

<b>Document Title</b>	Specification of Vehicle-2-X Facilities
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	795

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	R22-11

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• New architecture with V2x DM</li> </ul>
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Editorial Changes</li> </ul>
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Corrections of typos in chap. 8 definitions</li> <li>• Editorial Changes</li> </ul>
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Added support for C2CCC BSP 1.3</li> <li>• Editorial Changes</li> <li>• Service API bugs corrections</li> <li>• Changed Document Status from Final to published</li> </ul>
2018-10-31	4.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Added IVIM supporting</li> <li>• Added SPATEM supporting</li> <li>• Added MAPEM supporting</li> </ul>
2017-12-08	4.3.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Editorial Changes</li> </ul>
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Initial Release</li> </ul>

## 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
1.2.1	Cooperative Awareness (CA).....	7
1.2.2	Decentralized Environmental Notification (DEN).....	7
1.2.3	Vehicle Data Provider (VDP).....	8
1.2.4	Local Dynamic Map (LDM) .....	8
1.2.5	Infrastructure to Vehicle Information (IVI) .....	9
1.2.6	Road and Lane Topology (RLT) service .....	9
1.2.7	Traffic Light Maneuver (TLM) service .....	10
2	Acronyms and abbreviations .....	11
3	Related documentation.....	12
3.1	Input documents .....	12
3.2	Related standards and norms .....	12
3.3	Related specification .....	13
4	Constraints and assumptions .....	15
4.1	Limitations .....	15
4.2	Applicability to car domains .....	15
4.3	Authorisation Tickets and Pseudonyms.....	15
5	Dependencies to other modules .....	16
5.1	AUTOSAR DET (Default Error Tracer).....	16
5.2	AUTOSAR EcuM (Ecu State Manager) .....	16
5.3	V2x Vehicle Data Provider.....	16
5.4	AUTOSAR V2xDM .....	16
5.5	AUTOSAR V2xBtp.....	16
6	Requirements traceability .....	17
7	Functional specification .....	19
7.1	Startup behavior .....	20
7.2	General Format Specification .....	20
7.3	CA Functional Specification.....	20
7.3.1	CA Initialization, Activation and Deactivation .....	20
7.3.2	CAM Generation, Sending and Frequency Management .....	21
7.3.3	CAM Generation Frequency Management for RSU ITS-Ss.....	22
7.3.4	CAM Time Requirement .....	22
7.3.5	CAM Format Specification .....	23
7.3.6	Others .....	24
7.4	DEN Functional Specification .....	24
7.4.1	DEN Initialization.....	24
7.4.2	DENM Transmission Management .....	25
7.4.3	DENM Reception Management .....	25
7.4.4	DENM Repetition .....	25
7.4.5	DENM Keep Alive Forwarding (KAF) .....	25

7.4.6	DENM Format Specification.....	25
7.5	Message Reception Functional Specification .....	27
7.6	Path History .....	27
7.7	Error classification .....	28
7.7.1	Development Errors .....	28
7.7.2	Runtime Errors.....	28
7.7.3	Transient Faults .....	28
7.7.4	Production Errors.....	29
7.7.5	Extended Production Errors.....	29
8	API specification.....	30
8.1	Imported types.....	30
8.2	Type definitions .....	30
8.2.1	V2xFac_RxParamsType .....	30
8.3	Function definitions.....	32
8.3.1	V2xFac_Init.....	32
8.3.2	V2xFac_GetVersionInfo.....	33
8.3.3	V2xFac_V2xM_PreparePseudonymChange .....	33
8.3.4	V2xFac_V2xM_CommitPseudonymChange.....	34
8.3.5	V2xFac_V2xM_AbortPseudonymChange.....	35
8.3.6	V2xFac_V2xM_SetTGenCamDcc.....	36
8.3.7	V2xFac_V2xM_SetCaBsOperation.....	37
8.4	Call-back notifications.....	37
8.4.1	V2xFac_TxConfirmation .....	38
8.4.2	V2xFac_RxIndication .....	38
8.5	Scheduled functions .....	39
8.5.1	V2xFac_CaBs_MainFunction.....	39
8.5.2	V2xFac_DenBs_MainFunction.....	40
8.5.3	V2xFac_RxS_MainFunction .....	40
8.6	Expected Interfaces.....	40
8.6.1	Mandatory Interfaces .....	41
8.6.2	Optional Interfaces.....	41
8.7	Service Interfaces.....	41
8.7.1	Sender-Receiver-Interfaces .....	41
8.7.2	Client-Server-Interfaces .....	42
8.7.3	Implementation Data Types.....	47
9	Sequence diagrams .....	76
9.1	CAM Generation and Transmission.....	76
9.2	DENM Generation and Transmission .....	77
9.3	Message Reception.....	79
10	Configuration specification.....	80
10.1	Containers and configuration parameters .....	80
10.1.1	Variants .....	80
10.1.2	V2xFac .....	80
10.1.3	V2xFacGeneral.....	81
10.1.4	V2xFacConfig .....	86
10.1.5	V2xFacDmMsgConfig.....	86
11	Not applicable requirements .....	90

# 1 Introduction and functional overview

This document specifies the functionality, API and the configuration of the AUTOSAR Basic Software module Vehicle-2-X Facilities (V2xFac). The Vehicle-2-X Facilities layer together with the Vehicle-2-X Basic Transport (V2xBtp), the Vehicle-2-X GeoNetworking (V2xGn), Vehicle-2-X Management (V2xM), Vehicle-2-X Data Manager (V2xDM) and the communication driver layer forms the V2X stack within the AUTOSAR architecture.

The V2xFac module is designed to be hardware independent. The V2x Facilities layer of the ETSI architecture is split into the two AUTOSAR modules V2xFac and V2xDM.

The V2xFac module is dependent on services of V2X entities in the application layer and on lower V2xBtp module, and provides services to the V2xDM module.

## 1.1 Architectural overview

Positioning of the V2xFac module within the AUTOSAR BSW and the Layered Software architecture is shown in below.

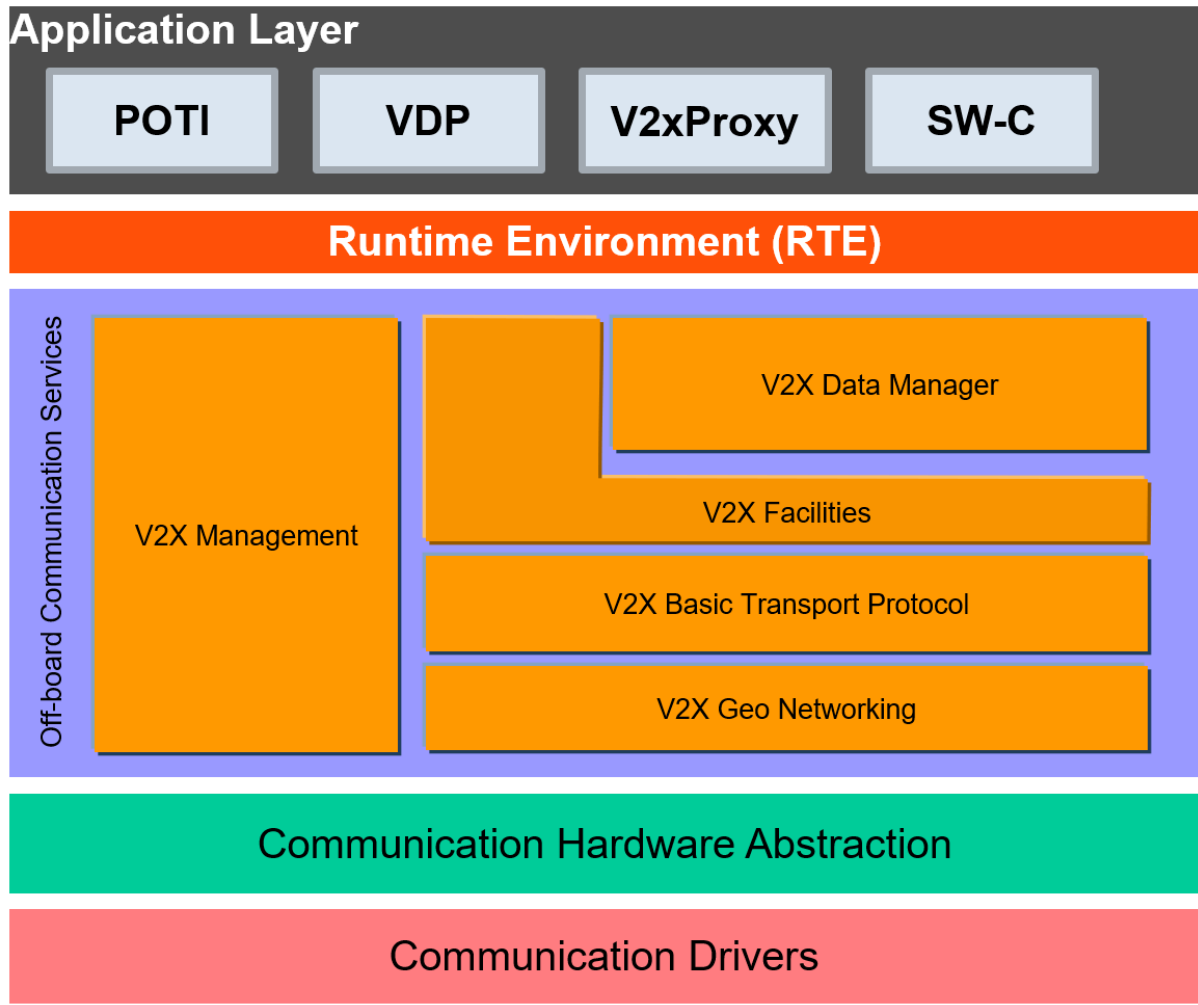


Figure 1 – AUTOSAR BSW software architecture - V2xFac scope

The V2xFac module supports common message management for data exchange between V2X applications.

It provides the basic services (BS) Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN) for transmission. In reception, it analyses the header of the received messages and provides the payload of the supported messages to the V2xDM module.

## 1.2 Functional overview

The V2xFac module implements the basic services CA and DEN for transmission. For reception, the header analysis is supported for the following services:

- CA
- DEN
- IVI
- RLT
- TLM.

Messages are provided to the V2XDM based on configuration.

Each type of message can be systematically discarded based on the configuration.

By default, CAM, DENM, IVIM, SPATEM and MAPEM messages can be received and their ITS header inspected before being provided to the V2xDM for disassembling. Alternatively, by configuration based on BTP port and ITS AID, other messages can be provided to the V2xDM module instead of being discarded.

## **1.2.1 Cooperative Awareness (CA)**

### **1.2.1.1 CA basic service in the AUTOSAR architecture**

The CA basic service is a facilities layer entity that operates the CAM protocol. It provides two services: sending and receiving of CAMs.

The CA basic service generates and sends CAMs to other ITS-Ss or it receives CAMs from ITS-Ss and provides their payload to the V2xDM module (see [10] chapter 4).

The CA basic service uses the services provided by the protocol entities of the lower layers of the V2X stack to disseminate the CAM.

Upon receiving a CAM, the CA basic service makes the content of the CAM available to the V2xDM module.

Selected sets of data of the received CAMs are given to the upper Application layer by the V2xDM module via the standardized AUTOSAR service interface.

It may interface with the AUTOSAR application layer in order to collect relevant information for CAM generation (Vehicle Data Provider - VDP).

### **1.2.1.2 CA basic service functional architecture**

“The CA basic service is part of the Application Support domain of the Facilities Layer according to ETSI TS 102 894-1 [12] shows the functional block diagram with the functional blocks of the CA basic service and interfaces to other facilities and layers.”

For sending and receiving CAMs, the CA basic service part of the V2xFac shall provide the following sub-functions

- Encode CAM
- Decode CAM headers
- CAM transmission management
- CAM reception management

For details see [10] chapter 5.2.

## **1.2.2 Decentralized Environmental Notification (DEN)**

### **1.2.2.1 DEN basic service in the AUTOSAR architecture**

The DEN basic service is a facilities layer entity that operates the DENM protocol. It provides services to entities at the AUTOSAR application layer and to the V2xDM module. (refer to [11] chapter 4.2)

The DEN basic service generates and sends DENMs to other ITS-Ss or it receives DENMs from other ITS-Ss and provides their payload to the V2xDM module (see [11] chapter 5 and 6).

Upon receiving a DENM, the DEN basic service checks the header of the DENM before providing it to the V2xDM module which makes selected content available to the V2X Applications.

### **1.2.2.2 DEN basic service functional architecture**

For sending and receiving DENMs, the DEN basic service shall provide the following sub-functions

- Encode DEN
- Decode DEN header
- DEN transmission management
- DEN reception management
- Keep-Alive forwarding

For Details see [11] chapter 5.3. Position and Time management (POTI)

The POTI, as specified in ETSI TS 102 890-3 [14], provides the position of the ITS-S and time information.

Within the AUTOSAR architecture POTI service is a V2X Application within the Application layer and is not part of V2xFac.

For details See [11] chapter 5.1.

### **1.2.3 Vehicle Data Provider (VDP)**

“The VDP is connected with the vehicle network and provides the vehicle status information.”

Within the AUTOSAR architecture VDP service is a V2X Application within the Application layer and is not part of V2xFac.

The VDP provides an interface to the lower layer (V2X Services).

The facilities basic services CA and DEN get vehicle relevant data from this interface. The V2xM gets e.g. position and time information from this interface.

### **1.2.4 Local Dynamic Map (LDM)**

The LDM as outlined in [15] is a database in the ITS-S, which may be updated with received CAM or DENM data.

V2x applications may retrieve information from the LDM for further processing.

Within the AUTOSAR architecture LDM service is a V2X Application within the Application layer and is not part of the V2xFac module.

For details see [15] chapter 5.1.



## **1.2.5 Infrastructure to Vehicle Information (IVI)**

### **1.2.5.1 IVI service in the AUTOSAR architecture**

The IVI service is a facilities layer entity that provides receiving of IVIMs. The IVI service receives IVIMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [20] chapter 7). Upon receiving an IVIM payload, the V2xDM module makes selected content of the IVIM available to the V2X Applications. Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

### **1.2.5.2 IVI service functional architecture**

The IVI service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [20] which shows the functional block diagram with the functional blocks of the IVI service and interfaces to other facilities and layers.

For receiving IVIMs, the IVI service part of the V2xFac shall provide the following sub-functions

- Decode IVIM header
- IVI reception management

## **1.2.6 Road and Lane Topology (RLT) service**

### **1.2.6.1 RLT service in the AUTOSAR architecture**

The RLT service is a facilities layer entity that provides receiving of MAPEMs. The RLT service receives MAPEMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [20] chapter 6). Upon receiving a MAPEM payload, the V2xDM module makes selected content of the MAPEM available to the V2X Applications. Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

### **1.2.6.2 RLT service functional architecture**

The RLT service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [20] shows the functional block diagram with the functional blocks of the RLT services and interfaces to other facilities and layers.

For receiving MAPEMs, the RLT service part of the V2xFac shall provide the following sub-functions

- Decode MAPEM header
- MAPEM reception management

## **1.2.7 Traffic Light Maneuver (TLM) service**

### **1.2.7.1 TLM service in the AUTOSAR architecture**

The TLM service is a facilities layer entity that provides receiving of SPATEMs. The TLM service receives SPATEMs from Infrastructure ITS-Ss and checks their header before providing their payload to the V2xDM module (see [20] chapter 5). Upon receiving a SPATEM payload, the V2xDM module makes selected content of the SPATEM available to the V2X Applications.

Since these messages do not require any separate processing, their reception in AUTOSAR is implemented via a generic service (see 8.5.3).

### **1.2.7.2 TLM service functional architecture**

The TLM service is part of the Application Support domain of the Facilities Layer according to ETSI TS 103 301 [20] which shows the functional blocks of the TLM services and interfaces to other facilities and layers.

For receiving SPATEMs, the TLM service part of the V2xFac shall provide the following sub-functions

- Decode SPATEM header
- SPATEM reception management

## 2 Acronyms and abbreviations

<b>Abbreviation / Acronym:</b>	<b>Description:</b>
DEM	Diagnostic Event Manager
DET	Default Error Tracer
API	Application Programming Interface
BS	Basic Service
BSW	Basic Software
BTP	Basic Transport Protocol
CA	Cooperative Awareness
CAM	Cooperative Awareness Message
DCC	Decentralized Congestion Control
DE	Data Element
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Messages
DF	Data Frame
EcuM	Electronic Control Unit Manager
ETSI	European Telecommunications Standards Institute
IF	Interface
ITS	Intelligent Transport System
ITS-S	ITS-Station
KAF	DENM Keep Alive Forwarding
LDM	Local Dynamic Map
POTI	Position and Time management
RSU	Road Side Unit
VDP	Vehicle Data Provider
VOD	Verification on Demand
V2X	Either vehicle to vehicle (V2V), or vehicle to infrastructure (V2I) and/or infrastructure to vehicle (I2V)
V2xM	Vehicle-2-X Management
V2xFac	Vehicle-2-X Facilities
V2xBtp	Vehicle-2-X Basic Transport
V2xGn	Vehicle-2-X Geo Networking
IVI	Infrastructure to Vehicle Information
IVIM	Infrastructure to Vehicle Information Message
RLT	Road and Lane Topology
MAPEM	MAP Extended Message
TLM	Traffic Light Maneuver
SPATEM	Signal Phase And Timing Extended Message

## 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] Specification of Default Error Tracer  
AUTOSAR\_SWS\_DefaultErrorTracer.pdf
- [5] Specification of ECU State Manager  
AUTOSAR\_SWS\_ECUSTateManager.pdf
- [6] Specification of V2XBasicTransport  
AUTOSAR\_SWS\_Vehicle-2-X BasicTransport.pdf
- [7] Specification of Module V2X Data Manager  
AUTOSAR\_SWS\_Vehicle-2-X Data Manager.pdf

### 3.2 Related standards and norms

- [8] IEC 7498-1 The Basic Model, IEC Norm, 1994
- [9] Intelligent Transport Systems (ITS); Communications Architecture  
ETSI EN 302 665 V1.1.1 (2010-09)
- [10] Intelligent Transport Systems (ITS); Vehicular Communications;  
Basic Set of Applications;  
Part 2: Specification of Cooperative Awareness Basic Service  
ETSI EN 302 637-2 V1.4.1 (2019-04)
- [11] Intelligent Transport Systems (ITS); Vehicular Communications;  
Basic Set of Applications;  
Part 3: Specifications of Decentralized Environmental Notification Basic  
Service  
ETSI EN 302 637-3 V1.3.1 (2019-04)
- [12] Intelligent Transport Systems (ITS); Users and applications requirements;  
Part 1: Facility layer structure, functional requirements and specifications  
ETSI TS 102 894-1 V1.1.1 (2013-08)
- [13] Intelligent Transport Systems (ITS); Users and applications requirements;  
Part 2: Applications and facilities layer common data dictionary  
ETSI TS 102 894-2 V1.3.1 (2018-08)

- [14] Intelligent Transport System (ITS); Facilities layer function;  
Part 3: Position and time facility specification"  
ETSI TS 102 890-3
- [15] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)  
ETSI EN 302 895 (V1.1.1) (2014-09)
- [16] Intelligent Transport Systems (ITS); OSI cross-layer topics;  
Part 11: Interface between networking and transport layer and facilities layer  
ETSI TS 102 723-11 V1.1.1 (2013-11)
- [17] 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)
- [18] 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)
- [19] C2C-CC BSP Requirement  
C2CCC\_RS\_2037\_BSP\_Requirements.docx
- [20] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services  
ETSI TS 103 301 V1.2.1(2018-08)
- [21] ISO/TS 19321 (2015): Intelligent transport systems -- Cooperative ITS --  
Dictionary of in-vehicle information (IVI) data structures
- [22] ISO/TS 19091 (2017): Intelligent transport systems -- Cooperative ITS -- Using V2I and I2V communications for applications related to signalized intersections
- [23] SAE J 2945/1, On-Board System Requirements for V2V Safety Communications
- [24] Intelligent Transport Systems (ITS); Vehicular Communications;  
GeoNetworking; Port Numbers for the Basic Transport Protocol (BTP)  
ETSI TS 103 248 V1.2.1

### **3.3 Related specification**

AUTOSAR provides a General Specification on Basic Software (SWS BSW General) [3] which is also valid for V2xFac.

Thus, the specification SWS BSW General [3] shall be considered as additional and required specification for V2xFac.

## **4 Constraints and assumptions**

### **4.1 Limitations**

- The V2X modules follow the guidance regarding the Day-1 scenarios defined by Car-2-Car-Consortium and C-Roads platform.
- The V2xFac module is only relevant for European Day1 V2X implementation. .

### **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 V2xFac module to other modules within the AUTOSAR basic software architecture. It outlines the modules that are required or optional for the realization of the V2xFac module and the V2xFac services that these modules use.

### 5.1 AUTOSAR DET (Default Error Tracer)

In development mode, the V2xFac module reports errors through the `Det_ReportError` function of the DET Module [4].

### 5.2 AUTOSAR EcuM (Ecu State Manager)

The EcuM [5] initializes the V2xFac module by calling `V2xFac_Init` specified in 8.3.1.

### 5.3 V2x Vehicle Data Provider

The V2xFac module retrieves vehicle relevant data from the VDP application by using the Sender-Receiver-Interface `V2xFacVdp` (see [SWS\_V2xFac\_00094]).

### 5.4 AUTOSAR V2xDM

The V2xFac module provides the Client-Server-Interface `V2xFacDenBs` for using the DEN basic service. The operations `TriggerEvent`, `UpdateEvent` or `TerminateEvent` are provided.

After having checked the validity of their header, the V2xFac module delivers the payload of the received messages to the V2xDM module. The V2xFac module assumes an interface for message reception (see [SWS\_V2xFac\_00092]) to be provided by the Vehicle-2-X Data Manager (V2xDM) module.

### 5.5 AUTOSAR V2xBtp

The V2xFac module assumes a transmit request primitive (`V2xBtp_Transmit` [6], see [SWS\_V2xFac\_00092]) to be provided by the V2xBtp module.

### 5.6 AUTOSAR V2xM

The V2xFac module assumes a request primitive (see [SWS\_V2xFac\_00092]) to be provided by the Vehicle-2-X Management (V2xM) module.



## 6 Requirements traceability

Requirement	Description	Satisfied by
SRS_BSW_00323	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	SWS_V2xFac_00106
SRS_BSW_00345	BSW Modules shall support pre-compile configuration	SWS_V2xFac_00238
SRS_V2X_00010	The implementation of the V2X system shall follow additional guidance given by C2C-CC requirements	SWS_V2xFac_00004, SWS_V2xFac_00008, SWS_V2xFac_00010, SWS_V2xFac_00014, SWS_V2xFac_00019, SWS_V2xFac_00116, SWS_V2xFac_20168, SWS_V2xFac_20185, SWS_V2xFac_20215, SWS_V2xFac_20256, SWS_V2xFac_20257, SWS_V2xFac_20297, SWS_V2xFac_20313
SRS_V2X_00190	The V2X system shall handle vehicle states in a consistent manner	SWS_V2xFac_20444, SWS_V2xFac_20445
SRS_V2X_00214	The V2X system shall allow applications to deactivate transmission of CAMs	SWS_V2xFac_00006
SRS_V2X_00232	The V2X system shall cooperate with tolling zone stations in vicinity	SWS_V2xFac_00307
SRS_V2X_00259	The V2X system shall manage the life time of all DENM packets	SWS_V2xFac_20259
SRS_V2X_00291	The V2X system shall only send messages with valid position and time	SWS_V2xFac_20215, SWS_V2xFac_20291
SRS_V2X_00301	The V2X system's Facility Layer shall handle DENM repetition	SWS_V2xFac_00029
SRS_V2X_00318	The V2X system's Facility Layer shall generate traces and path histories	SWS_V2xFac_20318
SRS_V2X_00693	The V2X system shall provide functionality for generating traces and path histories	SWS_V2xFac_20285, SWS_V2xFac_20286, SWS_V2xFac_20287, SWS_V2xFac_20288, SWS_V2xFac_20289, SWS_V2xFac_20302, SWS_V2xFac_20303, SWS_V2xFac_20304, SWS_V2xFac_20305, SWS_V2xFac_20306, SWS_V2xFac_20307, SWS_V2xFac_20308
SRS_V2X_00711	The V2X system's CA basic service shall be compliant to ETSI Specification of Cooperative Awareness Basic Service	SWS_V2xFac_00001, SWS_V2xFac_00004, SWS_V2xFac_00008, SWS_V2xFac_00009, SWS_V2xFac_00010, SWS_V2xFac_00011, SWS_V2xFac_00014, SWS_V2xFac_00015, SWS_V2xFac_00016, SWS_V2xFac_00019, SWS_V2xFac_00116, SWS_V2xFac_00231, SWS_V2xFac_00247, SWS_V2xFac_00248, SWS_V2xFac_00294, SWS_V2xFac_00295,

		SWS_V2xFac_00296, SWS_V2xFac_00301, SWS_V2xFac_00303, SWS_V2xFac_00306, SWS_V2xFac_20292, SWS_V2xFac_20297
SRS_V2X_00741	The V2X system's DEN basic service shall be compliant to ETSI Specifications of Decentralized Environmental Notification Basic Service	SWS_V2xFac_00001, SWS_V2xFac_00004, SWS_V2xFac_00025, SWS_V2xFac_00027, SWS_V2xFac_00232, SWS_V2xFac_00247, SWS_V2xFac_00248, SWS_V2xFac_00301, SWS_V2xFac_00303, SWS_V2xFac_00306
SRS_V2X_10001	The V2X system's Facility layer shall support receiving IVI messages	SWS_V2xFac_00247, SWS_V2xFac_00248, SWS_V2xFac_00301, SWS_V2xFac_00303, SWS_V2xFac_00305, SWS_V2xFac_00306
SRS_V2X_10003	The V2X system's Facility layer shall support receiving MAPEM messages	SWS_V2xFac_00247, SWS_V2xFac_00248, SWS_V2xFac_00301, SWS_V2xFac_00303, SWS_V2xFac_00306
SRS_V2X_10004	The V2X system's Facility layer shall support receiving SPAT extended messages	SWS_V2xFac_00247, SWS_V2xFac_00248, SWS_V2xFac_00301, SWS_V2xFac_00303, SWS_V2xFac_00306
SRS_V2X_26001	The V2X system shall provide selected information from a V2X message to the application layer and/or to the vehicle network.	SWS_V2xFac_00300

**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\_V2xFac\_0xxxx) are module specific and not inherited.
- SWS items starting with a leading 2 (SWS\_V2xFac\_2xxxx) are inherited from C2C-CC Basic System Profile

## 7 Functional specification

The V2xFac module operates the transmission part of the basic services Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN).

In reception, it checks the validity of the header of the received messages before providing their payload to the V2xDm module.

**[SWS\_V2xFac\_00231]** | The V2xFac module shall implement the transmission part of CA Basic Service as specified in [10] unless specified otherwise in this document | (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00232]** | The V2xFac module shall implement the transmission part of DEN Basic Service as specified in [11] unless specified otherwise in this document | (SRS\_V2X\_00741)

**[SWS\_V2xFac\_00300]** | The V2xFac module shall check the validity of the header of the received Facilities messages based on its configuration which shall define the association of Facilities message ID in the ITS header and BTP port. Incoming messages are passed to the V2xDm if configured. The V2xFac is responsible to ensure that a message is discarded if the receiving port does not match the expected message ID given via configuration. Dedicated services for certain messages (such as DenBs) might define additional validity checks for the incoming messages. The V2xFac is responsible to ensure that a message is discarded if it is categorized as invalid by its dedicated service. Additional verifications on ITS header fields such as for instance on Station ID in the scope of misbehavior detection or the consistence with supported protocol versions are also possible and left to implementors decision | (SRS\_V2X\_26001)

**[SWS\_V2xFac\_00301]** | The V2xFac module shall check the validity of the header of received CAM, DENM, IVIM, MAPEM and SPATEM as specified in [10], [11] and [20] unless specified otherwise in this document.

Facilities messages received on standardized ports as specified in

**[SWS\_V2xFac\_00247]** shall respect the one on one mapping between ports and Message IDs or be discarded. | (SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_10001, SRS\_V2X\_10003, SRS\_V2X\_10004)

**[SWS\_V2xFac\_00247]** | The V2xFac module shall only support the following messages in association to the following BTP ports:

Standardised BTP port number	Service	Message ID in ITS header	Rx/Tx
2001	CA	CAM	Rx and Tx
2002	DEN	DENM	Rx and Tx
2003	RLT	MAPEM	Rx only
2004	TLM	SPATEM	Rx only
2006	IVI	IVIM	Rx only

| (SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_10001, SRS\_V2X\_10003, SRS\_V2X\_10004)

**[SWS\_V2xFac\_20444]**

For a stationary vehicle, the system shall report the last estimated heading value. ]  
(SRS\_V2X\_00190)

**[SWS\_V2xFac\_20445]**

At system shutdown, the system shall store the last heading value and the corresponding gear position (forward, neutral or backward).  
At system start-up, the system shall report the heading value based on this stored heading value and the current gear position, until the vehicle is no longer stationary. ]  
(SRS\_V2X\_00190)

## 7.1 Startup behavior

**[SWS\_V2xFac\_00001]**

The function V2xFac\_Init (refer to chapter 8.3.2) of the V2xFac shall initialize the internal states of the V2xFac module.  
(SRS\_V2X\_00711, SRS\_V2X\_00741)

Note: The function V2xFac\_Init shall not be called before the Vehicle-2-X Management (V2xM) is initialized by the Electronic Control Unit Manager (EcuM).

**[SWS\_V2xFac\_00004]**

The function V2xFac\_Init shall initialize the basic services CA and DEN for transmission and the reception service for all message types as per configuration.]  
(SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_00010)

## 7.2 General Format Specification

**[SWS\_V2xFac\_20313]**

The data elements which constitute the content of the CAM and DENM shall be compliant to [13] ] (SRS\_V2X\_00010)

**[SWS\_V2xFac\_00248]**

The header format of received messages shall be compliant to [13]. ]  
(SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_10001, SRS\_V2X\_10003, SRS\_V2X\_10004)

## 7.3 CA Functional Specification

For details see [10] chapter 6.1.

### 7.3.1 CA Initialization, Activation and Deactivation

**[SWS\_V2xFac\_00116]**

The path history shall be cleared when the sending functionality is enabled via the V2xFac\_V2xM\_SetCaBsOperation API.] ( SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_00006] [**

CA basic service initialization shall enable the transmission of CAM messages] (SRS\_V2X\_00214)

**[SWS\_V2xFac\_00008] [**

The function V2xFac\_Init shall initialize the parameter T\_GenCam\_DCC [10] needed for the frequency management for CAMs according to T\_GenCamMax [10].

For details see [10] chapter 5.3.5 and [19]

] (SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_00009] [**

The function V2xFac\_Init shall initialize the parameter T\_GenCam [10] to the default value T\_GenCamMax.

For details see [10] chapter 6.1.3

] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00010] [**

The function V2xFac\_Init shall initialize the parameter N\_GenCam [10] to the default value 0.

] (SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_00011] [**

The function V2xFac\_Init shall initialize the parameter T\_CheckCamGen [10] to the default value equal to the configuration parameter T\_GenCamMin [10].

For details see [10] chapter 6.1.3

] (SRS\_V2X\_00711)

### 7.3.2 CAM Generation, Sending and Frequency Management

For CAM reception See 7.5.

**[SWS\_V2xFac\_00014] [**

The CA basic service shall periodically generate CAMs controlled by a CAM frequency management (For details see [10] chapter 6.1.3 and [19].)

] (SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_00015]** [

The generated CAMs shall be transmitted by the V2xBtp using the API function V2xBtp\_Transmit (see chapter 8.6.1).

] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00016]** [

The CA basic service shall receive CAMs via the callback function V2xFac\_RxIndication (see chapter 8.4).

] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00294]**

The MAX\_DANGLE [19] representing the delta angle (in degrees) between two generation rules checks shall use a value of 4°. ] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00295]**

The MAX\_DDISTANCE [19] representing the delta distance (in meters) between two generation rules checks shall use a value of 4 meters. ] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_00296]**

The MAX\_DSPEED [19] representing the delta speed between two generation rules checks shall use a value of 0,5 m/s. ] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_20297]**

The adjustable N\_GenCam parameter (see [10]) specified in the CAM Generation Frequency Management shall be set to 3 for the V2xFac module. ] (SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_20291]**

The V2xFac module shall transmit CAM messages as long as position and time information are available. ] (SRS\_V2X\_00291)

### 7.3.3 CAM Generation Frequency Management for RSU ITS-Ss

Generation of CA messages for road side units (RSU-ITS) is currently not supported by AUTOSAR.

### 7.3.4 CAM Time Requirement

**[SWS\_V2xFac\_00019]** [

The CAM generation shall follow time requirements according to [10] chapter 6.1.5 and [19].

] (SRS\_V2X\_00711, SRS\_V2X\_00010)

**[SWS\_V2xFac\_20168]** [

The V2xFac module shall check the timestamp in the security envelope compared to the reception time and accept only CAMs in the last time of 2 seconds and other messages within the last time of 10 minutes.

Due to the tolerance of the ITS station times and allowed clock deviation in [19], the V2xFac module shall accept messages coming from the future compared to ego vehicle clock:

- up to a maximum of 40 ms for vehicles (20 ms estimated deviation from ego vehicle + 20 ms deviation for transmitting vehicle).
- up to a maximum of 220 ms for RSUs (20 ms estimated deviation from ego vehicle + 200 ms deviation for transmitting RSU).

] (SRS\_V2X\_00010)

### 7.3.5 CAM Format Specification

For details about CAM data format refer to to the following ETSI documents:

See [10] chapter 7

See [10] Annex A: ASN.1 specification of CAM

See [10] Annex B: Description for data elements and data frames

See [13] Annex A, Annex B

**[SWS\_V2xFac\_20285]** [

The path history field inside the CAM low frequency (LF) container shall contain a PathHistory data element covering a distance of at least 200 m (K\_PHDISTANCE\_M parameter in [23], Appendix A.5).

An exception to the minimum covered distance by PathHistory shall be only made if either of the following conditions is fulfilled:

- the vehicle has not yet physically covered the distance with its current pseudonym (e.g., after vehicle startup or right after pseudonym change when driving)
- the maximum number of PathPoints is used while the overall length covered by the PathHistory still does not reach 200m.

Only in the above two cases the vehicle may send PathHistory information covering a distance below the 200 m lower limit.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20286]** [

The PathHistory in CAMs shall cover at most 500 m.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20287]** [

The V2xFac module shall send PathDeltaTime in every PathPoint of the PathHistory. Therefore, the PathHistory shall describe a time-ordered list (newest point first) of actually travelled geographical locations, including current ego position.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20288]** [

In cases where the vehicle does not move, i.e. PathPoint position information does not change, the PathDeltaTime of the first PathPoint shall still be updated with every CAM.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20289]** [

When the V2xFac module is stationary for a duration longer than the maximum value of PathDeltaTime (specified in [13]) the PathDeltaTime of the first PathPoint in the CAM shall be fixed to the maximum value.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20292]**

The traffic class value for CAM messages shall be set to 2. ] (SRS\_V2X\_00711)

**[SWS\_V2xFac\_20256]**

The V2xFac module shall use a Single Hop Broadcast (SHB) header on all CAM packets it sends. Therefore, the value of the transportType parameter shall be set to 0x50 ] (SRS\_V2X\_00010)

### 7.3.6 Others

**[SWS\_V2xFac\_00307]** Upon receiving a CAM message, the V2xFac module checks the station type of the emitter of this CAM message.

As an exception to the normal message handling, if the station type in the CAM message indicates an RSU, before discarding the message, the V2xFac module shall provide the position and the radius of the protected zone to the V2xM module through the V2xM\_SetTollingZoneInformation API.

If several CAM messages from the same RSU (same station Id) are received in sequence, the CAM messages shall be discarded without calling again the V2xM\_SetTollingZoneInformation API.] (SRS\_V2X\_00232)

## 7.4 DEN Functional Specification

As defined in ETSI documents (See [11] chapter 5.2) the DEN basic service is a facilities layer entity that implements the DEN protocol. It interfaces with ITS-S applications in order to receive the application request for DENM transmission and to provide the received DENM content to the ITS-S applications.

### 7.4.1 DEN Initialization

**[SWS\_V2xFac\_00025]**

The function V2xFac\_Init shall initialize an empty originating ITS-S message table. For details see [11] chapter 8.2.1.6

] (SRS\_V2X\_00741)



## 7.4.2 DENM Transmission Management

### [SWS\_V2xFac\_00027]

The DEN basic service is triggered by the V2x-Application via its service operations TriggerEvent, UpdateEvent or TerminateEvent from the service interface V2xFacDenBs (see chapter 8.7.2.1).

The function parameter “EventID” given by the above mentioned operations shall be mapped by the DEN basic service to the actionID generated for DENMs.

For details see [11] chapter 5.3 and 8.2

|(SRS\_V2X\_00741)

## 7.4.3 DENM Reception Management

See 7.5.

## 7.4.4 DENM Repetition

### [SWS\_V2xFac\_00029]

In between two consequent DENM updates, a DENM may be repeated by the DEN basic service.

For details see [11] chapter 6.1.2.3

|(SRS\_V2X\_00301)

## 7.4.5 DENM Keep Alive Forwarding (KAF)

KAF functionality for the DEN basic service as defined by ETSI is not supported.

See [11] chapter 5.3 and 8.3

## 7.4.6 DENM Format Specification

For details about DENM data format refer to to the following ETSI documents:

See [11] chapter 7,

See [11] Annex A: ASN.1 specification of DENM

See [11] Annex B: Description for data elements and data frames

See [13] Annex A, Annex B

### [SWS\_V2xFac\_20302]

The path history field inside the DEN messages shall contain Trace data elements covering a distance of at least 600 m (K\_PHDISTANCE\_M parameter in [23], Appendix A.5).

An exception to the minimum covered distance by Traces shall be only made if either of the following conditions is fulfilled:

- the vehicle has not yet physically covered the distance with its current pseudonym (e.g., after vehicle startup or right after pseudonym change when driving)
- the maximum number of PathPoints is used while the overall length covered by the PathHistory still does not reach 200m.

Only in the above two cases the vehicle may send Traces information covering a distance below the 600 m lower limit.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20303]** [

The Traces in the DENMs shall cover at most 1000 m.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20304]** [

The V2xFac module shall use the DENM traces as follow: The PathDeltaTime shall be sent in every PathPoint in the first DENM traces element. Therefore, the first element of the traces shall describe a time-ordered list (newest point first) of actually travelled geographical locations leading to the event position, including event position.] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20305]** [

The PathDeltaTime data elements of the PathPoints in the first DENM traces element shall only be updated if the DENM is updated.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20306]** [

In cases where the event detecting vehicle does not move, i.e. PathPoint position information does not change, the PathDeltaTime of the first PathPoint of the first DENM traces element shall still be updated with every DEN\_Update.

] (SRS\_V2X\_00693)

NOTE: This is only the case for stationary events where the detecting vehicle is identical to the event, e.g. a stationary vehicle warning. For dynamic events, e.g. dangerous situations, or events, where the event is not identical to the vehicle, e.g. adverse weather warning, this is not the case.

**[SWS\_V2xFac\_20307]** [

When the V2xFac module is stationary for a duration longer than the maximum value of PathDeltaTime (specified in [13]) the PathDeltaTime of the first PathPoint in the DENM shall be set to this maximum value and a new PathPoint shall be created.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20308]** [

Additional PathHistory elements may be present in the DENM traces. However, unlike the first element, these shall describe alternative routes to the event location. These routes may or may not be available at the time of detecting the event. In the alternative routes, the PathPoints shall be position-ordered (i.e. shortest-path routes) and they shall not include the PathDeltaTime.

] (SRS\_V2X\_00693)

**[SWS\_V2xFac\_20318]** [

The traces and path histories used by the V2xFac module shall be generated using the Design Method One as specified in [23], Appendix A.5.

The V2xFac module shall use the generation method with the following settings:

- $K\_PHALLOWABLEERROR\_M = 0,47\text{ m}$ , where  
 $PH\_ActualError < K\_PHALLOWABLEERROR\_M$
- Maximum distance between concise path points,  
 $K\_PH\_CHORDLENGTHTHRESHOLD = 22,5\text{ m}$
- $K\_PH\_MAXESTIMATEDRADIUS = REarthMeridian$
- $K\_PHSMALLDELTA\Phi\_R = 1\text{ degree}$
- $REarthMeridian = 6378.137\text{ km}$  (according to IUGG - International Union of Geodesy and Geophysics), used for great-circle or orthodromic distance calculation:

$$PH\_ActualChordLength = REarthMeridian * \cos^{-1}[\cos(lat1)\cos(lat1)\cos(long1 - long2) + \sin(lat1)\sin(lat2)]$$

] (SRS\_V2X\_00318)

#### [SWS\_V2xFac\_20257]

The V2xFac module shall use GeoBroadcast (GBC) headers on all DENM packets it sends. Therefore, the value of the transportType parameter shall be set to 0x40 ] (SRS\_V2X\_00010)

#### [SWS\_V2xFac\_20259]

The V2xFac module shall set the maxPacketLifetime parameter of the packets transport parameters TxParams of all GBC packets to the minimum of ValidityDuration, RepetitionInterval and itsGnMaxPacketLifetime, with ValidityDuration and RepetitionInterval values as requested by the application and itsGnMaxPacketLifetime value as specified in [18], Annex H. ] (SRS\_V2X\_00259)

## 7.5 Message Reception Functional Specification

#### [SWS\_V2xFac\_00303]

Upon receiving a message, the V2xFac module checks the validity and the consistency of the ITS header of the received messages, and, if the header is consistent and valid, makes the payload of the message available to the V2xDM module.

The payload of the received messages shall be sent to the V2xDM module via the callback function V2xDM\_V2xStackRxIndication (see chapter 8.6.1).]

(SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_10001, SRS\_V2X\_10003, SRS\_V2X\_10004)

## 7.6 Path History

#### [SWS\_V2xFac\_20185]

Facilities layer shall clear the own station's path history cache (used to fill into new messages) when the security entity changes its pseudonym identity.

] (SRS\_V2X\_00010)

**[SWS\_V2xFac\_20215]** [

Traces and path history data shall only be generated when position confidence and ITS time information are available] (SRS\_V2X\_00010,SRS\_V2X\_00291)

## 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\_V2xFac\_00106]** [

In case development error detection is enabled for the V2xFac module, the V2xFac module shall check API parameters for validity and report detected errors to the DET. ](SRS\_BSW\_00323)

**[SWS\_V2xFac\_00031]**[

<i>Type of error</i>	<i>Related error code</i>	<i>Error value</i>
API service called with wrong parameter	V2XFAC_E_PARAM	0x01
API service called with invalid pointer	V2XFAC_E_PARAM_POINTER	0x02
V2xFac initialization failed	V2XFAC_E_INIT_FAILED	0x03
API function called before the V2xFac module has been fully initialized	V2XFAC_E_UNINIT	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\_V2xFac\_00032]

<i>Module</i>	<i>Header File</i>	<i>Imported Type</i>
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_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_GnUpperProtocolType
	V2x_GeneralTypes.h	V2x_PseudonymType
	V2x_GeneralTypes.h	V2x_SecProfileType
V2x_GeneralTypes.h	V2x_TrafficClassIdType	
V2xBtp	V2xBtp.h	V2xBtp_TxParamsType

()

### 8.2 Type definitions

#### 8.2.1 V2xFac\_RxParamsType

[SWS\_V2xFac\_00034]

<b>Name</b>	V2xFac_RxParamsType
<b>Kind</b>	Structure
<b>Elements</b>	destinationPort

	<b>Type</b>	uint16
	<b>Comment</b>	Identifies the protocol entity at the ITS facilities layer at the destination of a BTP packet.
	destinationAddress	
	<b>Type</b>	V2x_GnAddressType
	<b>Comment</b>	Destination address for GeoUnicast packet
	destinationArea	
	<b>Type</b>	V2x_GnDestinationAreaType
	<b>Comment</b>	Destination area for GeoBroadcast/GeoAnycast packet.
	destinationType	
	<b>Type</b>	V2x_GnDestinationType
	<b>Comment</b>	Select which destination type (destinationAddress or destinationArea is used for this packet).
	sourcePositionVector	
	<b>Type</b>	V2x_GnLongPositionVectorType
	<b>Comment</b>	Geographical position for the source of the received GeoNetworking packet.
	securityReport	
	<b>Type</b>	V2xM_SecReportType
	<b>Comment</b>	Result information from the security operations for decryption and verification. This parameter is supplied by the V2xM module and forwarded up to the ITS Facilities layer passing through the Geo Networking and BTP layers.
	certificateId	
	<b>Type</b>	uint64
	<b>Comment</b>	Identification of source certificate, for example the certificate hash. This parameter is supplied by the V2xM and forwarded up to the ITS Facilities layer passing through the GeoNetworking and BTP layers.
	sspBits	
	<b>Type</b>	Array of uint8
	<b>Size</b>	4
	<b>Comment</b>	Sender permissions
	sspLength	
	<b>Type</b>	uint8
	<b>Comment</b>	Sender permissions length

	trafficClass	
	<b>Type</b>	V2x_TrafficClassIdType
	<b>Comment</b>	Traffic class, with which the GeoNetworking packet was generated by the source.
	remPacketLifetime	
	<b>Type</b>	uint16
	<b>Comment</b>	Remaining lifetime of the packet in [s].
	itsAid	
	<b>Type</b>	uint32
	<b>Comment</b>	The numerical value of the ITS-AID (Application Identifier).
<b>Description</b>	Wraps GeoNetworking parameters from V2xBtp	
<b>Available via</b>	V2xFac.h	

]()

## 8.3 Function definitions

### 8.3.1 V2xFac\_Init

[SWS\_V2xFac\_00082]

<b>Service Name</b>	V2xFac_Init	
<b>Syntax</b>	void V2xFac_Init ( void* CfgPtr )	
<b>Service ID [hex]</b>	0x01	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	CfgPtr	Points to a null pointer.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	Initializes the V2xFac module.	
<b>Available via</b>	V2xFac.h	



]()

### 8.3.2 V2xFac\_GetVersionInfo

[SWS\_V2xFac\_00084]

<b>Service Name</b>	V2xFac_GetVersionInfo	
<b>Syntax</b>	void V2xFac_GetVersionInfo ( Std_VersionInfoType* VersionInfoPtr )	
<b>Service ID [hex]</b>	0x02	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	None	
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	VersionInfo Ptr	Pointer to where to store the version information of this module.
<b>Return value</b>	None	
<b>Description</b>	Returns the version information of this module.	
<b>Available via</b>	V2xFac.h	

]()

[SWS\_V2xFac\_00085]

If V2xFacDevErrorDetect is enabled: If the VersionInfoPtr pointer parameter is invalid (e.g. NULL), the error-code V2XFAC\_E\_PARAM\_POINTER shall be reported to the DET module. ] ( )

### 8.3.3 V2xFac\_V2xM\_PreparePseudonymChange

[SWS\_V2xFac\_00086]

<b>Service Name</b>	V2xFac_V2xM_PreparePseudonymChange	
<b>Syntax</b>	Std_ReturnType V2xFac_V2xM_PreparePseudonymChange ( const V2x_PseudonymType* PseudonymPtr )	
<b>Service ID [hex]</b>	0x03	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	

<b>Parameters (in)</b>	PseudonymPtr	The Pseudonym provided by V2xM
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: operation successful E_NOT_OK: pseudonym change rejected
<b>Description</b>	By this API primitive the V2xFac module gets an indication that the given Pseudonym and hereby the StationId is about to be changed	
<b>Available via</b>	V2xFac_V2xM.h	

]()

#### [SWS\_V2xFac\_00136]

The function V2xFac\_V2xM\_PreparePseudonymChange shall prepare the setting of the pseudonym specific part of the StationId being used for packet transmission. ]()

#### [SWS\_V2xFac\_00137]

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

#### [SWS\_V2xFac\_00138]

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 V2XFAC\_E\_PARAM\_POINTER. ]()

### 8.3.4 V2xFac\_V2xM\_CommitPseudonymChange

#### [SWS\_V2xFac\_00140]

<b>Service Name</b>	V2xFac_V2xM_CommitPseudonymChange
<b>Syntax</b>	Std_ReturnType V2xFac_V2xM_CommitPseudonymChange ( void )
<b>Service ID [hex]</b>	0x04
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (in)</b>	None
<b>Parameters (inout)</b>	None

<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: operation successful E_NOT_OK: operation failed
<b>Description</b>	This function is called by the V2xM when all modules are OK with the pseudonym change and the change is to be committed.	
<b>Available via</b>	V2xFac_V2xM.h	

]()

#### [SWS\_V2xFac\_00141] [

The function V2xFac\_V2xM\_CommitPseudonymChange shall set the pseudonym specific part of the GeoNetworking Address being used for packet transmission and clean the path history. V2xFac shall store the access of the GeoNetworking Address for subsequent API calls. ]()

#### [SWS\_V2xFac\_00142] [

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

Note: The function requires previous preparation of the pseudonym via an API call to V2xFac\_V2xM\_PreparePseudonymChange.

### 8.3.5 V2xFac\_V2xM\_AbortPseudonymChange

#### [SWS\_V2xFac\_00144] [

<b>Service Name</b>	V2xFac_V2xM_AbortPseudonymChange	
<b>Syntax</b>	Std_ReturnType V2xFac_V2xM_AbortPseudonymChange ( void )	
<b>Service ID [hex]</b>	0x05	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	None	
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	Std_ReturnType	E_OK: operation successful E_NOT_OK: operation failed

<b>Description</b>	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.
<b>Available via</b>	V2xFac_V2xM.h

]()

**[SWS\_V2xFac\_00145]** [

The function V2xFac\_V2xM\_AbortPseudonymChange shall roll back the prepared pseudonym change. ]()

**[SWS\_V2xFac\_00146]** [

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

Note: The function requires previous preparation of the pseudonym via an API call to V2xFac\_V2xM\_PreparePseudonymChange.

### 8.3.6 V2xFac\_V2xM\_SetTGenCamDcc

**[SWS\_V2xFac\_00148]** [

<b>Service Name</b>	V2xFac_V2xM_SetTGenCamDcc	
<b>Syntax</b>	void V2xFac_V2xM_SetTGenCamDcc ( uint16 TGenCamDcc )	
<b>Service ID [hex]</b>	0x06	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	TGenCamDcc	The TGenCamDcc in [ms], provided by V2xM
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	By this API primitive the V2xFac module gets an indication of the current TGen CamDcc value.	
<b>Available via</b>	V2xFac_V2xM.h	

]()

**[SWS\_V2xFac\_00149]** [

The function V2xFac\_V2xM\_SetTGenCamDcc shall set the TGenCamDcc for subsequent API calls. ]()

**[SWS\_V2xFac\_00150]** [

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

**8.3.7 V2xFac\_V2xM\_SetCaBsOperation**

**[SWS\_V2xFac\_00152]** [

<b>Service Name</b>	V2xFac_V2xM_SetCaBsOperation	
<b>Syntax</b>	void V2xFac_V2xM_SetCaBsOperation ( boolean OperationState )	
<b>Service ID [hex]</b>	0x07	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	
<b>Parameters (in)</b>	OperationState	FALSE: CaBs disabled TRUE: CaBs enbaled
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	By this API primitive the V2xFac module gets an indication of the current operation state of the CA Basic Service.	
<b>Available via</b>	V2xFac_V2xM.h	

]()

**[SWS\_V2xFac\_00153]** [

The function V2xFac\_V2xM\_SetCaBsOperation shall enable or disable the CA Basic Service. ]()

**[SWS\_V2xFac\_00154]** [

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

**8.4 Call-back notifications**

This is a list of functions provided for other modules.

### 8.4.1 V2xFac\_TxConfirmation

#### [SWS\_V2xFac\_00087]

<b>Service Name</b>	V2xFac_TxConfirmation	
<b>Syntax</b>	void V2xFac_TxConfirmation ( uint16 TransactionId16 )	
<b>Service ID [hex]</b>	0x08	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Reentrant	
<b>Parameters (in)</b>	TransactionId16	TransactionId of the packet that has been transmitted
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	By this API primitive the V2xFac module gets a confirmation that the V2X message with a certain ID was send successfully.	
<b>Available via</b>	V2xFac.h	

()

#### [SWS\_V2xFac\_00156]

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT.>()

### 8.4.2 V2xFac\_RxIndication

#### [SWS\_V2xFac\_00088]

<b>Service Name</b>	V2xFac_RxIndication	
<b>Syntax</b>	void V2xFac_RxIndication ( uint32 TransactionId32, const V2xFac_RxParamsType* ReceiveParams, uint16 Length, const uint8* DataPtr )	
<b>Service ID [hex]</b>	0x09	
<b>Sync/Async</b>	Synchronous	
<b>Reentrancy</b>	Non Reentrant	

<b>Parameters (in)</b>	Transaction Id32	ID of the received packet. This ID is created in the V2xGn module and handed up in the protocol stack to be used for verification on demand.
	Receive Params	Wraps RxIndication parameters
	Length	Length of the data pointed by DataPtr.
	DataPtr	Payload of the received BTP packet.
<b>Parameters (inout)</b>	None	
<b>Parameters (out)</b>	None	
<b>Return value</b>	None	
<b>Description</b>	This API primitive is called by the V2xBtp module providing the data and the Geo Networking parameters of a received BTP packet to V2xFac module.	
<b>Available via</b>	V2xFac.h	

]()

#### [SWS\_V2xFac\_00158]

If development error detection is enabled: the function shall check that the service V2xFac\_Init was previously called. If the check fails, the function shall raise the development error V2XFAC\_E\_UNINIT. ]()

#### [SWS\_V2xFac\_00159]

If development error detection is enabled: the function shall check the parameter ReceiveParams for being valid. If the check fails, the function shall raise the development error V2XFAC\_E\_PARAM\_POINTER. ]()

#### [SWS\_V2xFac\_00160]

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 V2XFAC\_E\_PARAM\_POINTER. ]()

## 8.5 Scheduled functions

### 8.5.1 V2xFac\_CaBs\_MainFunction

#### [SWS\_V2xFac\_00090]

<b>Service Name</b>	V2xFac_CaBs_MainFunction
<b>Syntax</b>	void V2xFac_CaBs_MainFunction ( void )
<b>Service ID [hex]</b>	0x0a

<b>Description</b>	This is the main processing function of the CA basic service
<b>Available via</b>	SchM_V2xFac.h

]()

### 8.5.2 V2xFac\_DenBs\_MainFunction

[SWS\_V2xFac\_00091]

<b>Service Name</b>	V2xFac_DenBs_MainFunction
<b>Syntax</b>	void V2xFac_DenBs_MainFunction ( void )
<b>Service ID [hex]</b>	0x0b
<b>Description</b>	This is the main processing function of the DEN basic service.
<b>Available via</b>	SchM_V2xFac.h

]()

### 8.5.3 V2xFac\_RxS\_MainFunction

[SWS\_V2xFac\_00305]

<b>Service Name</b>	V2xFac_RxS_MainFunction
<b>Syntax</b>	void V2xFac_RxS_MainFunction ( void )
<b>Service ID [hex]</b>	0x0c
<b>Description</b>	This is the main processing function of the Facilities message reception service.
<b>Available via</b>	SchM_V2xFac.h

] (SRS\_V2X\_10001)

[SWS\_V2xFac\_00306]

The function shall process the Facilities message reception service as described in chapter 7.5. ] (SRS\_V2X\_00711, SRS\_V2X\_00741, SRS\_V2X\_10001, SRS\_V2X\_10003, SRS\_V2X\_10004)

## 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\_V2xFac\_00092]

<i>API Function</i>	<i>Header File</i>	<i>Description</i>
V2xBtp_Transmit	V2xBtp.h	This API is called by the V2xFac module to request sending a BTP-PDU to the peer BTP entity.
V2xDM_V2xStackRx-Indication	V2xDM.h	Function is called by the underlying V2x stack to provide a V2x message to the V2x Data Manager
V2xM_CalcDistance	V2xM.h	Calculates the distance between two geographical points on earth with the assumption that they are on elevation 0.
V2xM_CalcHeading-InTolerance	V2xM.h	Calculates if difference of heading values are within a tolerance value
V2xM_GetPosition-AndTime	V2xM.h	Provides the instantaneous position information.
V2xM_GetRefTimePtr	V2xM.h	Provides a pointer to the time reference of the V2X-Stack.
V2xM_SetTolling-ZoneInformation	V2xM.h	Set available tolling zone information. This is done from V2xFac that receives this information via CAM messages.

]()

## 8.6.2 Optional Interfaces

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

### [SWS\_V2xFac\_00093]

<i>API Function</i>	<i>Header File</i>	<i>Description</i>
Det_ReportError	Det.h	Service to report development errors.

]()

## 8.7 Service Interfaces

### 8.7.1 Sender-Receiver-Interfaces

#### 8.7.1.1 V2xFacVdp

##### [SWS\_V2xFac\_00094]

The V2xFac requires an interface V2xFacVdp as defined below to get data from the VDP application.

]()

**[SWS\_V2xFac\_00095]**

<b>Name</b>	V2xFacVdp	
<b>Comment</b>	Interface to receive data from VDP application	
<b>IsService</b>	false	
<b>Variation</b>	--	
<b>Data Elements</b>	vdpData	
	<b>Type</b>	V2xFac_CoopAwarenessType
	<b>Variation</b>	--

]()

## 8.7.2 Client-Server-Interfaces

### 8.7.2.1 V2xFacDenBs

The V2xFac module provides the Client-Server service Interface V2xFacDenBs to the application layer. The service Interface V2xFacDenBs shall implement the following operations.

- TriggerEvent
- UpdateEvent
- TerminateEvent

**[SWS\_V2xFac\_00098]** [

The V2X\_Facilities shall provide an interface V2xFacDenBs as defined below to provide the capability of event handling (triggering, updating and terminating DENMs).

]()

**[SWS\_V2xFac\_00099]**

<b>Name</b>	V2xFacDenBs		
<b>Comment</b>	Service of V2xFac module basic service DEN		
<b>IsService</b>	true		
<b>Variation</b>	--		
<b>Possible Errors</b>	0	E_OK	Operation successful
	1	E_NOT_OK	Operation failed
	2	E_ACTION_ID_NONEXISTENT	ActionID provided for Update/Termination does not exist
	3	E_DENM_	DENM couldn't be constructed

		UNCONSTRUCTABLE	
	4	E_DENM_TIME_OUT	DENM hasn't been sent before timeout of DENM has been reached

<b>Operation</b>	TerminateEvent		
<b>Comment</b>	Requests termination of an existing DENM ( see [11] chapter 4 and 5.4.1.4 )		
<b>Mapped to API</b>	--		
<b>Variation</b>	--		
<b>Parameters</b>	EventData		
	<b>Type</b>	V2xFac_DenMsgType	
	<b>Direction</b>	IN	
	<b>Comment</b>	Pre-filled DENM message structure, including the ActionID from Trigger Event	
	<b>Variation</b>	--	
	RepetitionDuration		
	<b>Type</b>	uint32	
	<b>Direction</b>	IN	
	<b>Comment</b>	Duration of the DENM repetition in units of milliseconds	
	<b>Variation</b>	--	
	RepetitionInterval		
	<b>Type</b>	uint16	
	<b>Direction</b>	IN	
	<b>Comment</b>	Interval of DENM repetition in units of milliseconds	
	<b>Variation</b>	--	
	DestinationArea		
	<b>Type</b>	V2xFac_GnDestinationAreaType	
	<b>Direction</b>	IN	
	<b>Comment</b>	Destination area for DENM dissemination as specified in ETSI EN 302 931.	
	<b>Variation</b>	--	
TrafficClass			
<b>Type</b>	V2xFac_TrafficClassIdType		

	<b>Direction</b>	IN
	<b>Comment</b>	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	<b>Variation</b>	--
	ActionID	
	<b>Type</b>	V2xFac_ActionIdType
	<b>Direction</b>	OUT
	<b>Comment</b>	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application
	<b>Variation</b>	--
<b>Possible Errors</b>	E_OK E_NOT_OK E_ACTION_ID_NONEXISTENT E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

<b>Operation</b>	TriggerEvent	
<b>Comment</b>	Requests creation of a new DENM ( see [11] chapter 4 and 5.4.1.2 )	
<b>Mapped to API</b>	--	
<b>Variation</b>	--	
<b>Parameters</b>	EventData	
	<b>Type</b>	V2xFac_DenMsgType
	<b>Direction</b>	IN
	<b>Comment</b>	Pre-filled DENM message structure
	<b>Variation</b>	--
	RepetitionDuration	
	<b>Type</b>	uint32
	<b>Direction</b>	IN
	<b>Comment</b>	Duration of the DENM repetition in units of milliseconds
	<b>Variation</b>	--
	RepetitionInterval	
	<b>Type</b>	uint16
	<b>Direction</b>	IN
	<b>Comment</b>	Interval of DENM repetition in units of milliseconds

	<b>Variation</b>	--
	DestinationArea	
	<b>Type</b>	V2xFac_GnDestinationAreaType
	<b>Direction</b>	IN
	<b>Comment</b>	Destination area for DENM dissemination as specified in ETSI EN 302 931.
	<b>Variation</b>	--
	TrafficClass	
	<b>Type</b>	V2xFac_TrafficClassIdType
	<b>Direction</b>	IN
	<b>Comment</b>	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	<b>Variation</b>	--
	ActionID	
	<b>Type</b>	V2xFac_ActionIdType
	<b>Direction</b>	OUT
	<b>Comment</b>	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application
<b>Variation</b>	--	
<b>Possible Errors</b>	E_OK E_NOT_OK E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

<b>Operation</b>	UpdateEvent	
<b>Comment</b>	Requests update of an existing DENM ( see [11] chapter 4 and 5.4.1.3 )	
<b>Mapped to API</b>	--	
<b>Variation</b>	--	
<b>Parameters</b>	EventData	
	<b>Type</b>	V2xFac_DenMsgType
	<b>Direction</b>	IN
	<b>Comment</b>	Pre-filled DENM message structure, including the ActionID from Trigger Event
	<b>Variation</b>	--
	RepetitionDuration	

	<b>Type</b>	uint32
	<b>Direction</b>	IN
	<b>Comment</b>	Duration of the DENM repetition in units of milliseconds
	<b>Variation</b>	--
	RepetitionInterval	
	<b>Type</b>	uint16
	<b>Direction</b>	IN
	<b>Comment</b>	Interval of DENM repetition in units of milliseconds
	<b>Variation</b>	--
	DestinationArea	
	<b>Type</b>	V2xFac_GnDestinationAreaType
	<b>Direction</b>	IN
	<b>Comment</b>	Destination area for DENM dissemination as specified in ETSI EN 302 931.
	<b>Variation</b>	--
	TrafficClass	
	<b>Type</b>	V2xFac_TrafficClassIdType
	<b>Direction</b>	IN
	<b>Comment</b>	GN traffic class of the DENM as defined in ETSI EN 302 636-4-1
	<b>Variation</b>	--
	ActionID	
<b>Type</b>	V2xFac_ActionIdType	
<b>Direction</b>	OUT	
<b>Comment</b>	The DEN basic service returns the actionID or other applicable identifier created by the DEN basic service to the requesting ITS-S application	
<b>Variation</b>	--	
<b>Possible Errors</b>	E_OK E_NOT_OK E_ACTION_ID_NONEXISTENT E_DENM_UNCONSTRUCTABLE E_DENM_TIME_OUT	

l)

### 8.7.3 Implementation Data Types

#### 8.7.3.1 V2xFac specific Implementation DataTypes

[SWS\_V2xFac\_00162]

<b>Name</b>	V2xFac_TrafficClassIdType
<b>Kind</b>	Type
<b>Derived from</b>	uint8
<b>Description</b>	Traffic class for sending DENMs
<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

[SWS\_V2xFac\_00163]

<b>Name</b>	V2xFac_GnDestinationAreaType	
<b>Kind</b>	Structure	
<b>Elements</b>	latitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Latitude [1/10 microdegree]
	longitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Longitude [1/10 microdegree]
	distanceA	
	<b>Type</b>	uint16
	<b>Comment</b>	Distance a of the geometric shape [meters]
	distanceB	
	<b>Type</b>	uint16
	<b>Comment</b>	Distance b of the geometric shape [meters]
	angle	
	<b>Type</b>	uint16
	<b>Comment</b>	Angle of the geometric shape [degrees from North]
	shape	
<b>Type</b>	V2xFac_GnAreaShapeType	
<b>Comment</b>	Shape type of the geometric area	

<b>Description</b>	Destination area for DENM dissemination as specified in ETSI EN 302 931.
<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00164]**

<b>Name</b>	V2xFac_GnAreaShapeType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_GNAREASHAPE_CIRCLE	0x00	Circle
	V2XFAC_GNAREASHAPE_RECT	0x01	Rectangle
	V2XFAC_GNAREASHAPE_ELLIPSE	0x02	Ellipsis
<b>Description</b>	Enumeration of a GeoNetworking Area Shape		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

### 8.7.3.2 Common Implementation DataTypes

**[SWS\_V2xFac\_00036]**

<b>Name</b>	V2xFac_ItsPduHeaderType		
<b>Kind</b>	Structure		
<b>Elements</b>	protocolVersion		
	<b>Type</b>	uint8	
	<b>Comment</b>	Version of ITS message and/or communication protocol	
	messageld		
	<b>Type</b>	uint8	
	<b>Comment</b>	Type of the ITS message.	
	stationId		
	<b>Type</b>	uint32	
<b>Comment</b>	Identifier of originating ITS-S		
<b>Description</b>	DF_ItsPduHeader as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.		
<b>Variation</b>	--		



<b>Available via</b>	Rte_V2xFac_Type.h
----------------------	-------------------

]()

**[SWS\_V2xFac\_00224]**

<b>Name</b>	V2xFac_DeltaReferencePositionType	
<b>Kind</b>	Structure	
<b>Elements</b>	deltaLatitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Defines offset latitude with regards to a referred latitude value.
	deltaLongitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Defines an offset longitude with regards to a referred longitude value.
	deltaAltitude	
	<b>Type</b>	sint16
	<b>Comment</b>	Defines an offset altitude with regards to a referred altitude value.
<b>Description</b>	DF_DeltaReferencePosition as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00037]**

<b>Name</b>	V2xFac_AltitudeType	
<b>Kind</b>	Structure	
<b>Elements</b>	altitudeValue	
	<b>Type</b>	sint32
	<b>Comment</b>	Altitude in a WGS84 co-ordinate system
	altitudeConfidence	
	<b>Type</b>	V2xFac_AltitudeConfidenceType
	<b>Comment</b>	Absolute accuracy of a reported altitude value
<b>Description</b>	DF_Altitude as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	

<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00165]**

<b>Name</b>	V2xFac_AltitudeConfidenceType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_ALTITUDECONFIDENCE_ALT_000_01	0x00	the altitude accuracy is equal to or less than 0.01 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_02	0x01	the altitude accuracy is equal to or less than 0.02 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_05	0x02	the altitude accuracy is equal to or less than 0.05 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_10	0x03	the altitude accuracy is equal to or less than 0.1 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_20	0x04	the altitude accuracy is equal to or less than 0.2 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_000_50	0x05	the altitude accuracy is equal to or less than 0.5 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_001_00	0x06	the altitude accuracy is equal to or less than 1 meter
	V2XFAC_ALTITUDECONFIDENCE_ALT_002_00	0x07	the altitude accuracy is equal to or less than 2 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_005_00	0x08	the altitude accuracy is equal to or less than 5 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_010_00	0x09	the altitude accuracy is equal to or less than 10 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_020_00	0x0a	the altitude accuracy is equal to or less than 20 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_050_00	0x0b	the altitude accuracy is equal to or less than 50 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_100_00	0x0c	the altitude accuracy is equal to or less than 100 meters
	V2XFAC_ALTITUDECONFIDENCE_ALT_200_00	0x0d	the altitude accuracy is equal to or less than 200 meters
V2XFAC_ALTITUDECONFIDENCE_ALT_OUTOFRANGE	0x0e	the altitude accuracy is out of range, i.e. greater than 200 meters	

	V2XFAC_ALTITUDECONFIDENCE_ALT_UNAVAILABLE	0x0f	the altitude accuracy information is unavailable
<b>Description</b>	Enumeration of DE_AltitudeConfidence as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00038]**

<b>Name</b>	V2xFac_PosConfidenceEllipseType		
<b>Kind</b>	Structure		
<b>Elements</b>	semiMajorConfidence		
	<b>Type</b>	uint16	
	<b>Comment</b>	Half of length of the major axis	
	semiMinorConfidence		
	<b>Type</b>	uint16	
	<b>Comment</b>	Half of length of the minor axis	
	semiMajorOrientation		
	<b>Type</b>	uint16	
<b>Comment</b>	Orientation direction of the ellipse major axis		
<b>Description</b>	DF_PosConfidenceEllipse as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00039]**

<b>Name</b>	V2xFac_HeadingType		
<b>Kind</b>	Structure		
<b>Elements</b>	headingValue		
	<b>Type</b>	uint16	
	<b>Comment</b>	Orientation of a heading with regards to the WGS84 north	
	headingConfidence		
	<b>Type</b>	uint8	

	<b>Comment</b>	Absolute accuracy of a reported heading value
<b>Description</b>	DF_Heading as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00040]**

<b>Name</b>	V2xFac_SpeedType	
<b>Kind</b>	Structure	
<b>Elements</b>	speedValue	
	<b>Type</b>	uint16
	<b>Comment</b>	Speed value
	speedConfidence	
	<b>Type</b>	uint8
	<b>Comment</b>	The absolute accuracy of a speed value
<b>Description</b>	DF_Speed as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00047]**

<b>Name</b>	V2xFac_ReferencePositionType	
<b>Kind</b>	Structure	
<b>Elements</b>	latitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Latitude of the geographical point
	longitude	
	<b>Type</b>	sint32
	<b>Comment</b>	Longitude of the geographical point
	posConfidenceEllipse	
	<b>Type</b>	V2xFac_PosConfidenceEllipseType

	<b>Comment</b>	Accuracy of the geographical position
	altitude	
	<b>Type</b>	V2xFac_AltitudeType
	<b>Comment</b>	Altitude and altitude accuracy of the geographical point
<b>Description</b>	DF_ReferencePosition as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00225]**

<b>Name</b>	V2xFac_ActionIdType	
<b>Kind</b>	Structure	
<b>Elements</b>	originatingStationID	
	<b>Type</b>	uint32
	<b>Comment</b>	Identifier for an ITS-S
	sequenceNumber	
	<b>Type</b>	uint16
	<b>Comment</b>	sequenceNumber
<b>Description</b>	DF_ActionID as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00059]**

<b>Name</b>	V2xFac_PathHistoryType	
<b>Kind</b>	Structure	
<b>Elements</b>	count	
	<b>Type</b>	uint8
	<b>Comment</b>	Number of valid elements within array.
	values	
	<b>Type</b>	Array of V2xFac_PathPointType

	<b>Size</b>	23
	<b>Comment</b>	--
<b>Description</b>	DF_PathHistory as defined in ETSI TS 102 894-2. Size of the Array shall be 23 as defined in ETSI EN 302 637-2.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00226]**

<b>Name</b>	V2xFac_ClosedLanesType		
<b>Kind</b>	Structure		
<b>Elements</b>	presence		
	<b>Type</b>	V2xFac_ClosedLanesPresenceType	
	<b>Comment</b>	Mark optional children present or not	
	hardShoulderStatus		
	<b>Type</b>	V2xFac_HardShoulderStatusType	
	<b>Comment</b>	Indicates the open/closing status of hard shoulder lanes	
	drivingLaneStatus		
	<b>Type</b>	V2xFac_DrivingLaneStatusType	
	<b>Comment</b>	Indicates whether a driving lane is open to traffic	
<b>Description</b>	DF_ClosedLanes as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00166]**

<b>Name</b>	V2xFac_ClosedLanesPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	hardShoulderStatus	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_ClosedLanesTypet			

<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00167]**

<b>Name</b>	V2xFac_HardShoulderStatusType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_HARDSHOULDERSTATUS_AVAILABLE_FOR_STOPPING	0x00	Hard shoulder lane available for stopping
	V2XFAC_HARDSHOULDERSTATUS_CLOSED	0x01	Hard shoulder lane closed
	V2XFAC_HARDSHOULDERSTATUS_AVAILABLE_FOR_DRIVING	0x02	Hard shoulder lane available for driving
<b>Description</b>	Enumeration of DE_HardShoulderStatus as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00168]**

<b>Name</b>	V2xFac_DrivingLaneStatusType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint16			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	outermostLaneClosed	0x2000	Bit 13: Outermost lane is closed
	bit	secondLaneFromOutside Closed	0x1000	Bit 12: Second lane from the outside is closed
	bit	thirdLaneFromOutside Closed	0x800	Bit 11: Third lane from the outside is closed
	bit	fourthLaneFromOutside Closed	0x400	Bit 10: Fourth lane from the outside is closed
	bit	fifthLaneFromOutside Closed	0x200	Bit 9: Fifth lane from the outside is closed
	bit	sixthLaneFromOutside Closed	0x100	Bit 8: Sixth lane from the outside is closed
bit	seventhLaneFromOutside	0x80	Bit 7: Seventh lane from the outside is	

		Closed		closed
	bit	eighthLaneFromOutside Closed	0x40	Bit 6: Eighth lane from the outside is closed
	bit	ninthLaneFromOutside Closed	0x20	Bit 5: Ninth lane from the outside is closed
	bit	tenthLaneFromOutside Closed	0x10	Bit 4: Tenth lane from the outside is closed
	bit	eleventhLaneFromOutside Closed	0x08	Bit 3: Eleventh lane from the outside is closed
	bit	twelfthLaneFromOutside Closed	0x04	Bit 2: Twelfth lane from the outside is closed
	bit	thirteenthLaneFrom OutsideClosed	0x02	Bit 1: Thirteenth lane from the outside is closed
	bit	fourteenthLaneFrom OutsideClosed	0x01	Bit 0 (LSB): Fourteenth lane from the outside is closed
<b>Description</b>	BitString DE_DrivingLaneStatus as defined in ETSI TS 102 894-2 V1.2.1.			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00074]**

<b>Name</b>	V2xFac_CauseCodeType		
<b>Kind</b>	Structure		
<b>Elements</b>	causeCode		
	<b>Type</b>	uint8	
	<b>Comment</b>	Encoded value of a traffic event type	
	subCauseCode		
	<b>Type</b>	uint8	
	<b>Comment</b>	Type of sub cause of a detected event	
<b>Description</b>	DF_CauseCode as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()



### 8.7.3.3 DENM specific Implementation DataTypes

[SWS\_V2xFac\_00304] [

Dangerous Goods and Road works containers are not supported by V2xFac module  
 ()

[SWS\_V2xFac\_00069][

<b>Name</b>	V2xFac_DenmMessageRootType	
<b>Kind</b>	Structure	
<b>Elements</b>	itsPduHeader	
	<b>Type</b>	V2xFac_ItsPduHeaderType
	<b>Comment</b>	Structure of the ItsPduHeader
	denm	
	<b>Type</b>	V2xFac_DenMsgType
	<b>Comment</b>	Structure of the DEN data
	transactionId	
	<b>Type</b>	uint32
<b>Comment</b>	TransactionId for received DENM	
<b>Description</b>	DENM root message as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

[SWS\_V2xFac\_00070][

<b>Name</b>	V2xFac_DenMsgType	
<b>Kind</b>	Structure	
<b>Elements</b>	presence	
	<b>Type</b>	V2xFac_DenMsgPresenceType
	<b>Comment</b>	Mark optional childs present or not
	management	
	<b>Type</b>	V2xFac_ManagementContainerType
	<b>Comment</b>	management container
	situation	
	<b>Type</b>	V2xFac_SituationContainerType

	<b>Comment</b>	situation container
		location
	<b>Type</b>	V2xFac_LocationContainerType
	<b>Comment</b>	location container
		alacarte
	<b>Type</b>	V2xFac_AlacarteContainerType
	<b>Comment</b>	alacarte container
<b>Description</b>	DecentralizedEnvironmentalNotificationMessage as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00199]**

<b>Name</b>	V2xFac_DenMsgPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	situation	0x04	Bit 2: Optional child present
	bit	location	0x02	Bit 1: Optional child present
	bit	alacarte	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_DenMsgType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00071]**

<b>Name</b>	V2xFac_ManagementContainerType		
<b>Kind</b>	Structure		
<b>Elements</b>		presence	
	<b>Type</b>	V2xFac_ManagementContainerPresenceType	
	<b>Comment</b>	Mark optional childs present or not	
		actionId	

	<b>Type</b>	V2xFac_ActionIdType
	<b>Comment</b>	Action identifier
	detectionTime	
	<b>Type</b>	uint64
	<b>Comment</b>	Time at which the event is detected
	referenceTime	
	<b>Type</b>	uint64
	<b>Comment</b>	Refers to the time at which a new DENM, an update DENM or a cancellation DENM is generated
	termination	
	<b>Type</b>	V2xFac_TerminationType
	<b>Comment</b>	Indicates if the type of generated DENM is a cancellation DENM or a negation DENM.
	eventPosition	
	<b>Type</b>	V2xFac_ReferencePositionType
	<b>Comment</b>	Geographical position of the detected event
	relevanceDistance	
	<b>Type</b>	V2xFac_RelevanceDistanceType
	<b>Comment</b>	The distance in which event information is relevant for the receiving ITS-S
	relevanceTrafficDirection	
	<b>Type</b>	V2xFac_RelevanceTrafficDirectionType
	<b>Comment</b>	Traffic direction that is relevant to information indicated in a message
	validityDuration	
	<b>Type</b>	uint32
	<b>Comment</b>	estimation of how long the event may persist
	transmissionInterval	
	<b>Type</b>	uint16
	<b>Comment</b>	Time interval between two consecutive message transmissions
	stationType	
	<b>Type</b>	uint8
	<b>Comment</b>	Station type information of the originating ITS-S

<b>Description</b>	ManagementContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.
<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00240]**

<b>Name</b>	V2xFac_TerminationType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_TERMINATION_ISCANCELLATION	0x00	Cancellation
	V2XFAC_TERMINATION_ISNEGATION	0x01	--
<b>Description</b>	Enumeration of Termination as defined in ETSI EN 302 637-3.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00200]**

<b>Name</b>	V2xFac_RelevanceDistanceType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_50_M	0x00	less than 50 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_100_M	0x01	less than 100 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_200_M	0x02	less than 200 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_500_M	0x03	less than 500 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_1000_M	0x04	less than 1000 m
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_5_KM	0x05	less than 5 km
	V2XFAC_RELEVANCEDISTANCE_LESS_THAN_10_KM	0x06	less than 10 km
	V2XFAC_RELEVANCEDISTANCE_OVER_10_KM	0x07	over 10 km
<b>Description</b>	Enumeration of DE_RelevanceDistance as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

[SWS\_V2xFac\_00201]

<b>Name</b>	V2xFac_RelevanceTrafficDirectionType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_RELEVANCETRAFFICDIRECTION_ALL_TRAFFIC_DIRECTIONS	0x00	all traffic directions
	V2XFAC_RELEVANCETRAFFICDIRECTION_UPSTREAM_TRAFFIC	0x01	upstream traffic
	V2XFAC_RELEVANCETRAFFICDIRECTION_DOWNSTREAM_TRAFFIC	0x02	downstream traffic
	V2XFAC_RELEVANCETRAFFICDIRECTION_OPPOSITE_TRAFFIC	0x03	opposite traffic
<b>Description</b>	Enumeration of DE_RelevanceTrafficDirection as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

[SWS\_V2xFac\_00202]

<b>Name</b>	V2xFac_ManagementContainerPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	termination	0x08	Bit 3: Optional child present
	bit	relevanceDistance	0x04	Bit 2: Optional child present
	bit	relevanceTrafficDirection	0x02	Bit 1: Optional child present
	bit	transmissionInterval	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_ManagementContainerType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

[SWS\_V2xFac\_00073]

<b>Name</b>	V2xFac_SituationContainerType
-------------	-------------------------------

<b>Kind</b>	Structure		
<b>Elements</b>	presence		
	<b>Type</b>	V2xFac_SituationContainerPresenceType	
	<b>Comment</b>	Mark optional childs present or not	
	informationQuality		
	<b>Type</b>	uint8	
	<b>Comment</b>	Quality level of the information provided by the ITS-S application	
	eventType		
	<b>Type</b>	V2xFac_CauseCodeType	
	<b>Comment</b>	Encoded value of a traffic event type	
	linkedCause		
	<b>Type</b>	V2xFac_CauseCodeType	
	<b>Comment</b>	Encoded value of a traffic event type	
	eventHistory		
	<b>Type</b>	V2xFac_EventHistoryType	
<b>Comment</b>	EventHistory		
<b>Description</b>	SituationContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

[SWS\_V2xFac\_00203]

<b>Name</b>	V2xFac_SituationContainerPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	linkedCause	0x02	Bit 1: Optional child present
	bit	eventHistory	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_SituationContainerType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

[SWS\_V2xFac\_00075]

<b>Name</b>	V2xFac_EventHistoryType	
<b>Kind</b>	Structure	
<b>Elements</b>	count	
	<b>Type</b>	uint8
	<b>Comment</b>	Number of valid elements within array.
	values	
	<b>Type</b>	Array of V2xFac_EventPointType
	<b>Size</b>	23
	<b>Comment</b>	--
<b>Description</b>	DF_EventHistory as defined in ETSI TS 102 894-2.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

[SWS\_V2xFac\_00076]

<b>Name</b>	V2xFac_EventPointType	
<b>Kind</b>	Structure	
<b>Elements</b>	presence	
	<b>Type</b>	V2xFac_EventPointPresenceType
	<b>Comment</b>	Mark optional childs present or not
	eventPosition	
	<b>Type</b>	V2xFac_DeltaReferencePositionType
	<b>Comment</b>	Offset position of a detected event point.
	eventDeltaTime	
	<b>Type</b>	uint16
	<b>Comment</b>	Time travelled by the detecting ITS-S since the previous detected event point.
	informationQuality	
	<b>Type</b>	uint8
	<b>Comment</b>	Information quality of the detection for this event point.
<b>Description</b>	DF_EventPoint as defined in ETSI TS 102 894-2. Values for data elements within this	

	structure shall be used according that document.
<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00204]**

<b>Name</b>	V2xFac_EventPointPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	eventDeltaTime	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_EventPointType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00077]**

<b>Name</b>	V2xFac_LocationContainerType		
<b>Kind</b>	Structure		
<b>Elements</b>	presence		
	<b>Type</b>	V2xFac_LocationContainerPresenceType	
	<b>Comment</b>	Mark optional childs present or not	
	eventSpeed		
	<b>Type</b>	V2xFac_SpeedType	
	<b>Comment</b>	Moving speed of a detected event	
	eventPositionHeading		
	<b>Type</b>	V2xFac_HeadingType	
	<b>Comment</b>	The heading direction of the event	
	traces		
	<b>Type</b>	V2xFac_TracesType	
	<b>Comment</b>	One or more paths	
	roadType		
	<b>Type</b>	V2xFac_RoadTypeType	



	<b>Comment</b>	Type of a road segment.
<b>Description</b>	LocationContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00241]**

<b>Name</b>	V2xFac_RoadTypeType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_ROADTYPE_URBAN_NOSTRUCTURALSEPARATIONTOOPPOSITELANES	0x00	Urban road without structural separation to opposite lanes.
	V2XFAC_ROADTYPE_URBAN_WITHSTRUCTURALSEPARATIONTOOPPOSITELANES	0x01	Urban road with structural separation to opposite lanes.
	V2XFAC_ROADTYPE_NONURBAN_NOSTRUCTURALSEPARATIONTOOPPOSITELANES	0x02	Non-urban road without structural separation to opposite lanes.
	V2XFAC_ROADTYPE_ONURBAN_WITHSTRUCTURALSEPARATIONTOOPPOSITELANES	0x03	Non-urban road with structural separation to opposite lanes.
<b>Description</b>	Enumeration of DE_RoadType as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00205]**

<b>Name</b>	V2xFac_TracesType	
<b>Kind</b>	Structure	
<b>Elements</b>	count	
	<b>Type</b>	uint8

	<b>Comment</b>	Number of valid elements within array.
		values
	<b>Type</b>	Array of V2xFac_PathHistoryType
	<b>Size</b>	7
	<b>Comment</b>	--
<b>Description</b>	DF_Traces as defined in ETSI TS 102 894-2. Size of the Array shall be 7.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00206]**

<b>Name</b>	V2xFac_LocationContainerPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	eventSpeed	0x04	Bit 2: Optional child present
	bit	eventPositionHeading	0x02	Bit 1: Optional child present
	bit	roadType	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_LocationContainerType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00078]**

<b>Name</b>	V2xFac_AlacarteContainerType		
<b>Kind</b>	Structure		
<b>Elements</b>	presence		
	<b>Type</b>	V2xFac_AlacarteContainerPresenceType	
	<b>Comment</b>	Mark optional childs present or not	
	lanePosition		
	<b>Type</b>	sint8	
	<b>Comment</b>	The lane position of the event position	
	impactReduction		

	<b>Type</b>	V2xFac_ImpactReductionContainerType
	<b>Comment</b>	--
	externalTemperature	
	<b>Type</b>	sint8
	<b>Comment</b>	Indicates the ambient temperature at the event position
	positioningSolution	
	<b>Type</b>	V2xFac_PositioningSolutionTypeType
	<b>Comment</b>	Indicates the positioning technology being used to estimate a geographical position
	stationaryVehicle	
	<b>Type</b>	V2xFac_StationaryVehicleContainerType
<b>Comment</b>	--	
<b>Description</b>	AlacarteContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

[SWS\_V2xFac\_00207]

<b>Name</b>	V2xFac_PositioningSolutionTypeType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_POSITIONINGSOLUTIONTYPE_NO_POSITIONING_SOLUTION	0x00	No GNSS
	V2XFAC_POSITIONINGSOLUTIONTYPE_SGNSS	0x01	Global Navigation Satellite System
	V2XFAC_POSITIONINGSOLUTIONTYPE_DGNSS	0x02	Differential GNSS
	V2XFAC_POSITIONINGSOLUTIONTYPE_SGNSSPLUSDR	0x03	GNSS and dead reckoning
	V2XFAC_POSITIONINGSOLUTIONTYPE_DGNSSPLUSDR	0x04	Differential GNSS and dead reckoning
	V2XFAC_POSITIONINGSOLUTIONTYPE_DR	0x05	dead reckoning
<b>Description</b>	Enumeration of DE_PositioningSolutionType as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		

<b>Available via</b>	Rte_V2xFac_Type.h
----------------------	-------------------

]()

**[SWS\_V2xFac\_00208]**

<b>Name</b>	V2xFac_AlacarteContainerPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	lanePosition	0x20	Bit 5: Optional child present
	bit	impactReduction	0x10	Bit 4: Optional child present
	bit	externalTemperature	0x08	Bit 3: Optional child present
	bit	roadWorks	0x04	Bit 2: Optional child present
	bit	positioningSolution	0x02	Bit 1: Optional child present
	bit	stationaryVehicle	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_AlacarteContainerType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00079]**

<b>Name</b>	V2xFac_ImpactReductionContainerType		
<b>Kind</b>	Structure		
<b>Elements</b>	heightLonCarrLeft		
	<b>Type</b>	uint8	
	<b>Comment</b>	Height of left longitudinal carrier of the vehicle from base to top	
	heightLonCarrRight		
	<b>Type</b>	uint8	
	<b>Comment</b>	Height of right longitudinal carrier of the vehicle from base to top	
	posLonCarrLeft		
	<b>Type</b>	uint8	
	<b>Comment</b>	Distance from the centre of vehicle front bumper to the front of the left longitudinal carrier of vehicle	
posLonCarrRight			

	<b>Type</b>	uint8
	<b>Comment</b>	Distance from the centre of vehicle front bumper to the front of the right longitudinal carrier of vehicle
	positionOfPillars	
	<b>Type</b>	V2xFac_PositionOfPillarsType
	<b>Comment</b>	Indicates the perpendicular inter-distance of neighbouring pillar
	posCentMass	
	<b>Type</b>	uint8
	<b>Comment</b>	Indicates the perpendicular distance from the centre of mass of an empty load vehicle
	wheelBaseVehicle	
	<b>Type</b>	uint8
	<b>Comment</b>	Perpendicular distance between front and rear axle of the wheel base of vehicle
	turningRadius	
	<b>Type</b>	uint8
	<b>Comment</b>	The smallest circular turn (i.e. U-turn) that the vehicle is capable of making
	posFrontAx	
	<b>Type</b>	uint8
	<b>Comment</b>	Perpendicular distance between the vehicle front line of the bounding box and the front wheel axle in 10 centimetres
	positionOfOccupants	
	<b>Type</b>	V2xFac_PositionOfOccupantsType
	<b>Comment</b>	indicates whether a in vehicle seat is occupied at the moment when the impactReduction is generated
	vehicleMass	
	<b>Type</b>	uint16
	<b>Comment</b>	Mass of an empty loaded vehicle in multiple of 100 kg
	requestResponseIndication	
	<b>Type</b>	V2xFac_RequestResponseIndicationType
	<b>Comment</b>	This DE includes whether an ITS message is transmitted as request from ITS-S or a response transmitted from ITS-S after receiving request from other ITS-Ss
<b>Description</b>	ImpactReductionContainer as defined in ETSI EN 302 637-3. Values for data	

	elements within this structure shall be used according that document.
<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00209]**

<b>Name</b>	V2xFac_PositionOfPillarsType		
<b>Kind</b>	Structure		
<b>Elements</b>	count		
	<b>Type</b>	uint8	
	<b>Comment</b>	Number of valid elements within array.	
	values		
	<b>Type</b>	Array of uint8	
	<b>Size</b>	3	
	<b>Comment</b>	--	
<b>Description</b>	DF_PositionOfPillars as defined in ETSI TS 102 894-2. Size of the Array shall be 3.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00210]**

<b>Name</b>	V2xFac_PositionOfOccupantsType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint32			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	row1LeftOccupied	0x80000	Bit 19: row 1 left occupied
	bit	row1RightOccupied	0x40000	Bit 18: row 1 right occupied
	bit	row1MidOccupied	0x20000	Bit 17: row 1 mid occupied
	bit	row1NotDetectable	0x10000	Bit 16: row 1 not detectable
	bit	row1NotPresent	0x8000	Bit 15: row 1 not present
	bit	row2LeftOccupied	0x4000	Bit 14: row 2 left occupied
	bit	row2RightOccupied	0x2000	Bit 13: row 2 right occupied
	bit	row2MidOccupied	0x1000	Bit 12: row 2 mid occupied

	bit	row2NotDetectable	0x800	Bit 11: row 2 not detectable
	bit	row2NotPresent	0x400	Bit 10: row 2 not present
	bit	row3LeftOccupied	0x200	Bit 9: row 3 left occupied
	bit	row3RightOccupied	0x100	Bit 8: row 3 right occupied
	bit	row3MidOccupied	0x80	Bit 7: row 3 mid occupied
	bit	row3NotDetectable	0x40	Bit 6: row 3 not detectable
	bit	row3NotPresent	0x20	Bit 5: row 3 not present
	bit	row4LeftOccupied	0x10	Bit 4: row 4 left occupied
	bit	row4RightOccupied	0x08	Bit 3: row 4 right occupied
	bit	row4MidOccupied	0x04	Bit 2: row 4 mid occupied
	bit	row4NotDetectable	0x02	Bit 1: row 4 not detectable
	bit	row4NotPresent	0x01	Bit 0 (LSB): row 4 not present
<b>Description</b>	BitString DE_PositionOfOccupants as defined in ETSI TS 102 894-2.			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00242]**

<b>Name</b>	V2xFac_RequestResponseIndicationType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_REQUESTRESPONSEINDICATION_REQUEST	0x00	Request
	V2XFAC_REQUESTRESPONSEINDICATION_RESPONSE	0x01	Response
<b>Description</b>	Enumeration of DE_RequestResponseIndication as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

**[SWS\_V2xFac\_00081]**

<b>Name</b>	V2xFac_StationaryVehicleContainerType
<b>Kind</b>	Structure
<b>Elements</b>	presence

	<b>Type</b>	V2xFac_StationaryVehicleContainerPresenceType
	<b>Comment</b>	Mark optional childs present or not
	stationarySince	
	<b>Type</b>	V2xFac_StationarySinceType
	<b>Comment</b>	Duration in minutes of a vehicle being stationary
	stationaryCause	
	<b>Type</b>	V2xFac_CauseCodeType
	<b>Comment</b>	Additional information to describe causes of the stationary vehicle
	numberOfOccupants	
	<b>Type</b>	uint8
	<b>Comment</b>	Number of occupants in a vehicle
	vehicleIdentification	
	<b>Type</b>	V2xFac_VehicleIdentificationType
	<b>Comment</b>	Provides information related to the identification of a vehicle
	energyStorageType	
<b>Type</b>	V2xFac_EnergyStorageType	
<b>Comment</b>	Type of energy being used and stored	
<b>Description</b>	StationaryVehicleContainer as defined in ETSI EN 302 637-3. Values for data elements within this structure shall be used according that document.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

[SWS\_V2xFac\_00216]

<b>Name</b>	V2xFac_StationarySinceType		
<b>Kind</b>	Type		
<b>Derived from</b>	uint8		
<b>Range</b>	V2XFAC_STATIONARYSINCE_LESS_THAN_1_MINUTE	0x00	less than 1 minute
	V2XFAC_STATIONARYSINCE_LESS_THAN_2_MINUTES	0x01	less than 2 minutes
	V2XFAC_STATIONARYSINCE_LESS_THAN_15_MINUTES	0x02	less than 15 minutes



	V2XFAC_STATIONARYSINCE_EQUAL_OR_GREATER_15_MINUTES	0x03	equal or greater 15 minutes
<b>Description</b>	Enumeration of DE_StationarySince as defined in ETSI TS 102 894-2.		
<b>Variation</b>	--		
<b>Available via</b>	Rte_V2xFac_Type.h		

]()

[SWS\_V2xFac\_00217]

<b>Name</b>	V2xFac_EnergyStorageType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	hydrogenStorage	0x40	Bit 6: hydrogen storage
	bit	electricEnergyStorage	0x20	Bit 5: electric energy storage
	bit	liquidPropaneGas	0x10	Bit 4: liquid propane gas
	bit	compressedNaturalGas	0x08	Bit 3: compressed natural gas
	bit	diesel	0x04	Bit 2: diesel
	bit	gasoline	0x02	Bit 1: gasoline
	bit	ammonia	0x01	Bit 0 (LSB): ammonia
<b>Description</b>	BitString DE_EnergyStorage as defined in ETSI TS 102 894-2.			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

[SWS\_V2xFac\_00218]

<b>Name</b>	V2xFac_StationaryVehicleContainerPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	stationarySince	0x20	Bit 5: Optional child present
	bit	stationaryCause	0x10	Bit 4: Optional child present
	bit	carryingDangerousGoods	0x08	Bit 3: Optional child present
	bit	numberOfOccupants	0x04	Bit 2: Optional child present

	bit	vehicleIdentification	0x02	Bit 1: Optional child present
	bit	energyStorageType	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_StationaryVehicleContainerType			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00230]**

<b>Name</b>	V2xFac_VehicleIdentificationType			
<b>Kind</b>	Structure			
<b>Elements</b>	presence			
	<b>Type</b>	V2xFac_VehicleIdentificationPresenceType		
	<b>Comment</b>	Mark optional childs present or not		
	wmiNumber			
	<b>Type</b>	V2xFac_WmiNumberType		
	<b>Comment</b>	World Manufacturer Identifier (WMI)		
	vds			
	<b>Type</b>	V2xFac_VdsType		
<b>Comment</b>	Vehicle Descriptor Section (VDS)			
<b>Description</b>	DF_VehicleIdentification as defined in ETSI TS 102 894-2. Values for data elements within this structure shall be used according that document.			
<b>Variation</b>	--			
<b>Available via</b>	Rte_V2xFac_Type.h			

]()

**[SWS\_V2xFac\_00223]**

<b>Name</b>	V2xFac_VehicleIdentificationPresenceType			
<b>Kind</b>	Bitfield			
<b>Derived from</b>	uint8			
<b>Elements</b>	<b>Kind</b>	<b>Name</b>	<b>Mask</b>	<b>Description</b>
	bit	wmiNumber	0x02	Bit 1: Optional child present
	bit	vds	0x01	Bit 0 (LSB): Optional child present
<b>Description</b>	Presence flags for V2xFac_VehicleIdentificationType			

<b>Variation</b>	--
<b>Available via</b>	Rte_V2xFac_Type.h

]()

**[SWS\_V2xFac\_00243]**

<b>Name</b>	V2xFac_WmiNumberType	
<b>Kind</b>	Structure	
<b>Elements</b>	count	
	<b>Type</b>	uint8
	<b>Comment</b>	Number of valid elements within array.
	values	
	<b>Type</b>	Array of uint8
	<b>Size</b>	3
	<b>Comment</b>	--
<b>Description</b>	DE_WMInumber as defined in ETSI TS 102 894-2. Size of the Array shall be 3.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

**[SWS\_V2xFac\_00244]**

<b>Name</b>	V2xFac_VdsType	
<b>Kind</b>	Structure	
<b>Elements</b>	count	
	<b>Type</b>	uint8
	<b>Comment</b>	Number of valid elements within array.
	values	
	<b>Type</b>	Array of uint8
	<b>Size</b>	6
	<b>Comment</b>	--
<b>Description</b>	DE_VDS as defined in ETSI TS 102 894-2. Size of the Array shall be 6.	
<b>Variation</b>	--	
<b>Available via</b>	Rte_V2xFac_Type.h	

]()

## 9 Sequence diagrams

### 9.1 CAM Generation and Transmission

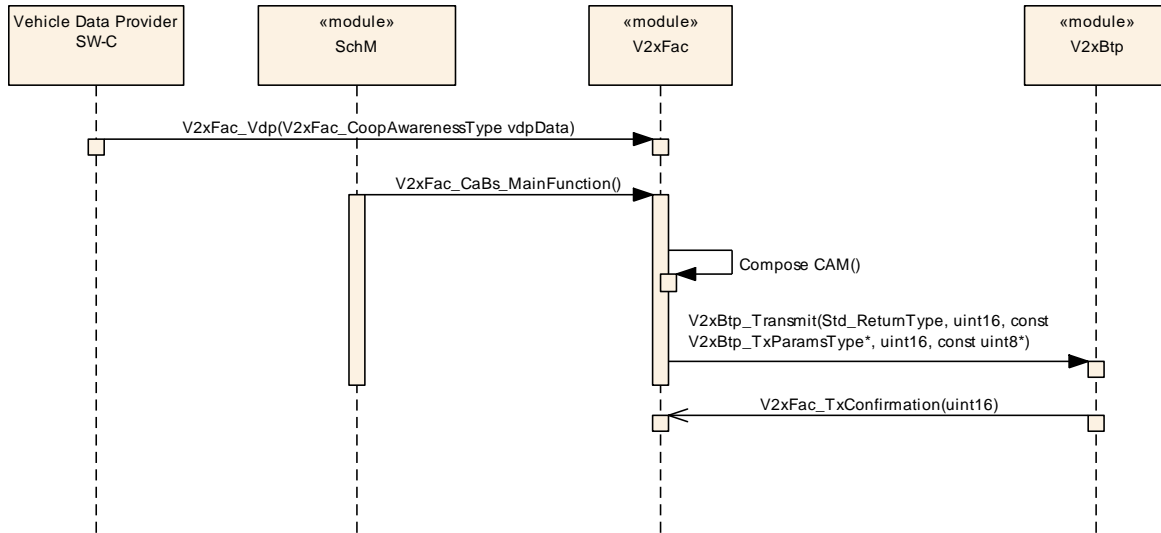


Figure 9.1 CAM Generation and Transmission

## 9.2 DENM Generation and Transmission

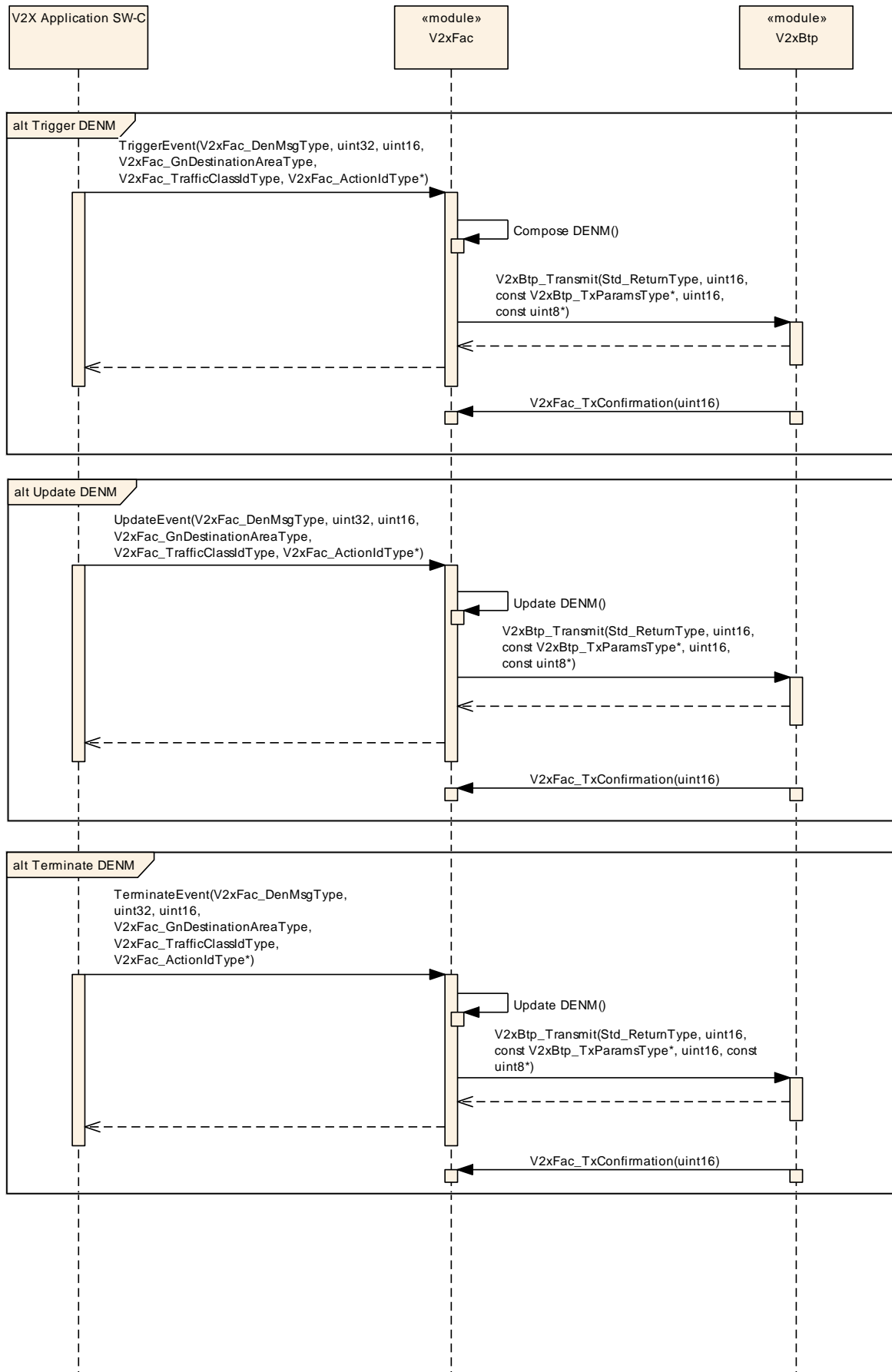


Figure 9.2 DENM Generation and Transmission

### 9.3 Message Reception

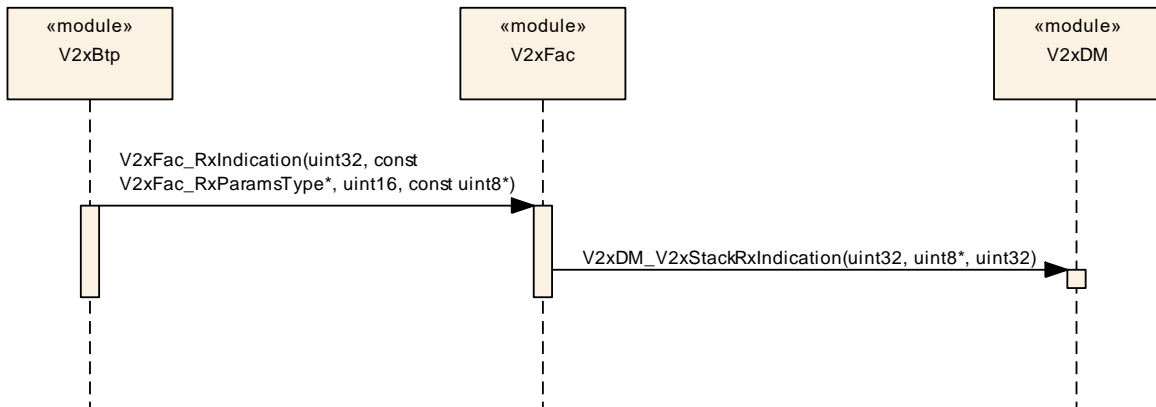


Figure 9.3 Message Reception

## 10 Configuration specification

Chapter 10.1 specifies the structure (containers) and the parameters of the module V2xFac.

Chapter 10.2 specifies additionally published information of the module V2xFac.

### 10.1 Containers and configuration parameters

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

#### 10.1.1 Variants

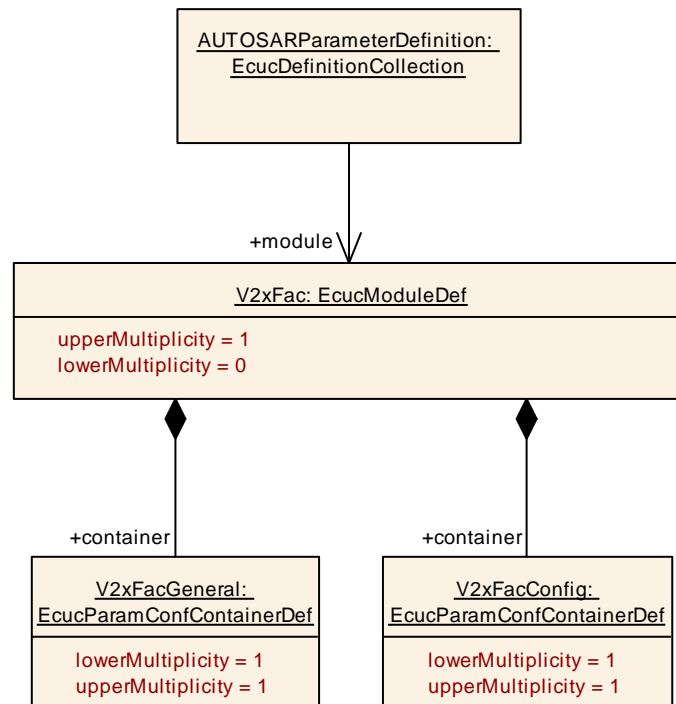
**[SWS\_V2xFac\_00238]** [ The V2xFac module only supports VARIANT-PRE-COMPILE ] (SRS\_BSW\_00345)

#### 10.1.2 V2xFac

<b>SWS Item</b>	[ECUC_V2xFac_00001]
<b>Module Name</b>	V2xFac
<b>Description</b>	Configuration of the V2xFac module.
<b>Post-Build Variant Support</b>	false
<b>Supported Config Variants</b>	VARIANT-PRE-COMPILE

Included Containers		
Container Name	Multiplicity	Scope / Dependency
V2xFacConfig	1	This container contains the configuration parameters and sub containers of the AUTOSAR V2xFac module.
V2xFac-General	1	This container contains the general configuration parameters of the Vehicle-2-X Facilities





### 10.1.3 V2xFacGeneral

<b>SWS Item</b>	[ECUC_V2xFac_00002]
<b>Container Name</b>	V2xFacGeneral
<b>Parent Container</b>	V2xFac
<b>Description</b>	This container contains the general configuration parameters of the Vehicle-2-X Facilities
<b>Configuration Parameters</b>	

<b>SWS Item</b>	[ECUC_V2xFac_00006]
<b>Parameter Name</b>	V2xFacCaBsMainFunctionPeriod
<b>Parent Container</b>	V2xFacGeneral
<b>Description</b>	This parameter defines the schedule period of V2xFac_CaBs_Main Function. Unit: [s]
<b>Multiplicity</b>	1
<b>Type</b>	EcucFloatParamDef
<b>Range</b>	]0 .. INF[
<b>Default value</b>	0.1

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

<b>SWS Item</b>	[ECUC_V2xFac_00005]		
<b>Parameter Name</b>	V2xFacDenBsMainFunctionPeriod		
<b>Parent Container</b>	V2xFacGeneral		
<b>Description</b>	This parameter defines the schedule period of V2xFac_DenBs_Main Function.Unit: [s]		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	0.1		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_V2xFac_00004]		
<b>Parameter Name</b>	V2xFacDevErrorDetect		
<b>Parent Container</b>	V2xFacGeneral		
<b>Description</b>	Switches the Default Error Tracer (Det) detection and notification ON or OFF. <ul style="list-style-type: none"> <li>• true: enabled (ON)</li> <li>• false: disabled (OFF)</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		

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

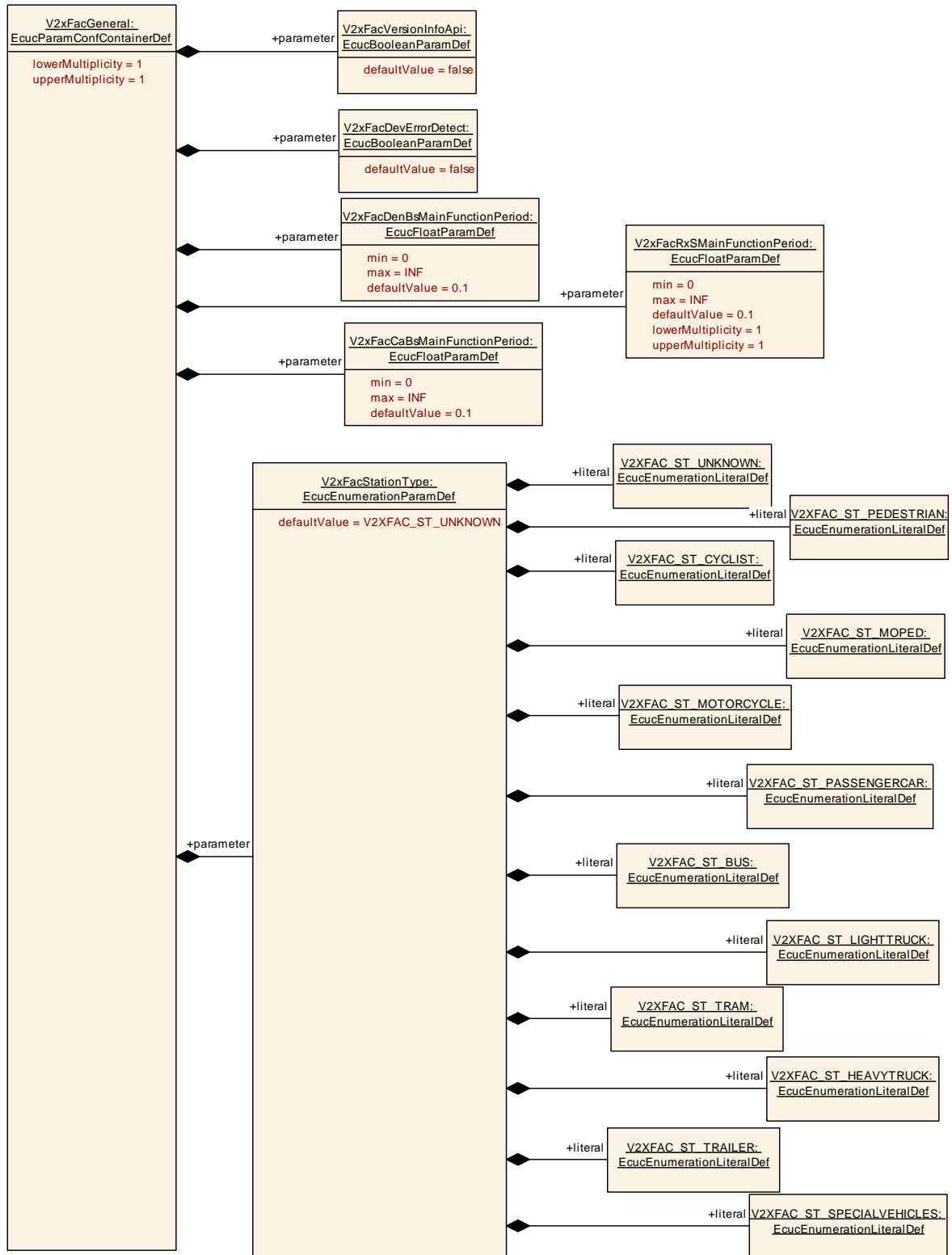
<b>SWS Item</b>	[ECUC_V2xFac_00015]		
<b>Parameter Name</b>	V2xFacRxSMainFunctionPeriod		
<b>Parent Container</b>	V2xFacGeneral		
<b>Description</b>	This parameter defines the schedule period of V2xFac_RxS_Main Function.Unit: [s]		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucFloatParamDef		
<b>Range</b>	]0 .. INF[		
<b>Default value</b>	0.1		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_V2xFac_00007]		
<b>Parameter Name</b>	V2xFacStationType		
<b>Parent Container</b>	V2xFacGeneral		
<b>Description</b>	This configuration value defines the station type information of the originating ITS-S, RoadSideUnit (15) not supported by AUTOSAR.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucEnumerationParamDef		
<b>Range</b>	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	--	
<b>Default value</b>	V2XFAC_ST_UNKNOWN		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_V2xFac_00003]		
<b>Parameter Name</b>	V2xFacVersionInfoApi		
<b>Parent Container</b>	V2xFacGeneral		
<b>Description</b>	Enable/disables the API for reading the version information of the V2x Fac Module. <ul style="list-style-type: none"> <li>• true: enabled (ON)</li> <li>• false: disabled (OFF)</li> </ul>		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucBooleanParamDef		
<b>Default value</b>	false		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

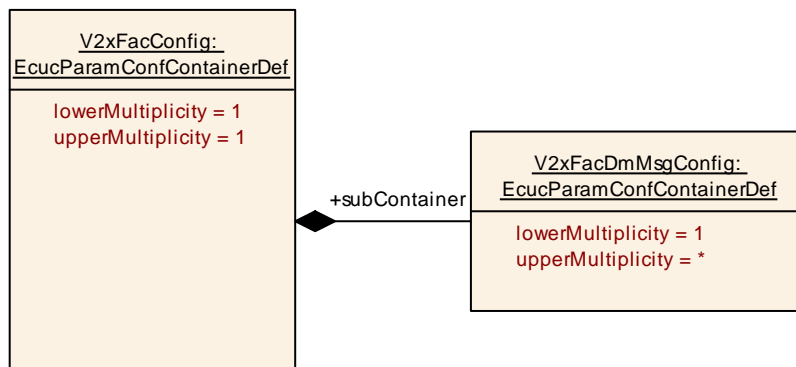
**No Included Containers**



### 10.1.4 V2xFacConfig

<b>SWS Item</b>	[ECUC_V2xFac_00011]
<b>Container Name</b>	V2xFacConfig
<b>Parent Container</b>	V2xFac
<b>Description</b>	This container contains the configuration parameters and sub containers of the AUTOSAR V2xFac module.
<b>Configuration Parameters</b>	

Included Containers		
Container Name	Multiplicity	Scope / Dependency
V2xFacDmMsg-Config	1..*	This container contains the configuration of all messages that shall be passed on to the V2x Data Manager.



### 10.1.5 V2xFacDmMsgConfig

<b>SWS Item</b>	[ECUC_V2xFac_00012]
<b>Container Name</b>	V2xFacDmMsgConfig
<b>Parent Container</b>	V2xFacConfig

<b>Description</b>	This container contains the configuration of all messages that shall be passed on to the V2x Data Manager.		
<b>Post-Build Variant Multiplicity</b>	true		
<b>Multiplicity Configuration Class</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Configuration Parameters</b>			

<b>SWS Item</b>	[ECUC_V2xFac_00013]		
<b>Parameter Name</b>	V2xFacDmPortId		
<b>Parent Container</b>	V2xFacDmMsgConfig		
<b>Description</b>	This parameter contains the destination port identifier that references the V2x message.		
<b>Multiplicity</b>	1		
<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 65535		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

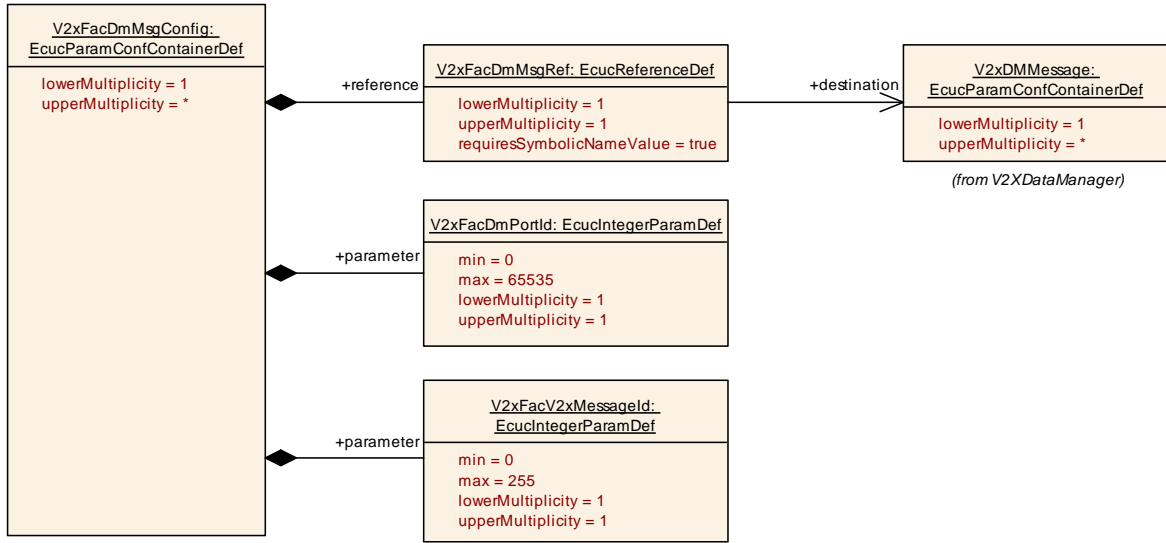
<b>SWS Item</b>	[ECUC_V2xFac_00016]		
<b>Parameter Name</b>	V2xFacV2xMessageld		
<b>Parent Container</b>	V2xFacDmMsgConfig		
<b>Description</b>	<p>This parameter contains the message identifier related to the V2X message that can be received on the associated BTP port as specified in ETSI ITS 102 894-2.</p> <p>Note: Nothing precludes to use several message configurations to receive the same type of message on several BTP ports or several messages on the same BTP port, even if this is not intended by the standard.</p>		
<b>Multiplicity</b>	1		

<b>Type</b>	EcucIntegerParamDef		
<b>Range</b>	0 .. 255		
<b>Default value</b>	--		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

<b>SWS Item</b>	[ECUC_V2xFac_00014]		
<b>Parameter Name</b>	V2xFacDmMsgRef		
<b>Parent Container</b>	V2xFacDmMsgConfig		
<b>Description</b>	Symbolic name reference to [V2xDMMMessage]		
<b>Multiplicity</b>	1		
<b>Type</b>	Symbolic name reference to V2xDMMMessage		
<b>Post-Build Variant Value</b>	false		
<b>Value Configuration Class</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: local		

**No Included Containers**





## 11 Not applicable requirements