAC_ST_UNKNOWN | -- | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | |
|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00018] | |
| **Parameter Name** | V2xGnMainFunctionPeriod | |
| **Parent Container** | V2xGnGeneral | |
| **Description** | Specifies the period of main function V2xGn_MainFunction in seconds. V2x Gn does not require this information but the BSW scheduler. | |
| **Multiplicity** | 1 | |
| **Type** | EcucFloatParamDef | |
| **Range** | ]0 .. INF[ | |
| **Default value** | 0.001 | |

| | | | |
|---|---|---|---|
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

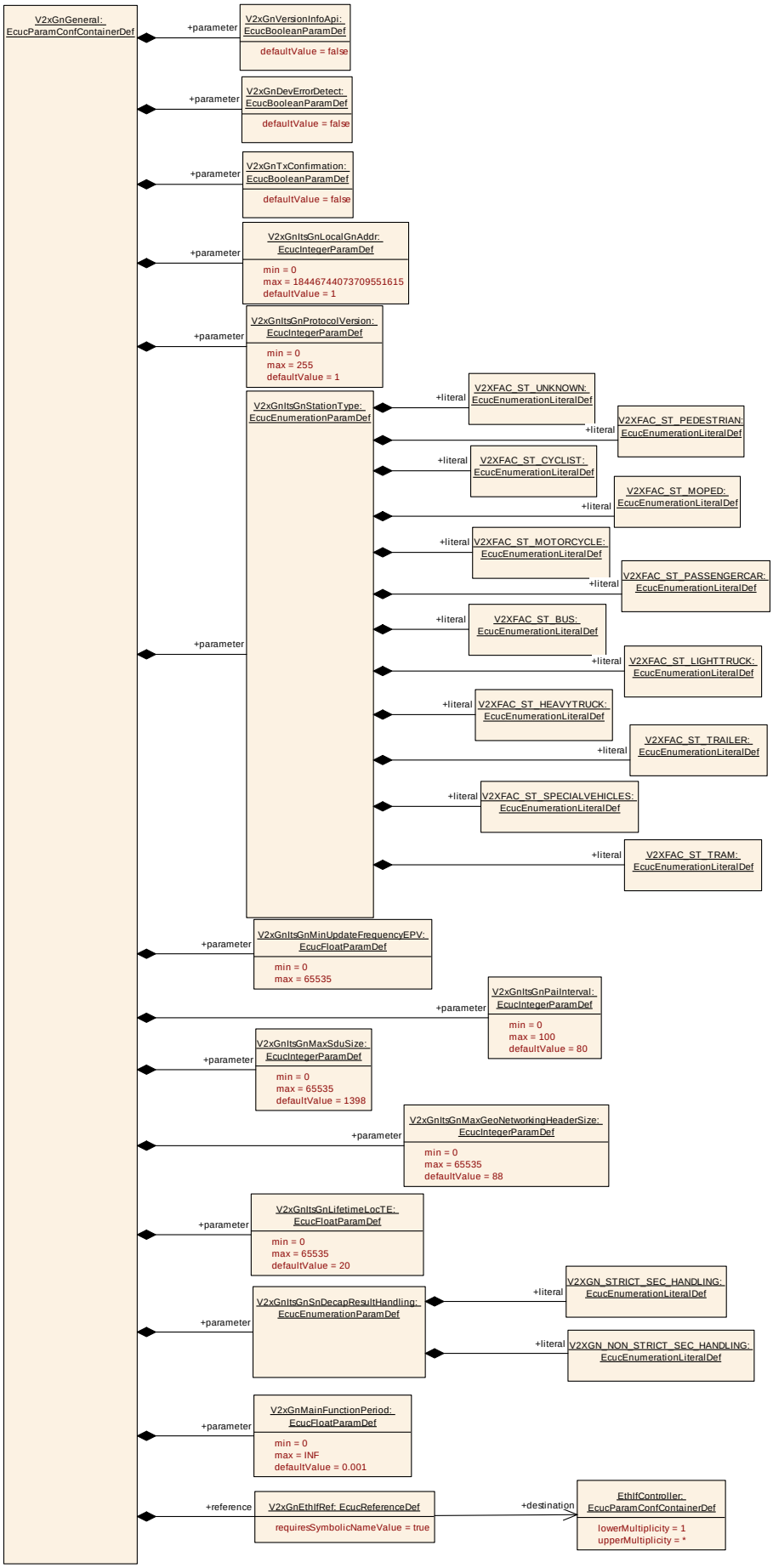
| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00007] | | |
| **Parameter Name** | V2xGnTxConfirmation | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | When enabled, transmission status information will be forwarded to the upper layer.<br>• true: enabled (ON)<br>• false: disable (OFF) | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: ECU | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00005] |
| **Parameter Name** | V2xGnVersionInfoApi |
| **Parent Container** | V2xGnGeneral |
| **Description** | Enable/disables the API for reading the version information of the V2xGn Module.<br>• true: enabled (ON)<br>• false: disabled (OFF) |
| **Multiplicity** | 1 |

| | | | |
|---|---|---|---|
| **Type** | EcucBooleanParamDef | | |
| **Default value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00019] | | |
| **Parameter Name** | V2xGnEthIfRef | | |
| **Parent Container** | V2xGnGeneral | | |
| **Description** | This represents the reference to the Ethernet interface taken to transmit the V2X-Packets to. | | |
| **Multiplicity** | 1 | | |
| **Type** | Symbolic name reference to EthIfController | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

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


V2xGnGeneral: EcucParamConfContainerDef
+parameter
V2xGnVersionInfoApi: EcucBooleanParamDef
defaultValue = false
V2xGnDevErrorDetect: EcucBooleanParamDef
defaultValue = false
V2xGnTxConfirmation: EcucBooleanParamDef
defaultValue = false
V2xGnItsGnLocalGnAddr: EcucIntegerParamDef
min = 0
max = 18446744073709551615
defaultValue = 1
V2xGnItsGnProtocolVersion: EcucIntegerParamDef
min = 0
max = 255
defaultValue = 1
V2xGnItsGnStationType: EcucEnumerationParamDef
+literal
V2XFAC_ST_UNKNOWN: EcucEnumerationLiteralDef
V2XFAC_ST_PEDESTRIAN: EcucEnumerationLiteralDef
V2XFAC_ST_CYCLIST: EcucEnumerationLiteralDef
V2XFAC_ST_MOPED: EcucEnumerationLiteralDef
V2XFAC_ST_MOTORCYCLE: EcucEnumerationLiteralDef
V2XFAC_ST_PASSENGERCAR: EcucEnumerationLiteralDef
V2XFAC_ST_BUS: EcucEnumerationLiteralDef
V2XFAC_ST_LIGHTTRUCK: EcucEnumerationLiteralDef
V2XFAC_ST_HEAVYTRUCK: EcucEnumerationLiteralDef
V2XFAC_ST_TRAILER: EcucEnumerationLiteralDef
V2XFAC_ST_SPECIALVEHICLES: EcucEnumerationLiteralDef
V2XFAC_ST_TRAM: EcucEnumerationLiteralDef
V2xGnItsGnMinUpdateFrequencyEPV: EcucFloatParamDef
min = 0
max = 65535
V2xGnItsGnPaiInterval: EcucIntegerParamDef
min = 0
max = 100
defaultValue = 80
V2xGnItsGnMaxSduSize: EcucIntegerParamDef
min = 0
max = 65535
defaultValue = 1398
V2xGnItsGnMaxGeoNetworkingHeaderSize: EcucIntegerParamDef
min = 0
max = 65535
defaultValue = 88
V2xGnItsGnLifetimeLocTE: EcucFloatParamDef
min = 0
max = 65535
defaultValue = 20
V2xGnItsGnSnDecapResultHandling: EcucEnumerationParamDef
V2XGN_STRICT_SEC_HANDLING: EcucEnumerationLiteralDef
V2XGN_NON_STRICT_SEC_HANDLING: EcucEnumerationLiteralDef
V2xGnMainFunctionPeriod: EcucFloatParamDef
min = 0
max = INF
defaultValue = 0.001
+reference
V2xGnEthIfRef: EcucReferenceDef
requiresSymbolicNameValue = true
+destination
EthIfController: EcucParamConfContainerDef
lowerMultiplicity = 1
upperMultiplicity = *

### 10.1.4 V2xGnBeaconService

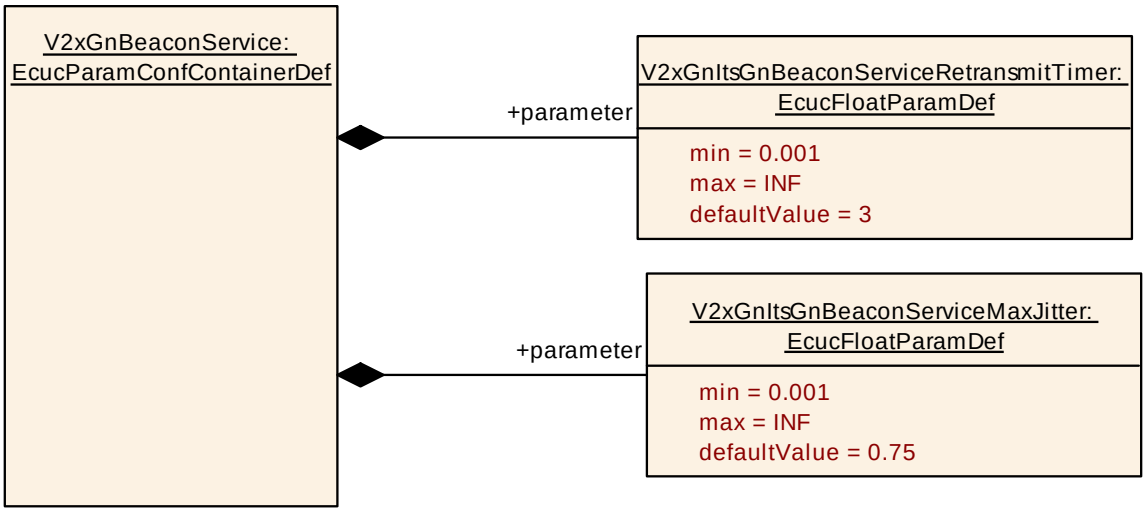
| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00003] |
| **Container Name** | V2xGnBeaconService |
| **Parent Container** | V2xGn |
| **Description** | This container contains the GeoNetworking configuration parameters related to the beacon service. |
| **Configuration Parameters** | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00021] | | |
| **Parameter Name** | V2xGnItsGnBeaconServiceMaxJitter | | |
| **Parent Container** | V2xGnBeaconService | | |
| **Description** | Maximum beacon jitter [s]. The Jitter is used for the beacon retransmission. The actual jitter value is a random number between 0 and V2xGnItsGnBeaconService MaxJitter. The function introduces a random component for the timer to avoid synchronization issues among GeoAdhoc routers. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | [0.001 .. INF] | | |
| **Default value** | 0.75 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00020] |
| **Parameter Name** | V2xGnItsGnBeaconServiceRetransmitTimer |
| **Parent Container** | V2xGnBeaconService |
| **Description** | Duration of Beacon service retransmit timer [s]. |
| **Multiplicity** | 1 |

| | | | |
|---|---|---|---|
| **Type** | EcucFloatParamDef | | |
| **Range** | [0.001 .. INF] | | |
| **Default value** | 3 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local<br>dependency: shall be a multiple of the V2xGnMainFunctionPeriod. | | |

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




### 10.1.5 V2xGnPacketForwarding

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00004] |
| **Container Name** | V2xGnPacketForwarding |
| **Parent Container** | V2xGn |
| **Description** | This container contains the GeoNetworking configuration parameters related to packet forwarding. |
| **Configuration Parameters** | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00032] |

| | | | |
|---|---|---|---|
| **Parameter Name** | V2xGnItsGnBcForwardingPacketBufferSize | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Size of BC forwarding packet buffer [Byte]. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 1024000 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00029] | | |
| **Parameter Name** | V2xGnItsGnCbfMaxTime | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Maximum duration a GeoNetworking packet shall be buffered in the CBF packet buffer [s] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | ]0 .. INF[ | | |
| **Default value** | 0.001 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00028] |
| **Parameter Name** | V2xGnItsGnCbfMinTime |
| **Parent Container** | V2xGnPacketForwarding |
| **Description** | Minimum duration a GeoNetworking packet shall be buffered in the CBF |

| | | | |
|---|---|---|---|
| | packet buffer [s] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | ]0 .. INF[ | | |
| **Default value** | 0.001 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00033] | | |
| **Parameter Name** | V2xGnItsGnCbfPacketBufferSize | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Size of CBF packet buffer [Byte] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 256000 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | |
|---|---|
| **SWS Item** | [ECUC_V2xGn_00022] |
| **Parameter Name** | V2xGnItsGnDefaultHopLimit |
| **Parent Container** | V2xGnPacketForwarding |
| **Description** | Default hop limit indicating the maximum number of hops a packet travels. |
| **Multiplicity** | 1 |
| **Type** | EcucIntegerParamDef |

| | | | |
|---|---|---|---|
| **Range** | 0 .. 255 | | |
| **Default value** | 10 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00030] | | |
| **Parameter Name** | V2xGnItsGnDefaultMaxCommunicationRange | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Default theoretical maximum communication range [m] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 65535 | | |
| **Default value** | 1000 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00024] | | |
| **Parameter Name** | V2xGnItsGnDefaultPacketLifetime | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Default packet lifetime [s]. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | [0 .. 6300] | | |
| **Default value** | 60 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |

| | | | |
|---|---|---|---|
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00034] | | |
| **Parameter Name** | V2xGnItsGnDefaultTrafficClass | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Forwarding: Default traffic class | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 255 | | |
| **Default value** | 0 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00035] | | |
| **Parameter Name** | V2xGnItsGnDplLength | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Length of Duplicate Packet List (DPL) per source (clause A.2 of [18]) | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 65535 | | |
| **Default value** | 8 | | |
| **Post-Build Variant Multiplicity** | true | | |
| **Multiplicity Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00031] | | |
| **Parameter Name** | V2xGnItsGnGeoAreaLineForwardingUsed | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Forwarding of GBC/GAC packet if GeoAdhoc router is located outside the destination GeoArea.<br>• true: enabled (ON)<br>• false: disabled (OFF) | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucBooleanParamDef | | |
| **Default value** | false | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00027] | | |
| **Parameter Name** | V2xGnItsGnMaxGeoAreaSize | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Maximum size of the geographical area for a GBC and GAC packet [km2]. If the geographical area size exceeds the maximum value, the GeoNetworking packet shall not be sent (source) and not be forwarded (forwarder). | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 18446744073709551615 | | |
| **Default value** | 80 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |

| | |
|---|---|
| **Scope / Dependency** | scope: local |

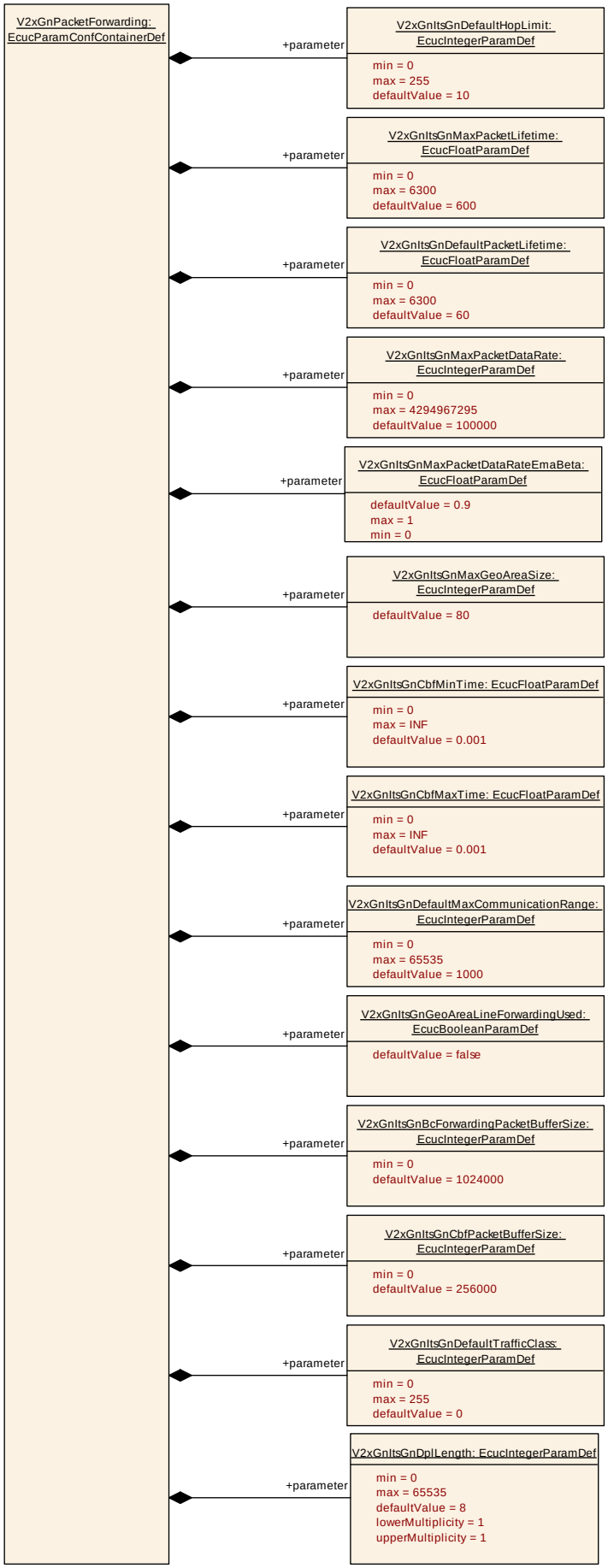
| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00025] | | |
| **Parameter Name** | V2xGnItsGnMaxPacketDataRate | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Maximum packet data rate for a GeoAdhoc router [Byte/s]. If the mean (EMA) packet data rate a of a GeoAdhoc router exceeds the value, packets from this GeoAdhoc router (source or sender) are not forwarded. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucIntegerParamDef | | |
| **Range** | 0 .. 4294967295 | | |
| **Default value** | 100000 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00026] | | |
| **Parameter Name** | V2xGnItsGnMaxPacketDataRateEmaBeta | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Weight factor for the Exponential Moving Average of the packet data rate PDR in percent. | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | ]0 .. 1] | | |
| **Default value** | 0.9 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |

| | |
|---|---|
| **Scope / Dependency** | scope: local |

| | | | |
|---|---|---|---|
| **SWS Item** | [ECUC_V2xGn_00023] | | |
| **Parameter Name** | V2xGnItsGnMaxPacketLifetime | | |
| **Parent Container** | V2xGnPacketForwarding | | |
| **Description** | Upper limit of the maximum lifetime [s] | | |
| **Multiplicity** | 1 | | |
| **Type** | EcucFloatParamDef | | |
| **Range** | [0 .. 6300] | | |
| **Default value** | 600 | | |
| **Post-Build Variant Value** | false | | |
| **Value Configuration Class** | **Pre-compile time** | X | All Variants |
| | **Link time** | -- | |
| | **Post-build time** | -- | |
| **Scope / Dependency** | scope: local | | |

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


V2xGnPacketForwarding: EcucParamConfContainerDef
+parameter
V2xGnItsGnDefaultHopLimit: EcucIntegerParamDef
min = 0
max = 255
defaultValue = 10
+parameter
V2xGnItsGnMaxPacketLifetime: EcucFloatParamDef
min = 0
max = 6300
defaultValue = 600
+parameter
V2xGnItsGnDefaultPacketLifetime: EcucFloatParamDef
min = 0
max = 6300
defaultValue = 60
+parameter
V2xGnItsGnMaxPacketDataRate: EcucIntegerParamDef
min = 0
max = 4294967295
defaultValue = 100000
+parameter
V2xGnItsGnMaxPacketDataRateEmaBeta: EcucFloatParamDef
defaultValue = 0.9
max = 1
min = 0
+parameter
V2xGnItsGnMaxGeoAreaSize: EcucIntegerParamDef
defaultValue = 80
+parameter
V2xGnItsGnCbfMinTime: EcucFloatParamDef
min = 0
max = INF
defaultValue = 0.001
+parameter
V2xGnItsGnCbfMaxTime: EcucFloatParamDef
min = 0
max = INF
defaultValue = 0.001
+parameter
V2xGnItsGnDefaultMaxCommunicationRange: EcucIntegerParamDef
min = 0
max = 65535
defaultValue = 1000
+parameter
V2xGnItsGnGeoAreaLineForwardingUsed: EcucBooleanParamDef
defaultValue = false
+parameter
V2xGnItsGnBcForwardingPacketBufferSize: EcucIntegerParamDef
min = 0
defaultValue = 1024000
+parameter
V2xGnItsGnCbfPacketBufferSize: EcucIntegerParamDef
min = 0
defaultValue = 256000
+parameter
V2xGnItsGnDefaultTrafficClass: EcucIntegerParamDef
min = 0
max = 255
defaultValue = 0
+parameter
V2xGnItsGnDplLength: EcucIntegerParamDef
min = 0
max = 65535
defaultValue = 8
lowerMultiplicity = 1
upperMultiplicity = 1

## 10.2 Published Information

For details refer to the chapter 10.3 “Published Information” in [3].

# 11 Not applicable requirements

**[SWS_V2xGn_NA_00001]** ⌈ This requirement references all not applicable access layer requirements ⌋ (SRS_V2X_00451, SRS_V2X_00322, SRS_V2X_00242, SRS_V2X_00391, SRS_V2X_00232, SRS_V2X_00245)

**[SWS_V2xGn_NA_00002]** ⌈ This requirement references all not applicable facility layer requirements ⌋ (SRS_V2X_00711, SRS_V2X_00291, SRS_V2X_00318, SRS_V2X_00741, SRS_V2X_00301)

**[SWS_V2xGn_NA_00003]** ⌈ This requirement references all not applicable security requirements ⌋ (SRS_V2X_00405, SRS_V2X_00413, SRS_V2X_00163, SRS_V2X_00412, SRS_V2X_00407, SRS_V2X_00406, SRS_V2X_00184, SRS_V2X_00174)

**[SWS_V2xGn_NA_00004]** ⌈ This requirement references all not applicable other requirements from SRS V2X ⌋ (SRS_V2X_00190, SRS_V2X_00193, SRS_V2X_00207, SRS_V2X_00214, SRS_V2X_00693, SRS_V2X_00189, SRS_V2X_00323, SRS_V2X_00511)