Document Title	Specification of I-PDU Multiplexer
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	182
Document Classification	Standard

Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History

Release	Changed by	Change Description
4.2.2	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
4.2.1	AUTOSAR Release Management	• Added Multiple PDU to Container Mapping • Extension of IpduMSelectorFieldLength
4.1.3	AUTOSAR Release Management	• Editorial changes and minor corrections • No major functional change
4.1.2	AUTOSAR Release Management	• Revised configuration structure of dynamic and static segments to enforce layout constraints already by the configuration structure • Few bug fixes and clarifications • Editorial changes • Removed chapter(s) on change documentation
4.1.1	AUTOSAR Administration	• Reworked according to the new SWS_BSWGeneral, harmonization of post-build configuration • Allowing reception of nothing but the static part
4.0.3	AUTOSAR Administration	• Minor bug fixes and editorial changes • Added configurable JIT-update
3.1.5	AUTOSAR Administration	• Updated: tables for mandatory and optional interfaces, SWS_IpduM_00020, SWS_IpduM_00027, SWS_IpduM_00028, SWS_IpduM_00032, SWS_IpduM_00060, SWS_IpduM_00068, SWS_IpduM_00083, SWS_IpduM_00104, ECUC_IpduM_00112, IPDUM117_Conf, SWS_IpduM_00143 and IPDUM162 • Removed: IPDUM013, IPDUM030, IPDUM050_Conf, IPDUM051_Conf, IPDUM063, IPDUM064, IPDUM065, IPDUM072, IPDUM099 and IPDUM154 • Added: pre-compile configuration variant (Chapter 10), ECUC_IpduM_00162, ECUC_IpduM_00163, ECUC_IpduM_00164 and SWS_IpduM_00165

Document Change History		
Release	**Changed by**	**Change Description**
3.1.4	AUTOSAR Administration	• Harmonization of FIBEX multiplexing and AUTOSAR multiplexing • Many small corrections based on conformance tests and validation activities • Legal disclaimer revised
3.1.1	AUTOSAR Administration	• Fixed generated figures and captions
3.0.1	AUTOSAR Administration	• SWS improvements by AUTOSAR Technical Office • Defined maximum I-PDU size for FlexRay to 254 bytes • Document meta information extended • Small layout adaptations made
2.1.15	AUTOSAR Administration	• Integrated into BSW Scheduler header file structure • Sequence diagrams clarified • Superfluous text removed • Maximum IPDU size clarified • Signature for IpduM_Transmit made consistent with rest of stack. • “Advice for users” revised • Revision Information” added • Legal disclaimer revised
2.0	AUTOSAR Administration	• Initial release

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Introduction and functional overview

This specification describes the functionality, APIs and the configuration of the AUTOSAR Basic Software module I-PDU Multiplexer IpduM.

PDU multiplexing means using the same PCI (Protocol Control Information) of a PDU (Protocol Data Unit) with more than one unique layout of its SDU (Service Data Unit). A selector field is a piece of the SDU of the multiplexed PDU. It is used to distinguish the contents of the multiplexed PDUs from each other.

Multiplexing of PDUs is currently known from CAN, but is not restricted to this communication system.

On sender-side, the I-PDU Multiplexer module is responsible to combine appropriate I-PDUs from COM to new, multiplexed I-PDUs and send them back to the PDU Router. On receiver-side, it is responsible to interpret the content of multiplexed I-PDUs and provide COM with its appropriate separated I-PDUs taking into account the value of the selector field.

2 Acronyms and abbreviations

Abbreviation / Acronym:	***Description:***
COM I-PDU	I-PDU assembled in the COM module out of COM Signals
contained I-PDU	I-PDU assembled into or extracted from a Container PDU
Container PDU	PDU containing I-PDUs and headers
dynamic part	see [4]
instance of an I-PDU	IpduM I-PDU with one specific layout and content
Instances of a Container	Instances of the same Container PDU
IpduM	I-PDU Multiplexer
IpduM I-PDU	I-PDU assembled in the IpduM module out of two COM I-PDUs
multiplexed I-PDU	see IpduM I-PDU
segment	The static or dynamic part may consist of more than one piece. These pieces are called segments. See also Chapter 7.2.1 and **Figure 3**.
selector field	see [4]
signal	see [5]
signal group	see [5]
static part	see [4]

3 Related documentation

3.1 Input documents

[1] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf

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

[3] Specification of RTE
AUTOSAR_SWS_RTE.pdf

[4] Requirements on I-PDU Multiplexer
AUTOSAR_SRS_IPDUMultiplexer.pdf

[5] Specification of Communication
AUTOSAR_SWS_COM.pdf

[6] General Specification of Basic Software Modules
AUTOSAR_SWS_BSWGeneral.pdf

3.2 Related standards and norms

None

3.3 Related specification

AUTOSAR provides a General Specification on Basic Software modules [6], which is also valid for IPDU Multiplexer.

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

4 Constraints and assumptions

4.1 Limitations

For transmission of multiplexed I-PDUs, minimum delay time observation cannot be taken into account. For more details, see [5] and 7.2.4.1.

4.2 Applicability to car domains

No restrictions.

4.3 Applicability to safety related environments

This document has been created in absence of a safety case and a safety plan. Thus, the direct results of this document can only be used within safety relevant systems after repeating certain process steps as required in the IEC 61508.

5 Dependencies to other modules

This chapter lists all the features from other modules that are used by the AUTOSAR IpduM and functionalities that are provided by AUTOSAR IpduM to other modules. Because the IpduM module deals with PDUs that are either sourced or sunk by other modules, care must be taken that shared configuration items are consistent between the modules.

The IpduM is arranged next to the PDU Router in the layered architecture of AUTOSAR; see [1] and **Figure 1**.

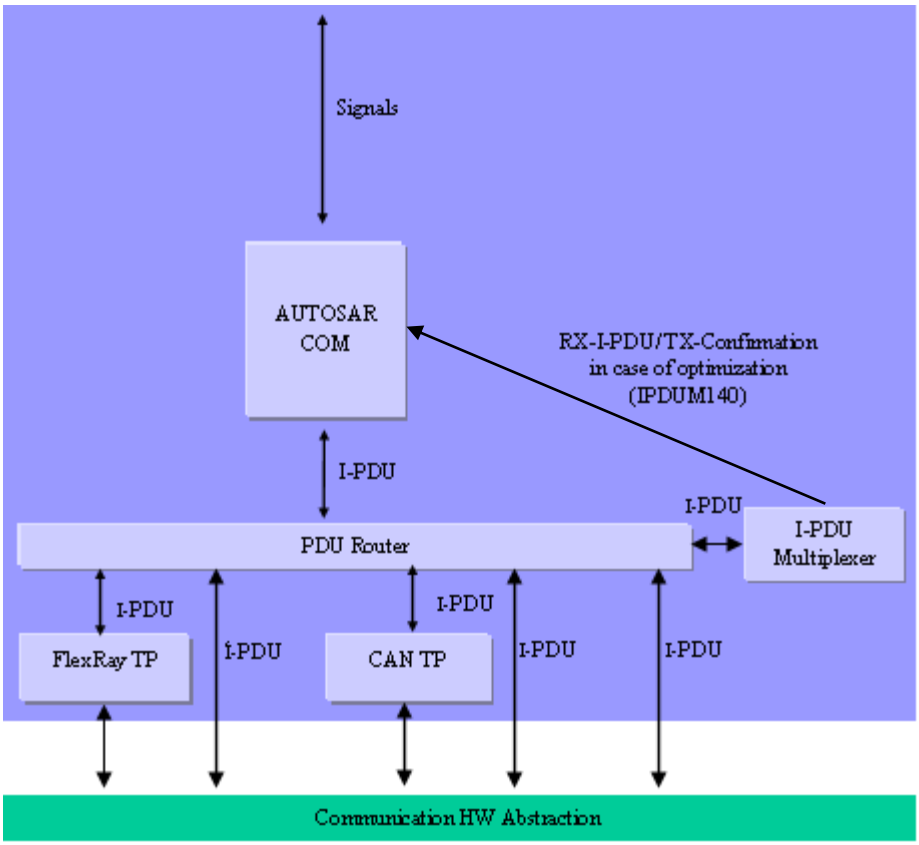

Figure 1 I-PDU Multiplexer in the AUTOSAR Architecture

5.1 AUTOSAR OS

[SWS_IpduM_00107] [The IpduM shall not directly access the AUTOSAR OS.] (SRS_BSW_00429)

5.2 RTE (BSW Scheduler)

The RTE includes the BSW-Scheduler (see [3]).

The IpduM module relies on the BSW-scheduler calling the IpduM_MainFunction function at a period as configured in IpduMConfigurationTimeBase.

5.3 PDU Router

The following summarizes the functionality IpduM needs from the PDU Router (for more details see Chapter 8.6):

- indication of incoming multiplexed or contained I-PDUs
- sending interface for outgoing I-PDUs (Container or Multiplexed PDUs)
- confirmation of I-PDUs which went out

The following list summarizes the functionality provided by the IpduM module for the PDU Router module:

- indication interface for incoming I-PDUs, which are de-multiplexed and for incoming Container-PDUs, which are to be disassembled
- sending interface for to be multiplexed I-PDUs and I-PDUs, which are to be assembled into a Container PDU
- confirmation interface for transmitted I-PDUs

The configuration of the PDU Router module (e.g. look-up tables) must be such that the I-PDUs, which belong to multiplexed I-PDUs and represent a static or a dynamic part of a multiplexed I-PDU, are routed to the IpduM module.

The configuration of the PDU-Router module (e.g. look-up tables) must be such that the relevant I-PDUs are routed to IpduM. These are:

- I-PDUs, which belong to multiplexed I-PDUs and represent a static or a dynamic part of a multiplexed I-PDU
- I-PDUs, which consist of static and dynamic parts to be de-multiplexed
- I-PDUs, which are to be assembled into a Container PDU
- Container PDUs to be disassembled

5.4 COM

The configuration of the IpduM module relies on a corresponding configuration of the AUTOSAR COM module. For each multiplexed I-PDU, there need to be different I-PDUs for the static part and each layout of the dynamic part. For further information configured in the COM module, see Chapter 7.1 and especially **Figure 3**.
The IpduM further assumes that the correct selector field values are already contained in the COM's modules I-PDU representing the dynamic parts. See also SWS_IpduM_00098.
The configuration of Container PDUs/ contained I-PDUs does not depend on the COM configuration.

5.5 File structure

5.5.1 Code file structure

This IpduM SWS does not define the code file structure completely.

5.5.2 Header file structure

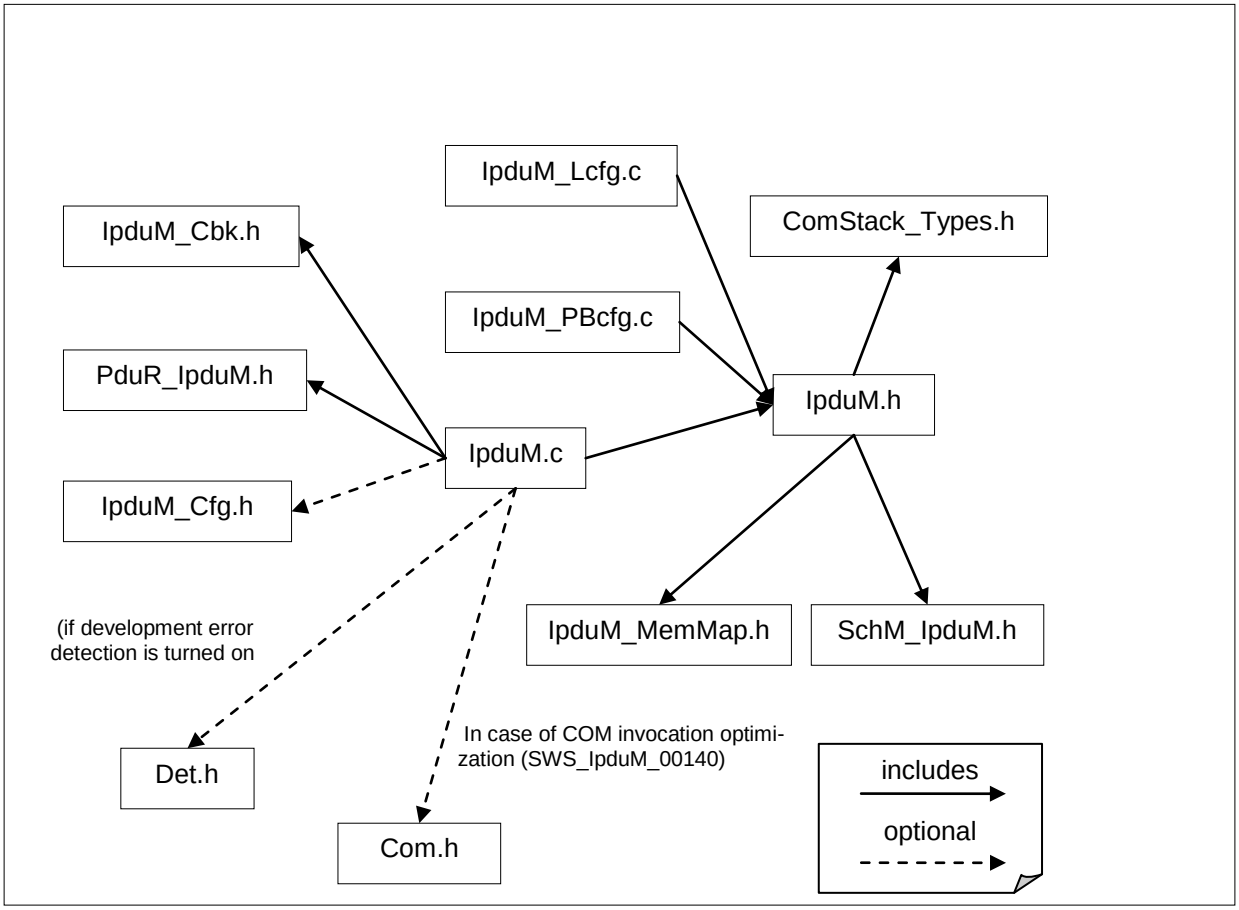

Figure 2 Header File Structure

[SWS_IpduM_00148] ⌈ The file IpduM.c shall include IpduM.h, IpduM_Cbk.h, PduR_IpduM.h, and optionally IpduM_Cfg.h, Det.h and Com.h.⌋ (SRS_BSW_00415)

[SWS_IpduM_00149] ⌈ The file IpduM_Lcfg.c shall include IpduM.h.⌋ (SRS_BSW_00415)

[SWS_IpduM_00150] ⌈ The file IpduM_PBcfg.c shall include IpduM.h.⌋ (SRS_BSW_00415)

[SWS_IpduM_00151] ⌈ File IpduM.h shall include MemMap.h, SchM_IpduM.h and ComStack_Types.h.⌋ (SRS_BSW_00415)

6 Requirements traceability

Requirement	Description	Satisfied by
SRS_BSW_00003	All software modules shall provide version and identification information	SWS_IpduM_00037
SRS_BSW_00005	Modules of the ÂµC Abstraction Layer (MCAL) may not have hard coded horizontal interfaces	SWS_IpduM_00999
SRS_BSW_00009	All Basic SW Modules shall be documented according to a common standard.	SWS_IpduM_00104, SWS_IpduM_00105
SRS_BSW_00101	The Basic Software Module shall be able to initialize variables and hardware in a separate initialization function	SWS_IpduM_00032, SWS_IpduM_00033
SRS_BSW_00162	The AUTOSAR Basic Software shall provide a hardware abstraction layer	SWS_IpduM_00999
SRS_BSW_00164	The Implementation of interrupt service routines shall be done by the Operating System, complex drivers or modules	SWS_IpduM_00999
SRS_BSW_00168	SW components shall be tested by a function defined in a common API in the Basis-SW	SWS_IpduM_00999
SRS_BSW_00171	Optional functionality of a Basic-SW component that is not required in the ECU shall be configurable at pre-compile-time	SWS_IpduM_00999
SRS_BSW_00314	All internal driver modules shall separate the interrupt frame definition from the service routine	SWS_IpduM_00999
SRS_BSW_00323	All AUTOSAR Basic Software Modules shall check passed API parameters for validity	SWS_IpduM_00028
SRS_BSW_00325	The runtime of interrupt service routines and functions that are running in interrupt context shall be kept short	SWS_IpduM_00999
SRS_BSW_00336	Basic SW module shall be	SWS_IpduM_00999

	able to shutdown	
SRS_BSW_00337	Classification of development errors	SWS_IpduM_00026, SWS_IpduM_00153, SWS_IpduM_00162
SRS_BSW_00339	Reporting of production relevant error status	SWS_IpduM_00999
SRS_BSW_00344	BSW Modules shall support link-time configuration	SWS_IpduM_00032
SRS_BSW_00357	For success/failure of an API call a standard return type shall be defined	SWS_IpduM_00102
SRS_BSW_00369	All AUTOSAR Basic Software Modules shall not return specific development error codes via the API	SWS_IpduM_00032, SWS_IpduM_00037, SWS_IpduM_00040, SWS_IpduM_00043, SWS_IpduM_00044, SWS_IpduM_00060
SRS_BSW_00375	Basic Software Modules shall report wake-up reasons	SWS_IpduM_00999
SRS_BSW_00377	A Basic Software Module can return a module specific types	SWS_IpduM_00999
SRS_BSW_00386	The BSW shall specify the configuration for detecting an error	SWS_IpduM_00999
SRS_BSW_00405	BSW Modules shall support multiple configuration sets	SWS_IpduM_00032
SRS_BSW_00406	A static status variable denoting if a BSW module is initialized shall be initialized with value 0 before any APIs of the BSW module is called	SWS_IpduM_00084
SRS_BSW_00407	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	SWS_IpduM_00037
SRS_BSW_00414	Init functions shall have a pointer to a configuration structure as single parameter	SWS_IpduM_00162, SWS_IpduM_00174
SRS_BSW_00415	Interfaces which are provided exclusively for one module shall be separated into a dedicated header file	SWS_IpduM_00148, SWS_IpduM_00149, SWS_IpduM_00150, SWS_IpduM_00151
SRS_BSW_00417	Software which is not part of the SW-C shall report error events only after the DEM is fully operational.	SWS_IpduM_00999
SRS_BSW_00422	Pre-de-bouncing of error status information is done	SWS_IpduM_00999

	within the DEM	
SRS_BSW_00423	BSW modules with AUTOSAR interfaces shall be describable with the means of the SW-C Template	SWS_IpduM_00999
SRS_BSW_00425	The BSW module description template shall provide means to model the defined trigger conditions of schedulable objects	SWS_IpduM_00103
SRS_BSW_00427	ISR functions shall be defined and documented in the BSW module description template	SWS_IpduM_00999
SRS_BSW_00429	BSW modules shall be only allowed to use OS objects and/or related OS services	SWS_IpduM_00107
SRS_BSW_00432	Modules should have separate main processing functions for read/receive and write/transmit data path	SWS_IpduM_00999
SRS_BSW_00433	Main processing functions are only allowed to be called from task bodies provided by the BSW Scheduler	SWS_IpduM_00999
SRS_BSW_00437	Memory mapping shall provide the possibility to define RAM segments which are not to be initialized during startup	SWS_IpduM_00999
SRS_BSW_00438	Configuration data shall be defined in a structure	SWS_IpduM_00159
SRS_IpduM_02801	The size in bits of the selector field shall be configurable	SWS_IpduM_00173
SRS_IpduM_02802	The position of the selector field within the PDU shall be configurable	SWS_IpduM_00173
SRS_IpduM_02803	It shall be possible not to assign a SDU layout to the unused selector field values	SWS_IpduM_00011
SRS_IpduM_02807	The I-PDU Multiplexer module shall be designed in a way that it does not produce any additional runtime	SWS_IpduM_00097
SRS_IpduM_02809	The initial values of the static part shall be derived	SWS_IpduM_00067, SWS_IpduM_00068, SWS_IpduM_00098, SWS_IpduM_00143

	from the COM configuration	
SRS_IpduM_02810	The PduR shall be configured to send parts of multiplexed I-PDUs to the IPduM on sender side	SWS_IpduM_00089, SWS_IpduM_00090, SWS_IpduM_00091
SRS_IpduM_02811	There shall be three different triggering conditions configurable that define when the combined multiplexed I-PDUs are sent to the lower layer	SWS_IpduM_00021, SWS_IpduM_00168
SRS_IpduM_02812	The PduR shall be configured to send multiplexed I-PDUs for de-multiplexing to the IPduM after they were received from the lower layer	SWS_IpduM_00041, SWS_IpduM_00042, SWS_IpduM_00086, SWS_IpduM_00140
SRS_IpduM_02813	The PduR shall be configured to send confirmations related to multiplexed I-PDUs to IPduM after receiving them from the lower layer	SWS_IpduM_00022, SWS_IpduM_00101
SRS_IpduM_02814	The confirmation shall depend upon selector field	SWS_IpduM_00019, SWS_IpduM_00020, SWS_IpduM_00023, SWS_IpduM_00024, SWS_IpduM_00087, SWS_IpduM_00088, SWS_IpduM_00152
SRS_IpduM_02816	On sender side the IPduM shall combine the static and the appropriate dynamic part within IPduM	SWS_IpduM_00015, SWS_IpduM_00017, SWS_IpduM_00166, SWS_IpduM_00169, SWS_IpduM_00171, SWS_IpduM_00172, SWS_IpduM_00223
SRS_IpduM_02817	On receiver side the IPduM extracts the static and dynamic parts of the multiplexed I-PDU	SWS_IpduM_00040, SWS_IpduM_00166
SRS_IpduM_02818	The IPduM confirms to COM the static part of the multiplexed I-PDU and the dynamic part	SWS_IpduM_00022
SRS_IpduM_02819	There shall be no queuing of transmission requests on sender side	SWS_IpduM_00020, SWS_IpduM_00023
SRS_IpduM_02820	Multiple I-PDU Mapping	SWS_IpduM_00175, SWS_IpduM_00179, SWS_IpduM_00180, SWS_IpduM_00181, SWS_IpduM_00182, SWS_IpduM_00183, SWS_IpduM_00184, SWS_IpduM_00185, SWS_IpduM_00186, SWS_IpduM_00187, SWS_IpduM_00188, SWS_IpduM_00189, SWS_IpduM_00190, SWS_IpduM_00191, SWS_IpduM_00192, SWS_IpduM_00193, SWS_IpduM_00194, SWS_IpduM_00195, SWS_IpduM_00196, SWS_IpduM_00197, SWS_IpduM_00198, SWS_IpduM_00199, SWS_IpduM_00200, SWS_IpduM_00201,

		SWS_IpduM_00202, SWS_IpduM_00203, SWS_IpduM_00207, SWS_IpduM_00208, SWS_IpduM_00210, SWS_IpduM_00211, SWS_IpduM_00212, SWS_IpduM_00213, SWS_IpduM_00214, SWS_IpduM_00215, SWS_IpduM_00216, SWS_IpduM_00217, SWS_IpduM_00218, SWS_IpduM_00220
SRS_IpduM_02821	The temporal order of I-PDUs shall be preserved	SWS_IpduM_00209, SWS_IpduM_00219, SWS_IpduM_00221, SWS_IpduM_00222
SRS_IpduM_02822	Two different Header Sizes shall be supported	SWS_IpduM_00177
SRS_IpduM_02823	The position of I-PDUs inside a Container shall be dynamic	SWS_IpduM_00178
SRS_IpduM_02824	The ID used in the header shall be independent of the Container	SWS_IpduM_00204, SWS_IpduM_00205, SWS_IpduM_00206, SWS_IpduM_00207

7 Functional specification

7.1 General

There are two different approaches of multiplexing several I-PDUs into one resulting PDU being transferred on the bus:

I-PDU Multiplexing means using the same I-PDU ID transferred from the PDU Router to the Communication Hardware Abstraction Layer with more than one unique layout of this I-PDU; see also [1].

Multiple PDU to Container Mapping means collecting several I-PDUs into one Container PDU. This Container PDU is then transferred via PduR as one (large) I-PDU. This way advantage of the larger frame sizes of newer bus systems can be taken, allowing an efficient usage of the bandwidth in combination with smaller I-PDU sizes (usually 8 bytes).

[SWS_IpduM_00097] ⌈The IpduM shall be implemented so that no other modules depend on it and that it is be possible to build a system without the IpduM module if it is not needed.⌋ (SRS_IpduM_02807)

7.2 I-PDU Multiplexing

7.2.1 Definitions and Layout

A multiplexed I-PDU consists of a static part and a dynamic part, where the static part consists of zero or more signals or signal groups. The dynamic part consists of the selector field and one or more signals or signal groups; see **Figure 3**.
The dynamic part of an I-PDU is comparable with a union of the programming language ***C***. Depending on the value of the selector field inside the I-PDU, the actual layout of the I-PDU is selected.

The position of the static and the dynamic part are configurable per I-PDU. The static and the dynamic part can be subdivided into different segments.
Only one selector field can be defined for each multiplexed I-PDU. The value of the selector field defines how the content of the dynamic part of the I-PDU will be interpreted. The selector field has a configurable size between 1 and 16 contiguous bits and its position can be defined by configuration, see ECUC_IpduM_00054.
See Chapter 10.2.2 for an overview of the IpduM configuration. Chapter 10.4 defines the configuration rules.

Multiplexing of PDUs is originally known from CAN, but it is not restricted to this communication system. The IpduM is layered next to the PDU Router above the interface layer (Communication Hardware Abstraction) in the AUTOSAR layer architecture and therefore this feature could be used for all bus systems, which can be handled by the PDU Router, for example FlexRay.

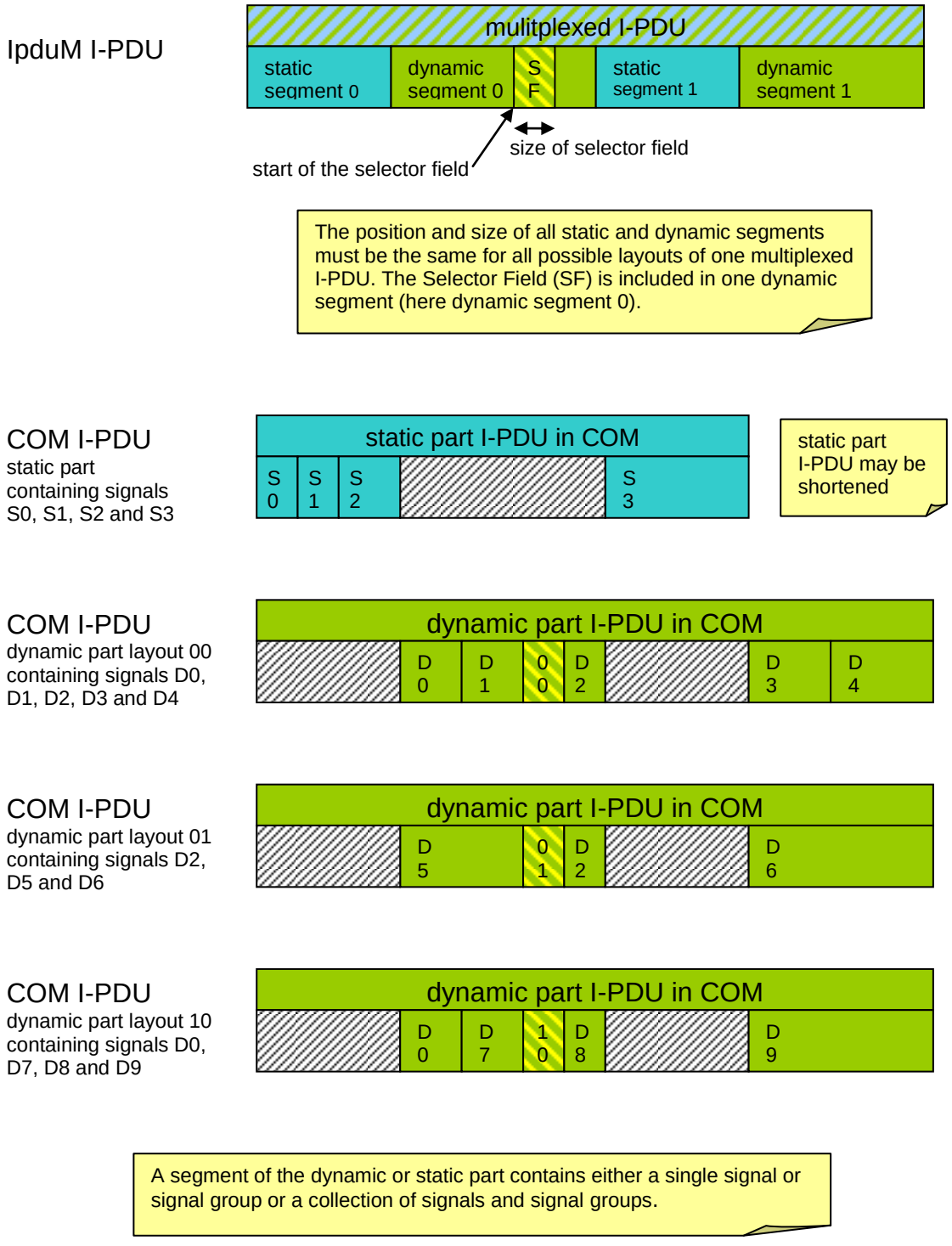

Figure 3 Possible layout of a multiplexed I-PDU with shortened static part

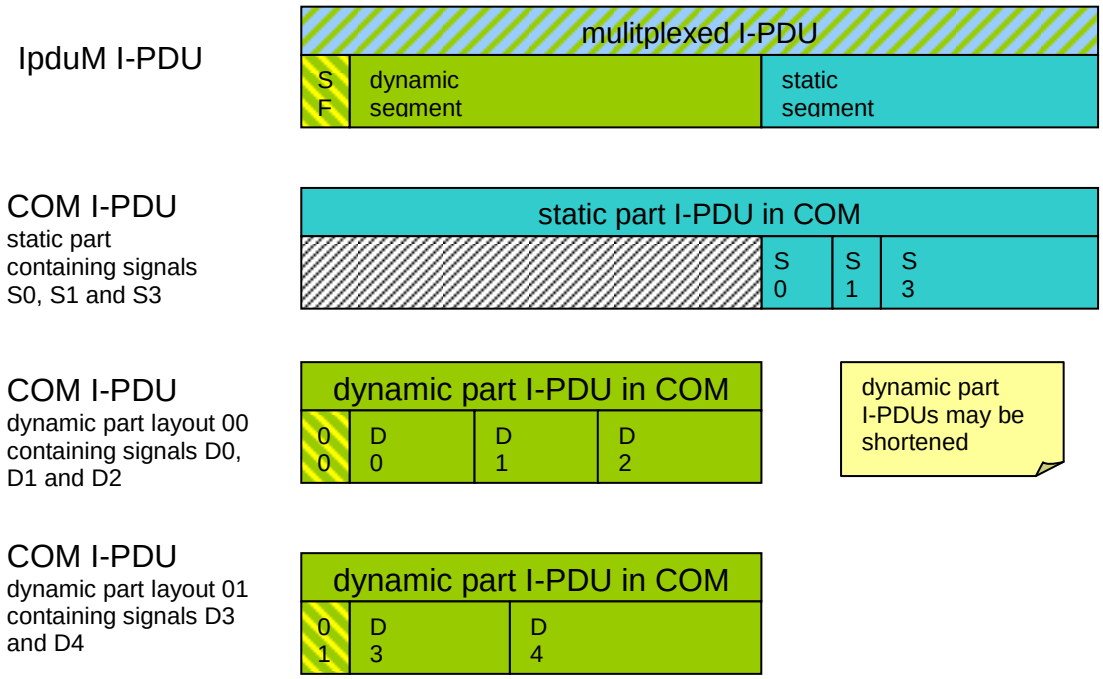

Figure 4 Possible layout of a multiplexed I-PDU with shortened dynamic parts

7.2.2 General

There is one COM I-PDU for the static part and one COM I-PDU for each layout of the dynamic part of one multiplexed IpduM I-PDU, so the IpduM combines at most two I-PDUs of COM.

[SWS_IpduM_00098] [The IpduM module shall not set the selector field.] (SRS_IpduM_02809)

The IpduM module relies on the configuration of the COM module. For each dynamic layout, an I-PDU needs to be configured in COM. Such I-PDUs already have to contain the correct selector field value. The selector field values in COM can be initialized by configuring them as signals that are initialized with an init value but are never written after initialization.

[SWS_IpduM_00173] [The IpduM shall respect the IpduMByteOrder when interpreting the selector field value.] (SRS_IpduM_02801, SRS_IpduM_02802)

For a detailed description of the transmission and reception of a multiplexed I-PDU see Chapter 7.2.4 and 7.2.5.

[SWS_IpduM_00140] [It shall be allowed to optimize the Rx- and Tx-Confirmation path from the IpduM module via the PDU Router module to the COM layer to call the COM API directly from the IpduM module without including the PDU Router. This

shall be indicated by setting the published parameter IpduMRxDirectComInvocation to TRUE, see ECUC_IpduM_00142. ⌋ (SRS_IpduM_02812)

In case of the COM invocation, optimization as defined above IpduM.c needs to include Com.h, see Figure 2 Header File Structure.

7.2.3 Initialization

The IpduM module provides an initialization function IpduM_Init defined in SWS_IpduM_00032. This function initializes all internal global variables and the buffers of the IpduM I-PDUs. For more details, see Chapter 8.3.1.

The environment of the IpduM shall call IpduM_Init before calling any other function of the IpduM module.

The implementer has to ensure that IPDUM_E_UNINIT is returned in development mode in case an API function is called before the module is initialized.

For the I-PDU data transmission pathway through the IpduM module, a buffer is allocated inside the IpduM module. This buffer needs to be initialized because it might be transmitted before it has been fully populated with data by the COM module. The initialization data of this buffer is derived from the initial values of the COM module's configuration as follows:

1) **[SWS_IpduM_00067]** ⌈The IpduM shall initialize its internal transmit buffers with the configured pattern IpduMIPduUnusedAreasDefault.⌋ (SRS_IpduM_02809)

2) **[SWS_IpduM_00068]** ⌈The initial signal values of the initial dynamic part shall be set according to initial values of the referenced COM I-PDU (IpduMInitialDynamicPart -> IpduMTxDynamicPart -> IpduMTxDynamicPduRef).⌋ (SRS_IpduM_02809)

3) **[SWS_IpduM_00143]** ⌈The initial signal values of the static part shall be set according to the intial values of the referenced COM I-PDU (IpduMTxStaticPart -> IpduMTxStaticPduRef)⌋ (SRS_IpduM_02809)

The selector field is contained within one segment of the intial dynamic part and therefore is initialized implicitly.

For optimization, the initial bit pattern for the buffer can be worked out at configuration-time and then copied at run-time.

7.2.4 Transmission

Inside COM, there are separated I-PDUs for the static part and one for each dynamic part of a multiplexed I-PDU.

The static part and the dynamic parts are treated in COM as separate I-PDUs with their own I-PDU IDs.

[SWS_IpduM_00015] ⌈For a multiplexed I-PDU IpduM shall merge the corresponding two COM I-PDUs representing the associated static part and the last received dynamic part into one single IpduM I-PDU with a new unique I-PDU ID. IpduM shall send out this new IpduM I-PDU to the PDU Router module, see also **Figure 1**.⌋ (SRS_IpduM_02816)

For details about the trigger of the transmission, see Chapter 7.2.4.2.

All control functionalities like deadline monitoring of the COM I-PDUs and update-bit evaluation are out of the scope of the IpduM and have to be done by the COM layer. For details about the timing-behavior of the new combined I-PDU see Chapter 7.2.4.2.

7.2.4.1 Transmission request

The IpduM module provides an IpduM_Transmit function so that the PDU Router is able to initiate the transmission of an I-PDU; see SWS_IpduM_00043.

[SWS_IpduM_00017] ⌈The function IpduM_Transmit shall assemble the multiplexed I-PDU, using the related static and dynamic part, and transmit it according to the trigger conditions/ modes as defined in SWS_IpduM_00021 and ECUC_IpduM_00125.⌋ (SRS_IpduM_02816)

As defined in Chapter 7.2.3, each outgoing I-PDU has an initial value so that, should an I-PDU be transmitted by the IpduM module before both static and dynamic parts have been sent from COM to the IpduM, a value defined by the configuration is transmitted.

[SWS_IpduM_00019] ⌈The configuration of the IpduM shall contain a dedicated timeout for each IpduM I-PDU within the IpduM module in the configuration parameter IpduMTxConfirmationTimeout. ⌋ (SRS_IpduM_02814)

This timeout defines until when the transmission confirmation for this I-PDU has to be received after the transmission. For transmission confirmation, see Chapter 7.2.4.3.

The timeout period shall take into account the delays in the lower layers.

[SWS_IpduM_00020] ⌈In case the IpduMTxConfirmationTimeout was configured to a value greater than 0, as long as the corresponding timeout timer has not elapsed, and no transmission confirmation for that multiplexed I-PDU was received, the function IpduM_Transmit shall not allow a new transmission request from the upper layer with a COM I-PDU that belongs to the same IpduM I-PDUs.⌋ (SRS_IpduM_02814, SRS_IpduM_02819)

In case IpduMTxConfirmationTimeout was omitted or configured to 0, the IpduM module does not block any new transmission requests.

[SWS_IpduM_00152] [As long as the timeout (defined in the configuration parameter IpduMTxConfirmationTimeout) has not elapsed and as long as no transmission confirmation for the IpduM I-PDU is received, the function IpduM_Transmit shall return with E_NOT_OK for a new transmission request from the upper layer with a COM I-PDU that belongs to the same IpduM I-PDUs.] (SRS_IpduM_02814)

If the IpduMTxConfirmationTimeout is omitted or configured to 0, the parts of the multiplexed I-PDU may be overwritten even in case they were not already sent or confirmed.

In case a multiplexed I-PDU is only triggered for sending by either updating the dynamic or static part, the non-triggering part might be overwritten if updated multiple times between two transmissions even with a configured IpduMTxConfirmationTimeout. This happens, since the confirmation timeout timer is only started, if the triggering part is updated.

It may be useful to configure the IpduM transmission confirmation timeout depended of the transmission deadline monitoring timeouts for the single COM I-PDUs of the COM layer configuration; see also [5].

7.2.4.2 Transmission trigger

The IpduM module receives the static and the dynamic part of a multiplexed I-PDU by separated two transmission requests as two single COM I-PDUs from the PDU Router module.

[SWS_IpduM_00021] [The IpduM module shall be configurable to send a transmission request for the new multiplexed I-PDU to the PDU Router because of the following trigger conditions/ modes:

- receiving a static part
- receiving a dynamic part
- receiving a static or a dynamic part
- does not trigger transmission because of receiving anything of this I-PDU (IpduMTxTriggerMode None) in case of TriggerTransmit

For configuration, see ECUC_IpduM_00052.] (SRS_IpduM_02811)

The four trigger conditions/ modes defined by SWS_IpduM_00021 allow controlling the transmission mode of the new assembled I-PDU by the transmission modes of the single I-PDUs sent by COM, see also [5].

Not all of four trigger conditions/ modes defined by SWS_IpduM_00021 guarantee the minimum delay time between consecutive transmissions of different instances of multiplexed I-PDUs, because if the transmission is triggered by static and dynamic part or only by the dynamic part, COM does not take care for the minimum delay time. COM treats the static part and the different dynamic parts as unrelated standalone I-PDUs.

The configuration “does not trigger transmission because of receiving anything” is needed if an I-PDU is only sent out because of a TriggerTransmit of a lower layer. With the API IpduM_TriggerTransmit it is possible for lower layers to trigger a send out of an I-PDU.

In case the IpduMTxTriggerMode is None and the lower layer triggers the transmission via IpduM_TriggerTransmit, the IpduMTxConfirmationPduId needs to be configured since this ID is also used for resolving the I-PDU in case of IpduM_TriggerTransmit, see also ECUC_IpduM_00158.

7.2.4.3 Just-In-Time update of parts

Sometimes it may be unwanted that the IpduM module not just sends out the locally stored parts, since these parts may contain outdated information e.g. update-bits. Therefore, the IpduM supports a per part configurable just-in-time update mechanism.

[SWS_IpduM_00168] ⌈In case the transmission of a multiplexed I-PDU is triggered by the update of one part and IpduMJitUpdate is configured to true for the second part, the IpduM module shall update the second part via PduR_IpduMTriggerTransmit before the multiplexed I-PDU is sent out via PduR_IpduMTransmit.⌋ (SRS_IpduM_02811)

[SWS_IpduM_00169] ⌈In case the contents of a multiplexed I-PDU is requested via IpduM_TriggerTransmit, the IpduM module shall update all parts which have IpduMJitUpdate configured to *true* before returning the contents of the multiplexed I-PDU.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00223] ⌈In case the IpduM shall update the dynamic part just-in-time, the latest dynamic part sent by the upper layer shall be updated or the dynamic part referenced by IpduMInitialDynamicPart if no dynamic part was sent before.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00171] ⌈In case the transmission of a multiplexed I-PDU is triggered by the update of one part and IpduMJitUpdate is configured to *true* for the second part, the multiplexed I-PDU shall not be send if the JIT-update request via PduR_IpduMTriggerTransmit returns E_NOT_OK.⌋ (SRS_IpduM_02816)

[SWS_IpduM_00172] ⌈In case the contents of a multiplexed I-PDU is requested via IpduM_TriggerTransmit and IpduMJitUpdate is configured to *true* for any multiplexed part, IpduM_TriggerTransmit shall return E_NOT_OK if any of the JIT-update requests via PduR_IpduMTriggerTransmit return E_NOT_OK.⌋ (SRS_IpduM_02816)

7.2.4.4 Transmission confirmation

Transmission confirmations are given to the IpduM module by the PDU Router according to the configuration of the I-PDUs in the PDU Router.

[SWS_IpduM_00022] ⌈If the IpduM receives a TxConfirmation for a specific IpduM I-PDU, it shall translate this confirmation into the corresponding confirmations for the

COM I-PDUs, which were contained in the last sent out multiplexed IpduM I-PDU.] (SRS_IpduM_02813, SRS_IpduM_02818)

Depending on the configuration of IpduMTxDynamicConfirmation (ECUC_IpduM_00163) and IpduMTxStaticConfirmation (ECUC_IpduM_00164), the IpduM will pass zero, one or two confirmations towards COM for one send request. The number of confirmations given to the upper layer does not depend on the IpduMTxTriggerMode.

Examples:
a) If neither IpduMTxDynamicConfirmation nor IpduMTxStaticConfirmation for the corresponding IpduMTxRequest is configured to true, no COM confirmation is generated.
b) If IpduMTxStaticConfirmation is configured to true but and IpduMTxDynamic-Confirmation is configured to false (or vice versa), then only one COM confirmation is generated.
c) If both IpduMTxStaticConfirmation and IpduMTxDynamicConfirmation is configured to true, then two COM confirmations are generated; to the I-PDU representing the static part and the I-PDU representing the dynamic part.

[SWS_IpduM_00023] [If the Tx-Confirmation is not received within the configured timeout IpduMTxConfirmationTimeout the IpduM shall allow new transmission requests for this specific I-PDU after timeout is elapsed.] (SRS_IpduM_02814, SRS_IpduM_02819)

[SWS_IpduM_00024] [The IpduM shall discard unexpected Tx-Confirmations silently. This may happen if a previously requested transmit request has been timed out, but is confirmed now.] (SRS_IpduM_02814)

There is no need for an error entry in the case of timeout violation because this is already done in COM, if needed. In the case of a proper configuration of the communication stack, the timeout violation in the IpduM modules occurs at the same time than the Deadline Monitoring violation in the COM module.

7.2.5 Reception

Every I-PDU which is received by the Communication Hardware Abstraction (CAN Interface, Lin Interface, FlexRay Interface) is given to the PDU Router. The PDU Router routes multiplexed I-PDUs to the IpduM module. The IpduM module separately routes the static and dynamic parts of the multiplexed I-PDU to their destinations.

It is known at configuration-time which incoming I-PDU IDs correspond to multiplexed I-PDUs with a static part configured. The I-PDU ID is all that is necessary to work out if there is a static part present.

As all multiplexed I-PDUs contain a dynamic part this part always has to be routed.

There are no requirements to handle or notify wrongly configured parts. Hence, if the received I-PDU contains segments not configured for reception on this ECU, they will

be ignored silently. Furthermore, if an I-PDU is configured with a PduLength of 0, it will also be ignored silently, since no meaningful processing can be configured.

This situation might occur in a gateway setting, if a multiplexed I-PDU is always routed onto another bus by the PDU Router, but contains a signal in one dynamic part that must be passed to the application. In this case, the multiplexed PDU would have to be routed to the IpduM as well.

7.3 Multiple-PDU-to-Container handling

IpduM supports a mapping of several I-PDUs to one Container PDU. Both contained and Container PDUs are *regular* PDUs from PduR's point of view.

IpduM relies on PduR beeing configured to forward send-PDUs mapped to a Container-PDU and received Container-PDUs to IpduM.

7.3.1 Layout of a Container

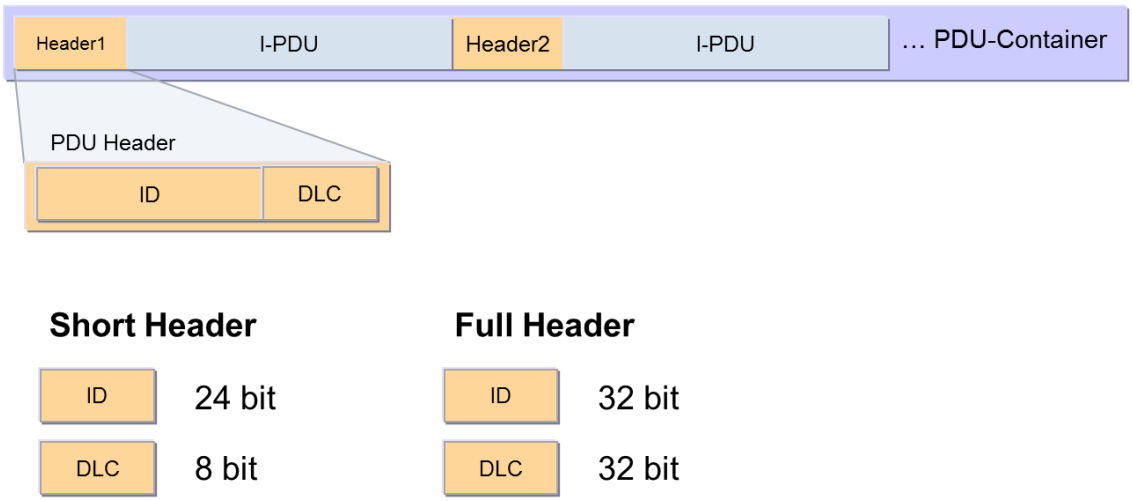

Figure 5 Layout of a Container

[SWS_IpduM_00175] [Inside a Container PDU IpduM shall place the header of a contained I-PDU in front of the contained I-PDU.] (SRS_IpduM_02820)

See also **Figure 5**: Layout of a Container PDU.

There is no configuration of the positions of contained I-PDUs inside the Container PDU, thus the position of an arbitrary contained I-PDU is determined by the length of payload (DLC) and headers of the preceding (added before) contained I-PDUs.

The number of I-PDUs contained in a Container PDU is limited by the maximum size of the Container PDU (PDULength of ECUC-PDU).

The order of the I-PDUs inside the Container PDU will be retained. This way all contained I-PDUs are extracted in the same order as they have been put into the Container PDU. See SWS_IpduM_00179 and SWS_IpduM_00209.

The IpduM supports two different header sizes (see ECUC_IpduM_00183: IpduMContainerHeaderSize):

- IPDUM_HEADERTYPE_SHORT with 24 bit ID and 8 bit length
- IPDUM_HEADERTYPE_LONG with 32 bit ID and 32 bit length

The header sizes are configured per Container PDU via IpduMContainerHeaderSize. Thus, it is valid for the whole Container PDU. Mixing of header sizes inside one Container PDU is not supported.

[SWS_IpduM_00177] ⌈Each I-PDU header shall consist of ID field and length field in the byte order determined by IpduMHeaderByteOrder.⌋ (SRS_IpduM_02822)

[SWS_IpduM_00178] ⌈Placing of headers and payloads of contained I-PDUs inside a Container PDU shall be contiguous without any gap.⌋ (SRS_IpduM_02823)

Rationale: This allows iterating over a Container PDU by considering the header size and payload lengths (DLC from header).

This has to be ensured by the implementation of the container collection algorithm, since contained I-PDUs have no dedicated (configured) position inside a Container PDU.

7.3.2 Transmission

Due to the following requirements IpduM will make sure that instances of a contained I-PDU (same PDU-ID) are transmitted (passed to PduR inside their Container PDUs) in exactly the same order as they are passed to IpduM.

[SWS_IpduM_00179] ⌈When a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to *IPDUM_COLLECT_QUEUED* (see ECUC_IpduM_00198) is passed to IpduM via IpduM_Transmit, IpduM shall identify the associated Container PDU and append the contained I-PDU to its payload even if a previous instance of the contained I-PDU is already present in that Container PDU.⌋ (SRS_IpduM_02820)

This way a Container PDU can include more than one instance of the same I-PDU. The resulting behavior is FIFO-like in order to preserve the order of I-PDU instances being transmitted. Thus, the upper layer(s) of the receiving IpduM can implement either last-is-best or FIFO semantics.

[SWS_IpduM_00180] ⌈If a contained I-PDU has been added to a Container PDU that has not been triggered yet, and if the resulting payload is bigger than IpduMContainerTxSizeThreshold the Container PDU shall be triggered.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00181] ⌈When adding a contained I-PDU to a Container PDU which has not been triggered yet, and if IpduMContainedTxPduTrigger is set to

IPDUM_TRIGGER_ALWAYS, the Container PDU shall be triggered immediately.] (SRS_IpduM_02820)

[SWS_IpduM_00182] [If IpduMContainerTxTriggerMode is set to *IPDUM_DIRECT* and adding a contained I-PDU would exceed maximum size of the Container I-PDU, first the Container PDU shall be triggered. The contained I-PDU shall be added to a new instance of the Container PDU.] (SRS_IpduM_02820)

SWS_IpduM_00189 has to be considered also in case both SWS_IpduM_00181 and SWS_IpduM_00182 apply.

[SWS_IpduM_00183] [If IpduMContainerTxTriggerMode is set to *IPDUM_-TRIGGERTRANSMIT* and adding a contained I-PDU would exceed maximum size of the Container PDU, first the Container PDU shall be queued. Then the contained I-PDU shall be added to a new instance of the Container PDU.] (SRS_IpduM_02820)

Contained I-PDUs will be added to Container PDUs with IpduMContainerTxTrigger-Mode = IPDUM_TRIGGERTRANSMIT as long as they are neither full nor queued.

[SWS_IpduM_00184] [When adding the first contained I-PDU to a Container PDU, IpduM shall start the transmission timer of the Container PDU. The timer shall be initialized with either the Container PDU's timeout (IpduMContainerTxSendTimeout) or the contained I-PDU's timeout (IpduMContainedTxPduSendTimeout) whichever is smaller.] (SRS_IpduM_02820)

Rationale: This way a transmission is requested for a time-triggered bus.

Until the Container PDU is fetched (see SWS_IpduM_00194) or unless maximum size of the Container PDU is not exceeded further requested I-PDUs assigned to this container can be added.

[SWS_IpduM_00185] [When a contained I-PDU is added to a Container PDU, the transmission timer of the Container PDU shall be updated with the contained I-PDU's timeout (IpduMContainedTxPduSendTimeout) if it is less than the remaining time of the Container PDU.] (SRS_IpduM_02820)

[SWS_IpduM_00186] [When the transmission timer of the Container PDU has elapsed, the Container PDU shall be triggered.] (SRS_IpduM_02820)

[SWS_IpduM_00187] [After a Container PDU is triggered or being fetched by Trig-gerTransmit, IpduM shall calculate the overall size of the Container PDU. The total size builds up by the total of all payloads of the contained I-PDUs plus the total length of the corresponding headers. The result shall be the payload size of the Container PDU.] (SRS_IpduM_02820)

[SWS_IpduM_00188] [When a Container PDU is triggered, IpduM shall invoke PduR_IpduMTransmit.] (SRS_IpduM_02820)

[SWS_IpduM_00189] [In case the IpduMContainerTxConfirmationTimeout is config-ured to a value greater than 0, as long as the corresponding transmission confirma-

tion timeout timer has not elapsed, and no transmission confirmation for that Container PDU was received, the IpduM shall wait for the TxConfirmation before invoking PduR_IpduMTransmit for the next instance of that Container PDU.⌋ (SRS_IpduM_02820)

In case IpduMTxConfirmationTimeout is omitted or configured to 0, the IpduM module does not delay any new transmission attempts.

[SWS_IpduM_00190] ⌈In case the IpduMContainerTxConfirmationTimeout is configured to a value greater than 0, and the corresponding transmission confirmation timeout timer has elapsed, or the transmission confirmation for that Container PDU was received, the IpduM shall invoke PduR_IpduMTransmit for the next oldest instance of that Container PDU in the next main function cycle at the latest.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00191] ⌈In case IpduMContainerTxTriggerMode is set to *IPDUM_DIRECT*, and PduR_IpduMTransmit returns *E_OK* for that Container PDU, IpduM shall remove that instance from the queue.⌋ (SRS_IpduM_02820)

In this case instances of a Container-PDU can be lost, if a queue inside CanIf is used since a newer instance could overwrite a previous one. Such last-is-best behavior might not be desired in this case.

[SWS_IpduM_00192] ⌈When passing a Container PDU to PduR the Parameter PduInfoPtr shall contain a pointer to the assembled Container PDU in SduDataPtr and the total length (according to SWS_IpduM_00187) in SduLength.⌋ (SRS_IpduM_02820)

7.3.2.1 Triggered Transmission and Last-is-Best semantics

[SWS_IpduM_00193] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_TRIGGERTRANSMIT*, IpduM shall keep and provide buffered data until it is fetched by a call to IpduM_TriggerTransmit.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00194] ⌈If IpduMContainerTxTriggerMode is set to *IPDUM_TRIGGERTRANSMIT*, IpduM_TriggerTransmit shall copy the oldest Conainer PDU instance in the queue. If the queue is empty/ non-existent, the current instance of the Container PDU is copied. If the current instance of the Container PDU is empty/ non-existent as well, *E_NOT_OK* is returned by IpduM_TriggerTransmit.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00220] ⌈For contained I-PDUs, with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST, IpduM shall use PduR_IpduMTriggerTransmit to fetch the PDU data from its upper layer immediately before it transfers the container I-PDU to the lower layer.⌋ (SRS_IpduM_02820)

While it seems natural to use IpduMContainedTxPduCollectionSemantics IPDUM_COLLECT_LAST_IS_BEST in combination with IpduMContainerTxTriggerMode IPDUM_TRIGGERTRANSMIT, it may also be used in combination with IPDUM_DIRECT.

As soon as a contained I-PDU is configured to use last-is-best semantics, the user accepts that not necessarily all instances/ values of this contained I-PDU are visible on the wire. On the other hand, queued collection semantics guarantees that every instance/ value of the contained I-PDU is visible on the wire.

[SWS_IpduM_00221] [IpduM shall store contained I-PDUs in the Container PDU in the same order in which they are passed to IpduM. If the Container PDU already contains an instance of a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to *IPDUM_COLLECT_LAST_IS_BEST*, IpduM shall replace the already existing instance without modifying the order of the contained I-PDUs already collected.] (SRS_IpduM_02821)

[SWS_IpduM_00222] [In case PduR_IpduMTriggerTransmit returns *E_NOT_OK* for a contained I-PDU, IpduM shall omit this contained I-PDU silently. The associated Container PDU shall be transmitted anyway without the omitted contained I-PDU. All contained I-PDUs behind the skipped one shall be moved up by the size of the omitted contained I-PDU including its header.] (SRS_IpduM_02821)

[SWS_IpduM_00201] [When adding the first contained I-PDU to a Container PDU with the parameter IpduMContainerTxFirstContainedPduTrigger set to TRUE, IpduM shall call PduR_IpduMTransmit.] (SRS_IpduM_02820)

Rationale: This way a transmission is requested for a time-triggered bus.

7.3.2.2 Queueing

In case more than one instance of a Container PDU has to be kept by IpduM, up to IpduMContainerQueueSize instances can be stored in addition to the current instance. The current instance is one instance of the Container PDU that currently contained I-PDUs are being added to. After this instance has either been queued or copied to the lower layer, i.e. after a TriggerTransmit or Transmit API call depending on the configuration of IpduMContainerTxTriggerMode, no more contained I-PDUs can be added to this instance.

[SWS_IpduM_00195] [If PduR_IpduMTransmit has returned *E_NOT_OK*, the same transmit request shall be repeated in the next main function cycle. The instance of that Container PDU is queued in the meantime.] (SRS_IpduM_02820)

See also SWS_IpduM_00199.

[SWS_IpduM_00196] [If the IpduM receives a TxConfirmation for a specific Container PDU, it shall translate this confirmation into the corresponding confirmations for those contained I-PDUs having IpduMContainedTxPduConfirmation set to *TRUE* and were contained in the last sent out instance of the Container I-PDU.
If the same contained I-PDU is present more than once, this results in multiple TxConfirmations] (SRS_IpduM_02820)

[SWS_IpduM_00198] [The IpduM shall discard unexpected Tx-Confirmations silently.] (SRS_IpduM_02820)

This may happen if a previously requested transmit request has been timed out, but is confirmed now.

[SWS_IpduM_00199] [If creating a new instance of a Container PDU would exceed IpduMContainerQueueSize the oldest instance shall be discarded. If IpduMContainerQueueSize is not configured the local instance shall be discarded.
If Development Error Detection is configured (ECUC_IpduM_00132) IPDUM_E_QUEUEOVFL shall be reported to DET.] (SRS_IpduM_02820)

[SWS_IpduM_00200] [A Container PDU instance shall be dropped from the queue if it is fetched by TriggerTransmit.] (SRS_IpduM_02820)

7.3.3 Reception

[SWS_IpduM_00202] [If IpduMContainerPduProcessing is set to *IPDUM_PROCESSING_IMMEDIATE*, the processing of the received Container PDUs shall be executed in the context of IpduM_RxIndication. Otherwise it is deferred to the next main function cycle. All deferred Container PDUs shall be processed in the order of their reception.] (SRS_IpduM_02820)

[SWS_IpduM_00203] [If by a call of IpduM_RxIndication a Container PDU is received, the contained I-PDUs shall be extracted.] (SRS_IpduM_02820)

[SWS_IpduM_00204] [For each contained I-PDU, the ID from its header shall be used to identify the corresponding I-PDU (Parameter IpduMContainedPduHeaderId).] (SRS_IpduM_02824)

[SWS_IpduM_00205] [If for the Container PDU IpduMContainerRxAcceptContainedPdu is set to *IPDUM_ACCEPT_CONFIGURED*, IpduM shall expect and match only contained I-PDUs that reference the Container PDU in IpduMContainedRxInContainerPduRef.] (SRS_IpduM_02824)

[SWS_IpduM_00206] [If for the Container PDU IpduMContainerRxAcceptContainedPdu is set to *IPDUM_ACCEPT_ALL*, IpduM shall expect and match all contained I-PDUs independent of IpduMContainedRxInContainerPduRef.] (SRS_IpduM_02824)

[SWS_IpduM_00207] [If an extracted contained I-PDU could not be matched based on its ID it shall be discarded silently.] (SRS_IpduM_02820, SRS_IpduM_02824)

[SWS_IpduM_00208] [For each contained I-PDU the length given in its header shall be used as the length of the corresponding I-PDU.] (SRS_IpduM_02820)

[SWS_IpduM_00209] [Each contained I-PDU shall be notified to PduR via PduR_IpduMRxIndication. IpduM shall indicate the contained I-PDUs in the same order as the I-PDUs are located inside the Container PDU.] (SRS_IpduM_02821)

[SWS_IpduM_00210] ⌈When processing a received Container PDU and detecting a header containing the ID 0 the processing for this Container PDU shall be stopped and the remaining bytes shall be ignored.⌋ (SRS_IpduM_02820)

Rationale: A header ID of 0 means that Container PDU has been filled with padding bytes and no further data is contained.

SWS_IpduM_00210 does not mean that always a 0 has to be expected at the end of the payload.

7.3.3.1 Queueing

[SWS_IpduM_00211] ⌈If a Container PDU is received and IpduMContainerPduProcessing is set to *IPDUM_PROCESSING_DEFERRED*, the Container PDU shall be queued.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00212] ⌈If receiving a new instance of a Container PDU would exceed IpduMContainerQueueSize the oldest instance shall be discarded. If Development Error Detection is configured (ECUC_IpduM_00132) *IPDUM_E_QUEUEOVFL* shall be reported to DET.⌋ (SRS_IpduM_02820)

7.3.3.2 Errorhandling

There are bus systems where it is not possible to set an arbitrary size for the transmitted L-PDU (e.g. CanFD). The valid payload length of a Container PDU can be derived from the contained headers. Therefore, the difference to the actual length of the Container PDU can be considered padding.

Assumption is that underlying bus modules are configured such that the padded values do not build up a valid header.

[SWS_IpduM_00213] ⌈When processing a received Container PDU and detecting a header where the payload length exceeds the remaining bytes of the container the processing for this Container PDU shall be stopped and the remaining bytes shall be ignored. If Development Error Detection is configured (ECUC_IpduM_00132) *IPDUM_E_HEADER* shall be reported to DET.⌋ (SRS_IpduM_02820)

A header with a payload length greater than the remaining byte is invalid. No further header is to be expected behind it.

[SWS_IpduM_00214] ⌈If the remaining bytes in a Container PDU are less than the configured IpduMContainerHeaderSize **(ECUC_IpduM_00183)** the remaining bytes shall be ignored.⌋ (SRS_IpduM_02820)

7.4 Error classification

The following errors and exceptions shall be detectable by the IpduM module depending on its build version (development/ production mode):

Type or error	*Relevance*	*Related error code*	*Value [hex]*
[SWS_IpduM_00026] ⌈API service called with wrong parameter⌋ (SRS_BSW_00337)	Development	IPDUM_E_PARAM	0x10
[SWS_IpduM_00162] ⌈NULL pointer checking⌋ (SRS_BSW_00337, SRS_BSW_00414)	Development	IPDUM_E_PARAM_POINTER	0x11
[SWS_IpduM_00153] ⌈API service used without module initialization⌋ (SRS_BSW_00337)	Development	IPDUM_E_UNINIT	0x20
[SWS_IpduM_00174] ⌈Invalid configuration set selection⌋ (SRS_BSW_00414)	Development	IPDUM_E_INIT_FAILED	0x21
[SWS_IpduM_00215] ⌈Erroneous header detected.⌋ (SRS_IpduM_02820)	Development	IPDUM_E_HEADER	0x30
[SWS_IpduM_00216]⌈Container Queue overflow⌋ (SRS_IpduM_02820)	Development	IPDUM_E_QUEUEOVFL	0x31

Table 1: Mapping of the AUTOSAR IpduM module's development error IDs

7.5 Error detection and notification

[SWS_IpduM_00028] ⌈If IpduMDevErrorDetect is configured to TRUE, all IpduM APIs shall check their input parameters and report detected errors to DET by IPDUM_E_PARAM for normal parameter and IPDUM_E_PARAM_POINTER for pointer parameters.⌋ (SRS_BSW_00323)

8 API specification

8.1 Imported types

This chapter lists all imported types and the corresponding header files.

[SWS_IpduM_00102] [

Module	Imported Type
ComStack_Types	PduIdType
	PduInfoType
Std_Types	Std_ReturnType
	Std_VersionInfoType

] (SRS_BSW_00357)

8.2 Type definitions

8.2.1 IpduM_ConfigType

[SWS_IpduM_00159] [

Name:	IpduM_ConfigType
Type:	Structure
Range:	Implementation specific.
Description:	This is the type of the data structure containing the initialization data for the I-PDU multiplexer.

] (SRS_BSW_00438)

8.3 Function definitions

This is a list of functions provided for upper layer modules.

8.3.1 IpduM_Init

[SWS_IpduM_00032] [

Service name:	IpduM_Init

Syntax:	`void IpduM_Init(`
	`  const IpduM_ConfigType* config`
	`)`

Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	config Implementation specific structure with configuration parameters.
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	Initializes the I-PDU Multiplexer.

] (SRS_BSW_00344, SRS_BSW_00405, SRS_BSW_00101, SRS_BSW_00369)

[SWS_IpduM_00033] ⌈The function IpduM_Init shall initialize all module-related global variables.⌋ (SRS_BSW_00101)

[SWS_IpduM_00084] ⌈The behavior of the IpduM is unspecified until a correct call to IpduM_Init is made.⌋ (SRS_BSW_00406)

8.3.2 IpduM_GetVersionInfo

[SWS_IpduM_00037] ⌈

Service name:	IpduM_GetVersionInfo	
Syntax:	`void IpduM_GetVersionInfo( Std_VersionInfoType* versioninfo )`	
Service ID[hex]:	0x01	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	versioninfo	Pointer to where to store the version information of this module.
Return value:	None	
Description:	Service returns the version information of this module.	

⌋ (SRS_BSW_00407, SRS_BSW_00369, SRS_BSW_00003)

8.3.3 IpduM_Transmit

[SWS_IpduM_00043] ⌈

Service name:	IpduM_Transmit	
Syntax:	`Std_ReturnType IpduM_Transmit( PduIdType PdumTxPduId, const PduInfoType* PduInfoPtr )`	
Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Non Reentrant for the same PDU-ID. Reentrant for different PDU-ID.	
Parameters (in):	PdumTxPduId	ID of I-PDU to be transmitted.
	PduInfoPtr	A pointer to a structure with I-PDU related data that shall be transmitted: data length and pointer to I-SDU buffer
Parameters (inout):	None	
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: Transmit request is accepted E_NOT_OK: Transmit request is not accepted
Description:	Service is called by the PDU-Router to request a transmission.	

⌋ (SRS_BSW_00369)
For a detailed description read Chapter 7.2.4.1.

8.4 Call-back notifications

8.4.1 IpduM_RxIndication

[SWS_IpduM_00040] ⌈

Service name:	IpduM_RxIndication	

Syntax:	`void IpduM_RxIndication(`
	`  PduIdType RxPduId,`
	`  const PduInfoType* PduInfoPtr`
	`)`	

Service ID[hex]:	0x42	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	RxPduId	ID of the received I-PDU.
	PduInfoPtr	Contains the length (SduLength) of the received I-PDU and a pointer to a buffer (SduDataPtr) containing the I-PDU.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Indication of a received I-PDU from a lower layer communication interface module.	

⌋ (SRS_BSW_00369, SRS_IpduM_02817)

[SWS_IpduM_00041] ⌈If there is a static part configured in a multiplexed SDU received from the PDU Router, the function IpduM_RxIndication transforms the incoming I-PDU ID into the correct I-PDU ID for the static part's destination and then forwards the SDU via the PDU Router, see PduR_IpduMRxIndication in the PDU Router SWS.⌋ (SRS_IpduM_02812)

[SWS_IpduM_00042] ⌈When a multiplexed I-PDU is received from the PDU Router the function IpduM_RxIndication uses the incoming I-PDU ID and the selector field to find out the correct I-PDU ID for the dynamic part's destination and then forwards the I-PDU via the PDU Router, see PduR_IpduMRxIndication in the PDU Router SWS.⌋ (SRS_IpduM_02812)

[SWS_IpduM_00217] ⌈When a Container PDU is received from the PDU Router, the function IpduM_RxIndication forwards the contained I-PDUs via the PDU Router, using PduR_IpduMRxIndication (see SWS_IpduM_00105) .⌋ (SRS_IpduM_02820)

[SWS_IpduM_00086] ⌈The function IpduM_RxIndication shall be callable in interrupt context, e.g. from receive interrupt. ⌋ (SRS_IpduM_02812)

8.4.2 IpduM_TxConfirmation

[SWS_IpduM_00044] ⌈

Service name:	IpduM_TxConfirmation

Syntax:	`void IpduM_TxConfirmation(`
	`  PduIdType TxPduId`
	`)`

Service ID[hex]:	0x40
Sync/Async:	Synchronous

Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	ID of the I-PDU that has been transmitted.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	The lower layer communication interface module confirms the transmission of an I-PDU.	

] (SRS_BSW_00369)

[SWS_IpduM_00088] [The function IpduM_TxConfirmation shall translate the confirmation received from the PDU Router into confirmations for the I-PDUs which where contained in the sent multiplexed I-PDU or Container PDU.] (SRS_IpduM_02814)

These confirmations are given again to the PDU Router that has to route them to COM.

[SWS_IpduM_00087] [The function IpduM_TxConfirmation shall be callable in interrupt context, e.g. from a transmit interrupt.] (SRS_IpduM_02814)

8.4.3 IpduM_TriggerTransmit

[SWS_IpduM_00060] [

Service name:	IpduM_TriggerTransmit	
Syntax:	`Std_ReturnType IpduM_TriggerTransmit( PduIdType TxPduId, PduInfoType* PduInfoPtr )`	
Service ID[hex]:	0x41	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant for different PduIds. Non reentrant for the same PduId.	
Parameters (in):	TxPduId	ID of the SDU that is requested to be transmitted.
Parameters (inout):	PduInfoPtr	Contains a pointer to a buffer (SduDataPtr) to where the SDU data shall be copied, and the available buffer size in SduLengh. On return, the service will indicate the length of the copied SDU data in SduLength.
Parameters (out):	None	
Return value:	Std_ReturnType	E_OK: SDU has been copied and SduLength indicates the number of copied bytes. E_NOT_OK: No SDU data has been copied. PduInfoPtr must not be used since it may contain a NULL pointer or point to invalid data.
Description:	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.	

] (SRS_BSW_00369)

[SWS_IpduM_00090] [Within the function IpduM_TriggerTransmit, the IpduM shall copy the contents of its I-PDU transmit buffer to the PDU buffer given by PduInfoPtr->SduDataPtr and update PduInfoPtr->SduLength with length of the copied data accordingly.] (SRS_IpduM_02810)

[SWS_IpduM_00091] ⌈The IpduM shall take care about the data consistency during providing the data. ⌋ (SRS_IpduM_02810)

Use case: This function is used e.g. by the LIN Master for sending out a LIN frame. In this case, the trigger transmit can be initiated by the Master schedule table itself or a received LIN header.
This function is also used by the FlexRay Interface for requesting PDUs to be sent in static part (synchronous to the FlexRay global time).

[SWS_IpduM_00089] ⌈The function IpduM_TriggerTransmit shall be callable in interrupt context. ⌋ (SRS_IpduM_02810)

8.5 Scheduled functions

Most of the functions of the IpduM module are called synchronous in the context of the upper layer (for transmission) and in the context of the lower layer (for reception). However, for the TxConfirmation timeout timer a scheduled function is needed.

[SWS_IpduM_00103] ⌈

Service name:	IpduM_MainFunction

Syntax:	`void IpduM_MainFunction(`
	`  void`
	`)`

Service ID[hex]:	0x10
Description:	Performs the processes of the activities that are not directly initiated by the calls from PDU-R.

⌋ (SRS_BSW_00425)

[SWS_IpduM_00101] ⌈The function IpduM_MainFunction shall perform the processing of the IpduM activities that are not directly initiated by the calls from PDU Router. This includes at least the TxConfirmation time observation.⌋ (SRS_IpduM_02813)

8.6 Expected Interfaces

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

8.6.1 Mandatory Interfaces

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

[SWS_IpduM_00104] ⌈

API function	Description

⌋(SRS_BSW_00009)

Actually, the IpduM module needs no APIs of other modules compulsorily, since the IpduM module could be used only for reception or transmission of multiplexed I-PDUs. In such a case the not used reception or transmission APIs of the PduR are optional. Hence, depending on the use-case all used APIs are optional.

8.6.2 Optional Interfaces

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

[SWS_IpduM_00105] ⌈

API function	Description
Det_ReportError	Service to report development errors.
PduR_IpduMRxIndication	Indication of a received I-PDU from a lower layer communication interface module.
PduR_IpduMTransmit	Requests transmission of an I-PDU.
PduR_IpduMTriggerTransmit	Within this API, the upper layer module (called module) shall check whether the available data fits into the buffer size reported by PduInfoPtr->SduLength. If it fits, it shall copy its data into the buffer provided by PduInfoPtr->SduDataPtr and update the length of the actual copied data in PduInfoPtr->SduLength. If not, it returns E_NOT_OK without changing PduInfoPtr.
PduR_IpduMTxConfirmation	The lower layer communication interface module confirms the transmission of an I-PDU.

⌋(SRS_BSW_00009)

8.6.3 Configurable interfaces

Not applicable

9 Sequence diagrams

9.1 Transmission of a multiplexed I-PDU and Transmit confirmation

The following sequence chart shows a transmit request initiated by the COM layer. The transmit request is for an I-PDU which has to be transmitted within a multiplexed I-PDU. In the IpduM module is configured that this transmitted I-PDU triggers the sending of the multiplexed I-PDU.

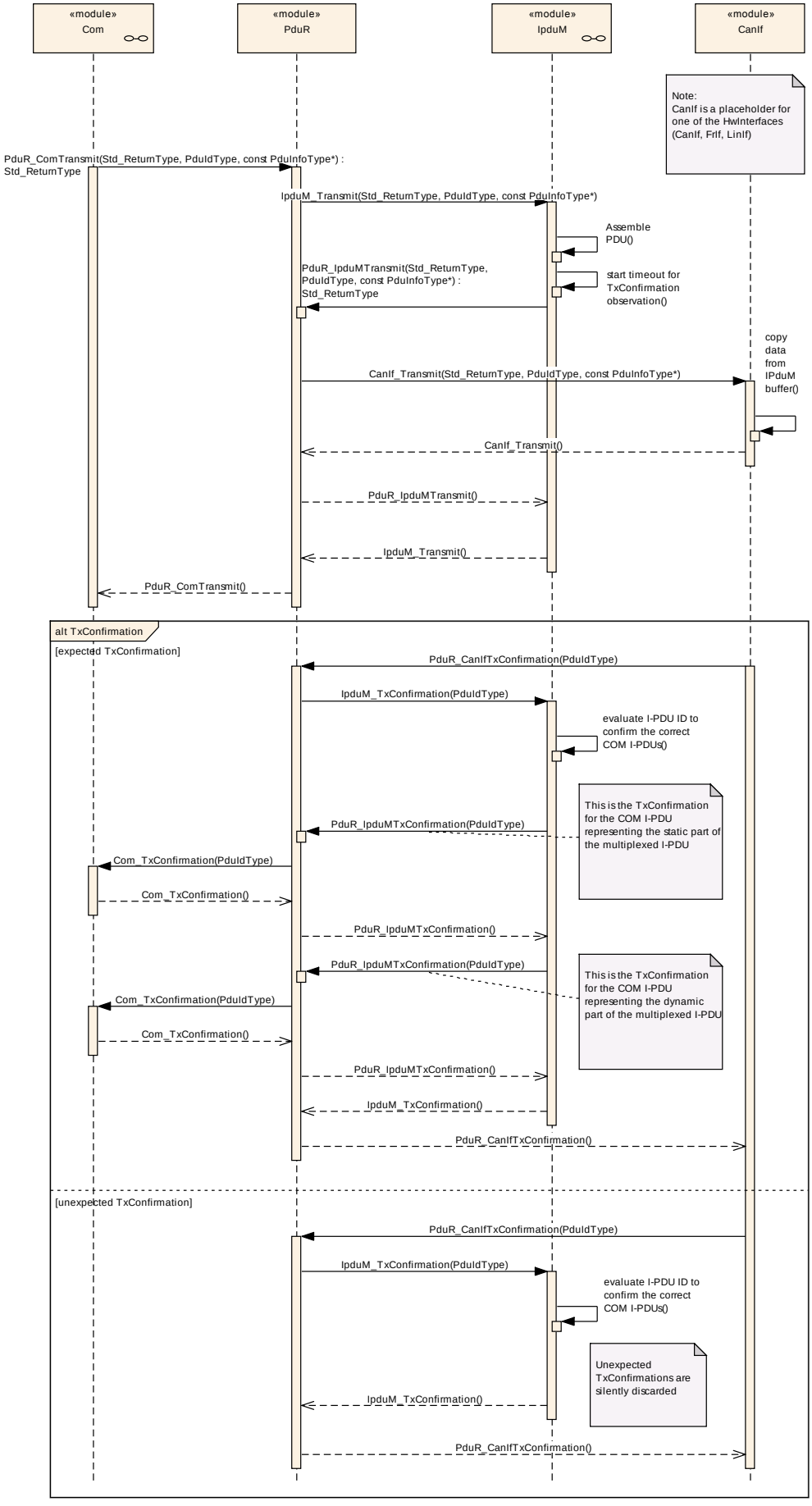
«module»
Com
«module»
PduR
«module»
IpduM
«module»
CanIf
Note:
CanIf is a placeholder for one of the HwInterfaces (CanIf, FrIf, LinIf)
PduR_ComTransmit(Std_ReturnType, PduIdType, const PduInfoType*) : Std_ReturnType
IpduM_Transmit(Std_ReturnType, PduIdType, const PduInfoType*)
Assemble PDU()
PduR_IpduMTransmit(Std_ReturnType, PduIdType, const PduInfoType*) : Std_ReturnType
start timeout for TxConfirmation observation()
copy data from IPduM buffer()
CanIf_Transmit(Std_ReturnType, PduIdType, const PduInfoType*)
CanIf_Transmit()
PduR_IpduMTransmit()
IpduM_Transmit()
PduR_ComTransmit()
alt TxConfirmation
[expected TxConfirmation]
PduR_CanIfTxConfirmation(PduIdType)
IpduM_TxConfirmation(PduIdType)
evaluate I-PDU ID to confirm the correct COM I-PDUs()
This is the TxConfirmation for the COM I-PDU representing the static part of the multiplexed I-PDU
PduR_IpduMTxConfirmation(PduIdType)
Com_TxConfirmation(PduIdType)
Com_TxConfirmation()
PduR_IpduMTxConfirmation()
PduR_IpduMTxConfirmation(PduIdType)
This is the TxConfirmation for the COM I-PDU representing the dynamic part of the multiplexed I-PDU
Com_TxConfirmation(PduIdType)
Com_TxConfirmation()
PduR_IpduMTxConfirmation()
IpduM_TxConfirmation()
PduR_CanIfTxConfirmation()
[unexpected TxConfirmation]
PduR_CanIfTxConfirmation(PduIdType)
IpduM_TxConfirmation(PduIdType)
evaluate I-PDU ID to confirm the correct COM I-PDUs()
Unexpected TxConfirmations are silently discarded
IpduM_TxConfirmation()
PduR_CanIfTxConfirmation()

Figure 6 Transmission and confirmation of multiplexed I-PDU with triggering

9.2 Transmission of a multiplexed I-PDU without Trigger

The following sequence chart shows a transmit request initiated by the COM layer. Because of the configuration of the IpduM, no transmit request for the IpduM I-PDU takes place. For configuration see ECUC_IpudM_00052.

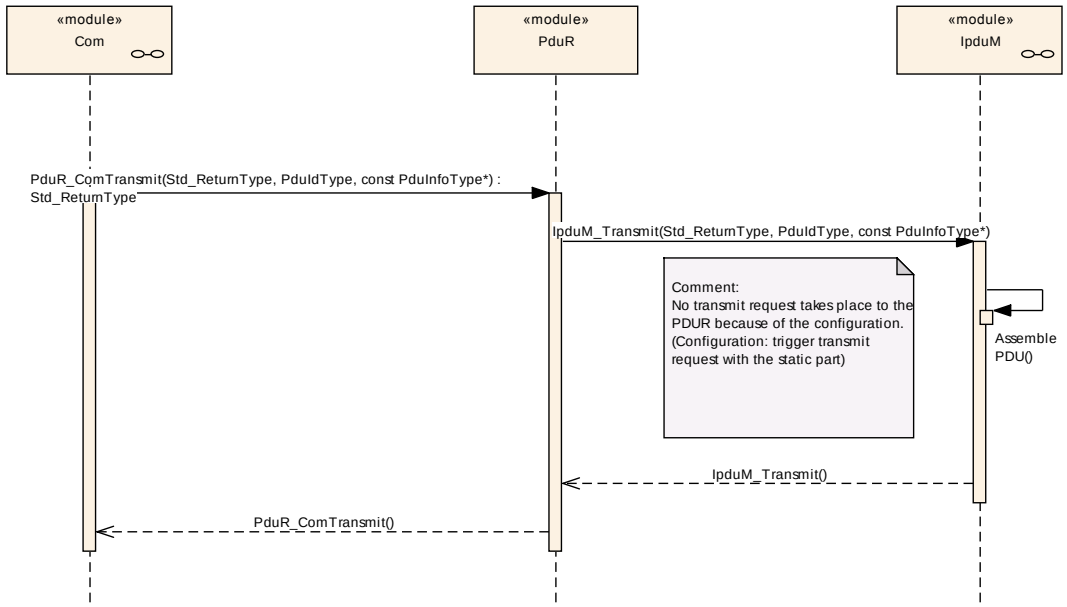

Figure 7 Transmission of a multiplexed I-PDU without triggering

9.3 Reception of the multiplexed I-PDU

The following sequence chart shows a reception of a multiplexed I-PDU. The I-PDU contains a static and a dynamic part and both are configured to create an RxIndication to the PDU Router module.

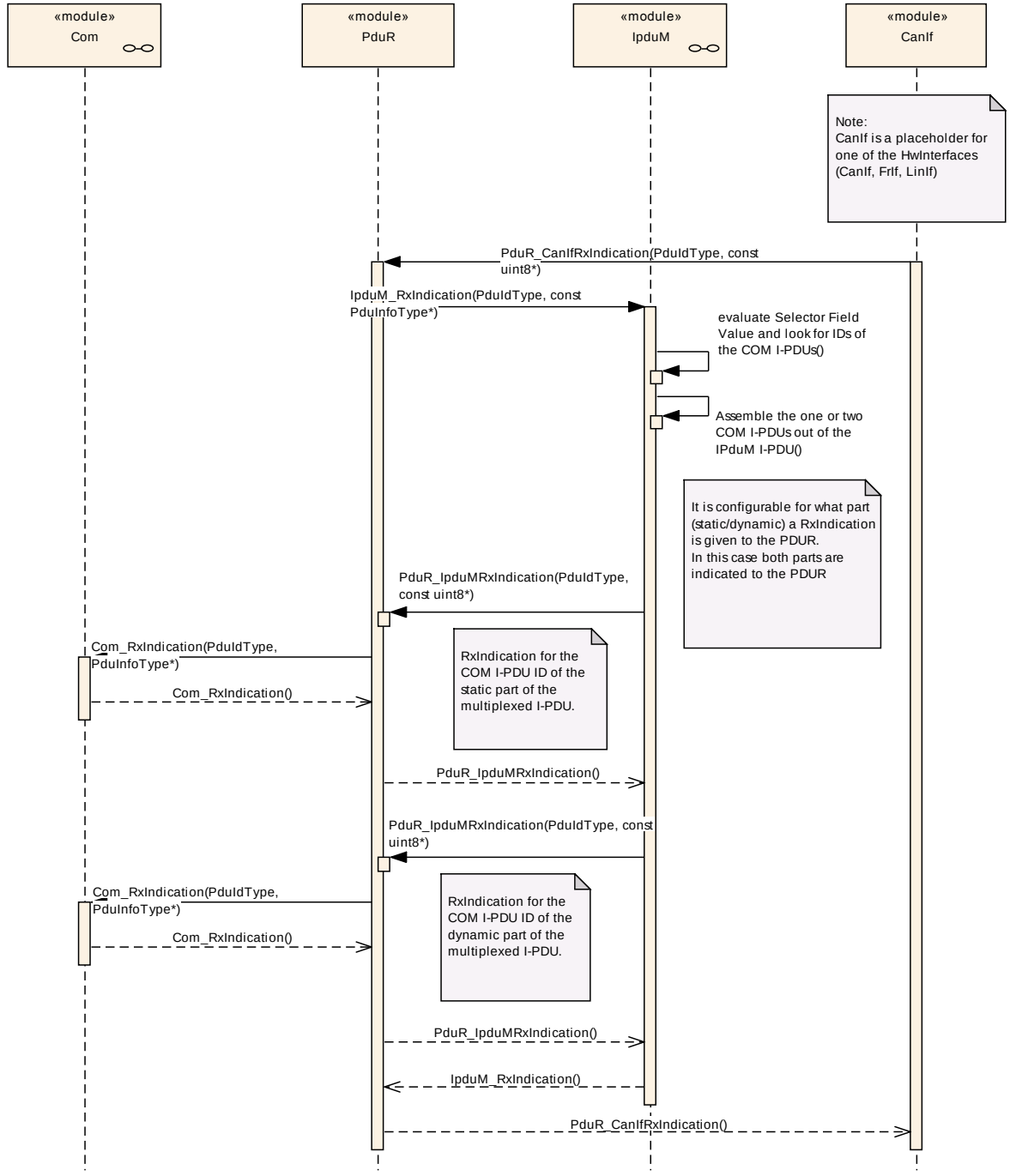

Figure 8 Reception of a multiplexed I-PDU

9.4 Trigger Transmit

The following sequence chart shows a Trigger Transmit request from an interface layer.

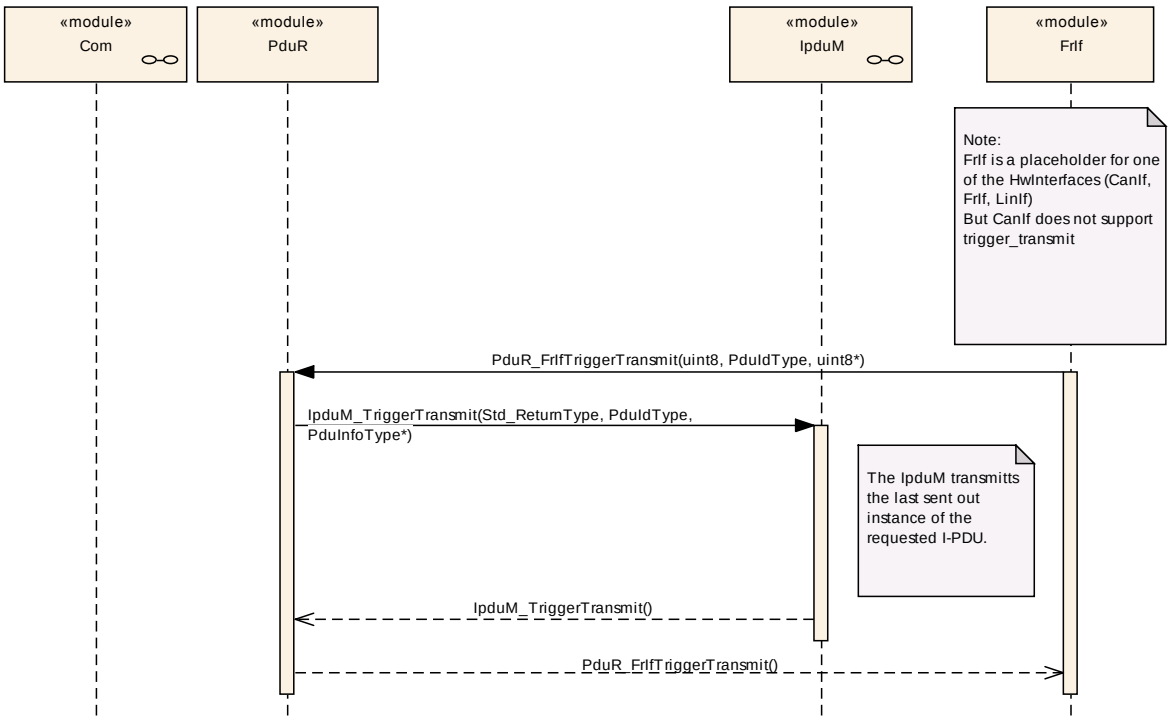

Figure 9 Trigger Transmit request from interface layer

9.5 Missing Transmit Confirmation

The following sequence chart shows the case that a TxConfirmation is not received by the IpduM module during the TX Confirmation timeout. After the timeout has elapsed, it is allowed to send the I-PDU again.

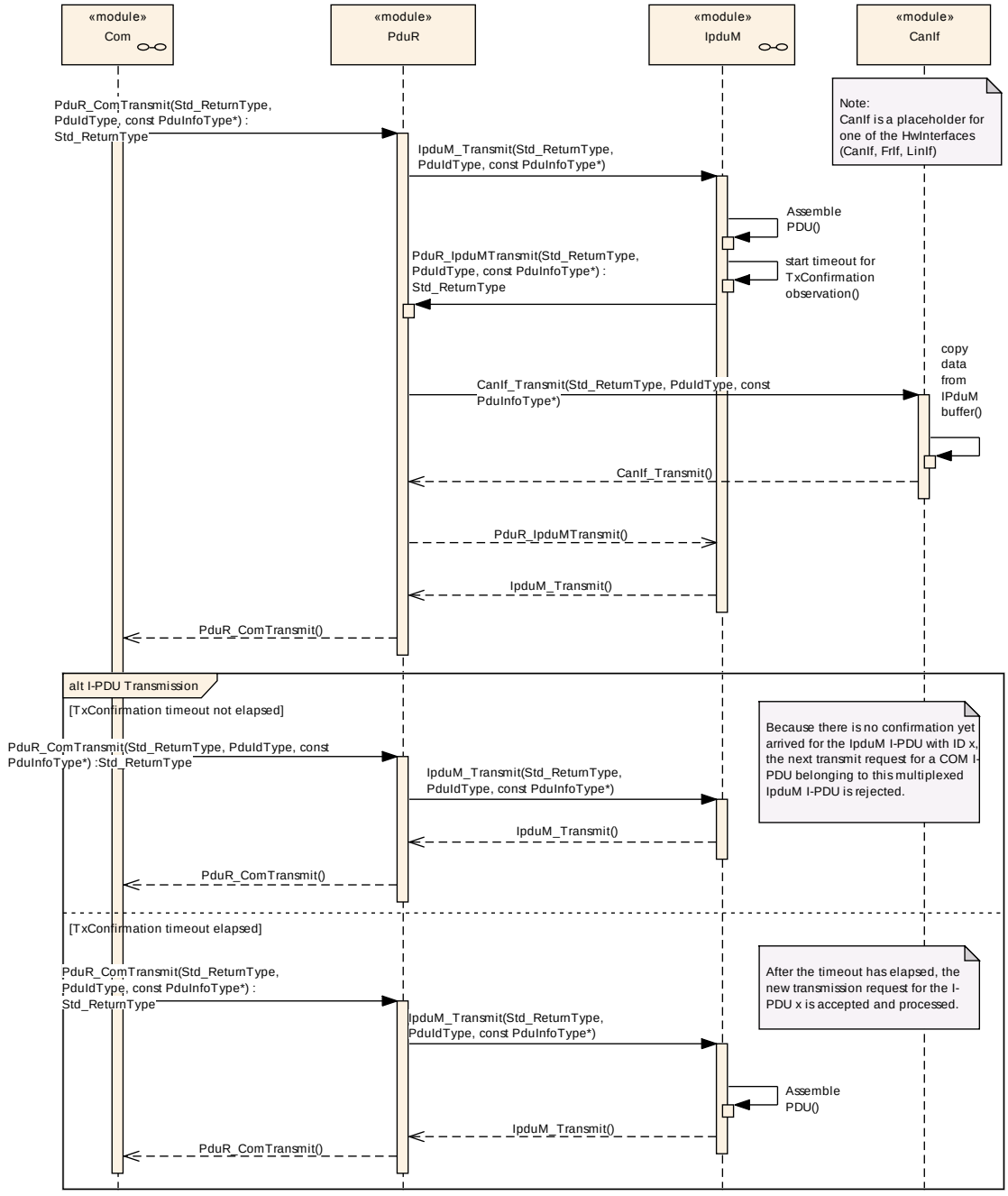

Figure 10 Missing Transmit Confirmation

10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers.

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

Chapter 10.3 specifies published information of the module IpduM.

10.1 How to read this chapter

For details, refer to the chapter 10.1 Introduction to configuration specification in SWS_BSWGeneral.

10.2 Containers and configuration parameters

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

10.2.1 Variants

There are three variants called: VARIANT-PRE-COMPILE, VARIANT-LINK-TIME and VARIANT-POST-BUILD.

The VARIANT-PRE_COMPILE is designed for modules that are purely configured at pre-compile time. In this variant, all configuration parameters are fixed at compile-time.

The VARIANT-LINK-TIME is designed for the use case where parameters that affect code generation are fixed at compile-time and all other configuration parameters are fixed at link-time.

The VARIANT-POST-BUILD is designed for parameters that affect code generation to be fixed at compile-time and all other parameters to be fixed at post build-time.

10.2.2 Configuration overview

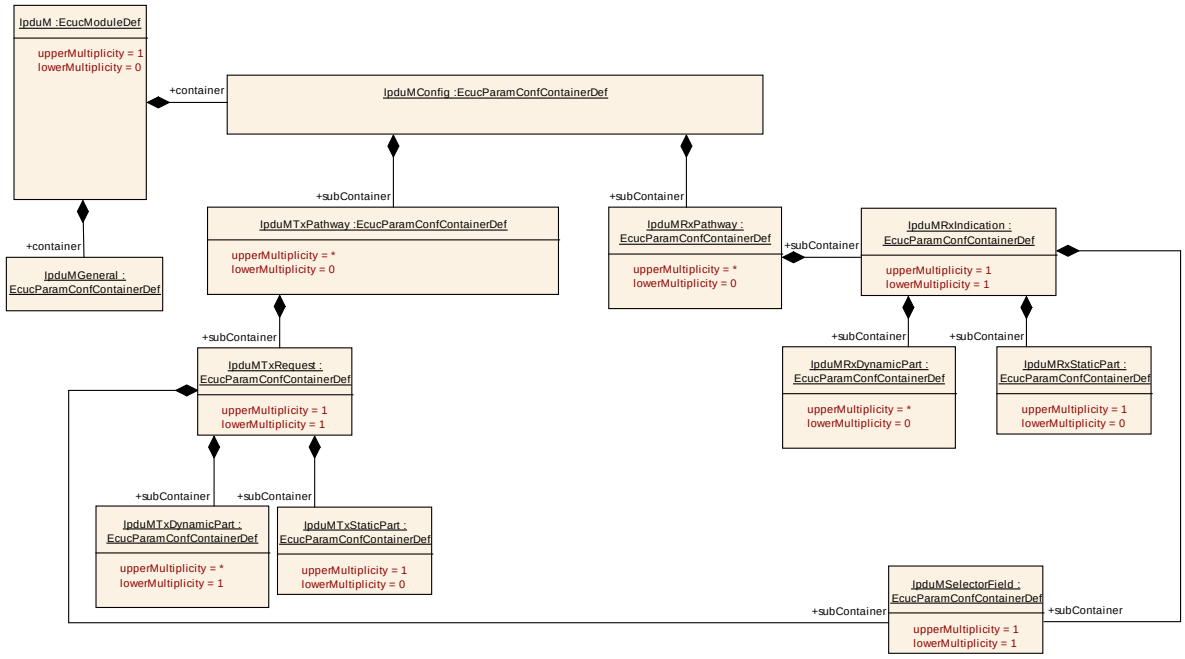

Figure 11 IpduM Configuration Overview

10.2.3 IpduM

Module Name	**IpduM**
Module Description	Configuration of the IpduM (Ipdu Multiplexer) module.
Post-Build Variant Support	true

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMConfig	1	This container contains the sub containers of the IpduM module. • The IpduMTxPathway subcontainer includes information about sent I-PDUs. • The IpduMRxPathway includes information about received I-PDUs. • The IpduMTxContainerPdu and IpduMTxContainedPdu include information about the sending of ContainerPdus. • The IpduMRxContainerPdu and IpduMRxContainedPdu include information about the reception of ContainerPdus.
IpduMGeneral	1	Contains the general configuration parameters of IpduM.
IpduMPublishedInformation	1	Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.

10.2.4 IpduMConfig

SWS Item	**ECUC_IpduM_00059 :**
Container Name	IpduMConfig
Description	This container contains the sub containers of the IpduM module. • The IpduMTxPathway subcontainer includes information about sent I-PDUs. • The IpduMRxPathway includes information about received I-PDUs. • The IpduMTxContainerPdu and IpduMTxContainedPdu include information about the sending of ContainerPdus. • The IpduMRxContainerPdu and IpduMRxContainedPdu include information about the reception of ContainerPdus.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00166 :**		
Name	IpduMMaxTxBufferSize		
Description	Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00165 :**		
Name	IpduMMaxTxPathwayCnt		
Description	Maximum number of transmitted IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-

			BUILD
	Post-build time	--	
Scope / Dependency	scope: local		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMContainedRxPdu	0..*	Configuration of a received contained Pdu.
IpduMContainedTxPdu	0..*	Configuration of a sender ContainedPdu.
IpduMContainerRxPdu	0..*	Configuration of a receiver ContainerPdu which may collect several ContainedPdus.
IpduMContainerTxPdu	0..*	Configuration of a transmitted container Pdu.
IpduMRxPathway	0..*	includes information about received I-PDUs
IpduMTxPathway	0..*	includes information about sent I-PDUs

10.2.5 IpduMGeneral

SWS Item	**ECUC_IpduM_00130 :**
Container Name	IpduMGeneral
Description	Contains the general configuration parameters of IpduM.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00131 :**		
Name	IpduMConfigurationTimeBase		
Description	The cycle time with which IpduM_MainFunction should be invoked (in seconds).		
Multiplicity	1		
Type	EcucFloatParamDef		
Range	0 .. 3600		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00132 :**		
Name	IpduMDevErrorDetect		
Description	Switches the Default Error Tracer (Det) detection and notification ON or OFF. • true: enabled (ON). • false: disabled (OFF).		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00197 :**		
Name	IpduMHeaderByteOrder		
Description	This parameter defines the ByteOrder of the headers inside a Container I-PDU.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	IPDUM_BIG_ENDIAN	Headers inside a Container I-PDU shall be ordered big endian.	
	IPDUM_LITTLE_ENDIAN	Headers inside a Container I-PDU shall be ordered little endian.	
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Depen-dency	scope: local		

SWS Item	**ECUC_IpduM_00133 :**		
Name	IpduMStaticPartExists		
Description	This is to allow optimizations in the case the IpduM will never be used with a static part. Note that this is a pre-compile option. If this is set to False then it will not be possible to add static parts after compilation. True: A static part may exist. False: A static part will never exist.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00134 :**		
Name	IpduMVersionInfoApi		
Description	Active/Deactivate the version information API. true: version information activated false: version information deactivated		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

No Included Containers

10.2.6 IpduMTxPathway

SWS Item	**ECUC_IpduM_00070 :**		
Container Name	IpduMTxPathway		
Description	Contains the configuration parameters transmitted I-PDUs by the IpduM module.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMTxRequest	1	configuration for a TxRequest

10.2.7 IpduMTxRequest

SWS Item	**ECUC_IpduM_00052 :**
Container Name	IpduMTxRequest
Description	This container is used to specify the configuration for Transmit requests. There will be one instance of this container for each I-PDU that can be requested for transmission (the outgoing I-PDUs) by the IpduM.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00162 :**		
Name	IpduMByteOrder		
Description	This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN		--
	LITTLE_ENDIAN		--
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00121 :**
Name	IpduMIPduUnusedAreasDefault
Description	IpduM module fills not used areas of an I-PDU with this bit-pattern If this attribute is omitted the IpduM module does not fill the I-PDU.
Multiplicity	0..1

Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00158 :**		
Name	IpduMTxConfirmationPduId		
Description	Handle Id used by the PduR for confirmation (IpduM_TxConfirmation) and for TriggerTransmit (IpduM_TriggerTransmit). The existence of this parameter is essential for the PduR generation tool to actually find a symbolicNameValue for the OutgoingPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00124 :**		
Name	IpduMTxConfirmationTimeout		
Description	This timeout (in seconds) defines the timeout period for monitoring the reception of the TxConfirmation. It is not used when an I-PDU is requested using the trigger transmit API.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. 3600		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00125 :**
Name	IpduMTxTriggerMode

Description	Selects whether to send the multiplexed I-PDU immediately or at some later date.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DYNAMIC_PART_TRIGGER		Writing the I-PDU representing the dynamic part does trigger a sending of the I-PDU.
	NONE		Only the buffer in the IpduM are written but not send is triggered, used for IpduM I-PDUs which are requested by TriggerTransmit.
	STATIC_OR_DYNAMIC_PART_TRIGGER		Writing the I-PDU representing the static or the dynamic part does trigger a sending of the I-PDU.
	STATIC_PART_TRIGGER		Writing the I-PDU representing the static part does trigger a sending of the I-PDU.
Post-Build Variant Value	true		
Value Configura-tion Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Depen-dency	scope: local		

SWS Item	**ECUC_IpduM_00157 :**		
Name	IpduMInitialDynamicPart		
Description	Reference to the dynamic part that shall be used to initialize this multiple-xed TX-I-PDU.		
Multiplicity	1		
Type	Reference to [IpduMTxDynamicPart]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00120 :**		
Name	IpduMOutgoingPduRef		
Description	Reference to the PDU defining the outgoing I-PDU. When the outgoing I-PDU is sent this is the I-PDU ID to give it. It is the IpduM I-PDU ID of the assembled I-PDU.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMSelectorField	1	Specifies the position of the selector field in the outgoing I-PDU.
IpduMTxDynamicPart	1..*	This (These) included container(s) must exist for each unique selector field value for this outgoing IpduM I-PDU.
IpduMTxDynamicSegment	1..*	The dynamic part of the multiplexed outgoing I-Pdu (referenced

		by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied out of the source I-Pdu that is referenced in the IpduMTxDynamicPart container and will be copied to the same location in the multiplexed outgoing I-Pdu. The segment layout for all dynamic Parts is always identical.
IpduMTxStaticPart	0..1	This included container configures the static part, if present.
IpduMTxStaticSegment	0..*	The static part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each segment in the source I-Pdu that is referenced in the IpduMTxStaticPart container will be copied to the same location in the multiplexed outgoing I-Pdu.

10.2.8 IpduMTxDynamicPart

SWS Item	**ECUC_IpduM_00056 :**		
Container Name	IpduMTxDynamicPart		
Description	Configuration parameters for an instance of a TxRequest call into the IpduM. When a Tx Request with the IpduMTxDynamicHandleId is received by the IpduM, all segments (defined in the IpduMDynamicSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used by the dynamic part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the dynamic part.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00167 :**		
Name	IpduMJitUpdate		
Description	If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00163 :**		
Name	IpduMTxDynamicConfirmation		
Description	A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the dynamic part is generated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00127 :**		
Name	IpduMTxDynamicHandleId		
Description	This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured dynamic segments are copied and the IpduMTxTriggerMode is honored.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00126 :**		
Name	IpduMTxDynamicPduRef		
Description	Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.9 IpduMTxDynamicSegment

SWS Item	**ECUC_IpduM_00168 :**
Container Name	IpduMTxDynamicSegment
Description	The dynamic part of the multiplexed outgoing I-Pdu (referenced by

	IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied out of the source I-Pdu that is referenced in the IpduMTxDynamicPart container and will be copied to the same location in the multiplexed outgoing I-Pdu. The segment layout for all dynamic Parts is always identical.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.10 IpduMTxStaticPart

SWS Item	**ECUC_IpduM_00082 :**
Container Name	IpduMTxStaticPart
Description	Configuration parameters for an instance of a Tx_Request call into the IpduM. When a Tx Request with the IpduMTxStaticHandleId is received by the IpduM, all segments (defined in the IpduMStaticSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used for the static part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the static part if it exists.

Configuration Parameters

SWS Item	**ECUC_IpduM_00167 :**		
Name	IpduMJitUpdate		
Description	If configured to true fetch the data of this part Just-In-Time via the trigger-Transmit API of the PduR.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00164 :**		
Name	IpduMTxStaticConfirmation		
Description	A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the static part is generated.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00129 :**		
Name	IpduMTxStaticHandleId		
Description	This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured static segments are copied and the IpduMTxTriggerMode is honored.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00128 :**		
Name	IpduMTxStaticPduRef		
Description	Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE

	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.11 IpduMTxStaticSegment

SWS Item	**ECUC_IpduM_00171 :**		
Container Name	IpduMTxStaticSegment		
Description	The static part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each segment in the source I-Pdu that is referenced in the IpduMTxStaticPart container will be copied to the same location in the multiplexed outgoing I-Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.12 IpduMRxPathway

SWS Item	**ECUC_IpduM_00071 :**		
Container Name	IpduMRxPathway		
Description	Contains the configuration parameters received I-PDUs by the IpduM module.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMRxIndication	1	configuration for RxIndication

10.2.13 IpduMRxIndication

SWS Item	**ECUC_IpduM_00047 :**
Container Name	IpduMRxIndication
Description	Contains the configuration for incoming RxIndication calls.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00162 :**		
Name	IpduMByteOrder		
Description	This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	BIG_ENDIAN	--	
	LITTLE_ENDIAN	--	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00109 :**	
Name	IpduMRxHandleId	
Description	This is the I-PDU ID of the incoming I-PDU. If an incoming RxIndication's I-PDU ID matches this value then it is unpacked according to the specification in this container.	
Multiplicity	1	
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)	
Range	0 .. 65535	

Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00108 :**		
Name	IpduMRxIndicationPduRef		
Description	Reference to the received Pdu representation in the ECU Configuration Description exchange file.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMRxDynamicPart	0..*	Each of these containers contains the configuration for one value of the selector field for the incoming I-PDU's dynamic part.
IpduMRxDynamicSegment	0..*	The dynamic part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxDynamicPart container and will be copied from the same location in the multiplexed incoming I-Pdu. The segment layout for all dynamic Parts is always identical.
IpduMRxStaticPart	0..1	This contains the configuration for the incoming I-PDU's static part. If the incoming I-PDU has no static part then this is omitted.
IpduMRxStaticSegment	0..*	The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.
IpduMSelectorField	1	This contains the location of the selector field. At run-time, the selector field is used to select which dynamic part is unpacked.

10.2.14 IpduMRxDynamicPart

SWS Item	**ECUC_IpduM_00048 :**		
Container Name	IpduMRxDynamicPart		
Description	This container contains the configuration for the dynamic part of incoming RxIndication calls. When an incoming received I-PDU's selector field matches the IpduMRxSelectorValue, the new outgoing I-PDU for the dynamic part is constructed as defined by the segments (defined in the IpduMDynamicSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingDynamicPduRef. In case no dynamic part shall be extracted from this received I-PDU this container does not exist. This use-case can occur in case a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00113 :**		
Name	IpduMRxSelectorValue		
Description	This is the selector value that this container refers to.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00112 :**		
Name	IpduMOutgoingDynamicPduRef		
Description	When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent PDU representation in the ECU Configuration Description exchange file.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.15 IpduMRxDynamicSegment

SWS Item	**ECUC_IpduM_00170 :**
Container Name	IpduMRxDynamicSegment

Description	The dynamic part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxDynamicPart container and will be copied from the same location in the multiplexed incoming I-Pdu. The segment layout for all dynamic Parts is always identical.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**		
Name	IpduMSegmentLength		
Description	Length of the segment in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 2032		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00159 :**		
Name	IpduMSegmentPosition		
Description	Segments bit position in the multiplexed Pdu.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.16 IpduMRxStaticPart

SWS Item	**ECUC_IpduM_00049 :**
Container Name	IpduMRxStaticPart
Description	This container contains the configuration for the static part of incoming RxIndication calls. On reception, the new outgoing I-PDU for the static part is constructed as defined by the segments (defined in the IpduMStaticSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingStaticPduRef.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00115 :**		
Name	IpduMOutgoingStaticPduRef		
Description	When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent Pdu representation in the ECU Configuration Description exchange file.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.17 IpduMRxStaticSegment

SWS Item	**ECUC_IpduM_00169 :**		
Container Name	IpduMRxStaticSegment		
Description	The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00114 :**	
Name	IpduMSegmentLength	
Description	Length of the segment in bits.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	1 .. 2032	
Default value	--	
Post-Build Variant Value	true	
Value Configuration Class	Pre-compile time	X VARIANT-PRE-COMPILE
	Link time	X VARIANT-LINK-TIME
	Post-build time	X VARIANT-POST-BUILD
Scope / Dependency	scope: local	

SWS Item	**ECUC_IpduM_00159 :**	
Name	IpduMSegmentPosition	
Description	Segments bit position in the multiplexed Pdu.	
Multiplicity	1	
Type	EcucIntegerParamDef	
Range	0 .. 2031	

Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

10.2.18 IpduMSelectorField

SWS Item	**ECUC_IpduM_00054 :**
Container Name	IpduMSelectorField
Description	This contains the location and the length of the selector field.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00160 :**		
Name	IpduMSelectorFieldLength		
Description	Length of the selector field in bits.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 16		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00161 :**		
Name	IpduMSelectorFieldPosition		
Description	Selector field bit position in the multiplexed Pdu. Range: 0..63 for CAN/ LIN I-PDUs, 0..511 for CAN FD I-PDUs, 0..2031 for FlexRay I-PDUs.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 2031		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

No Included Containers

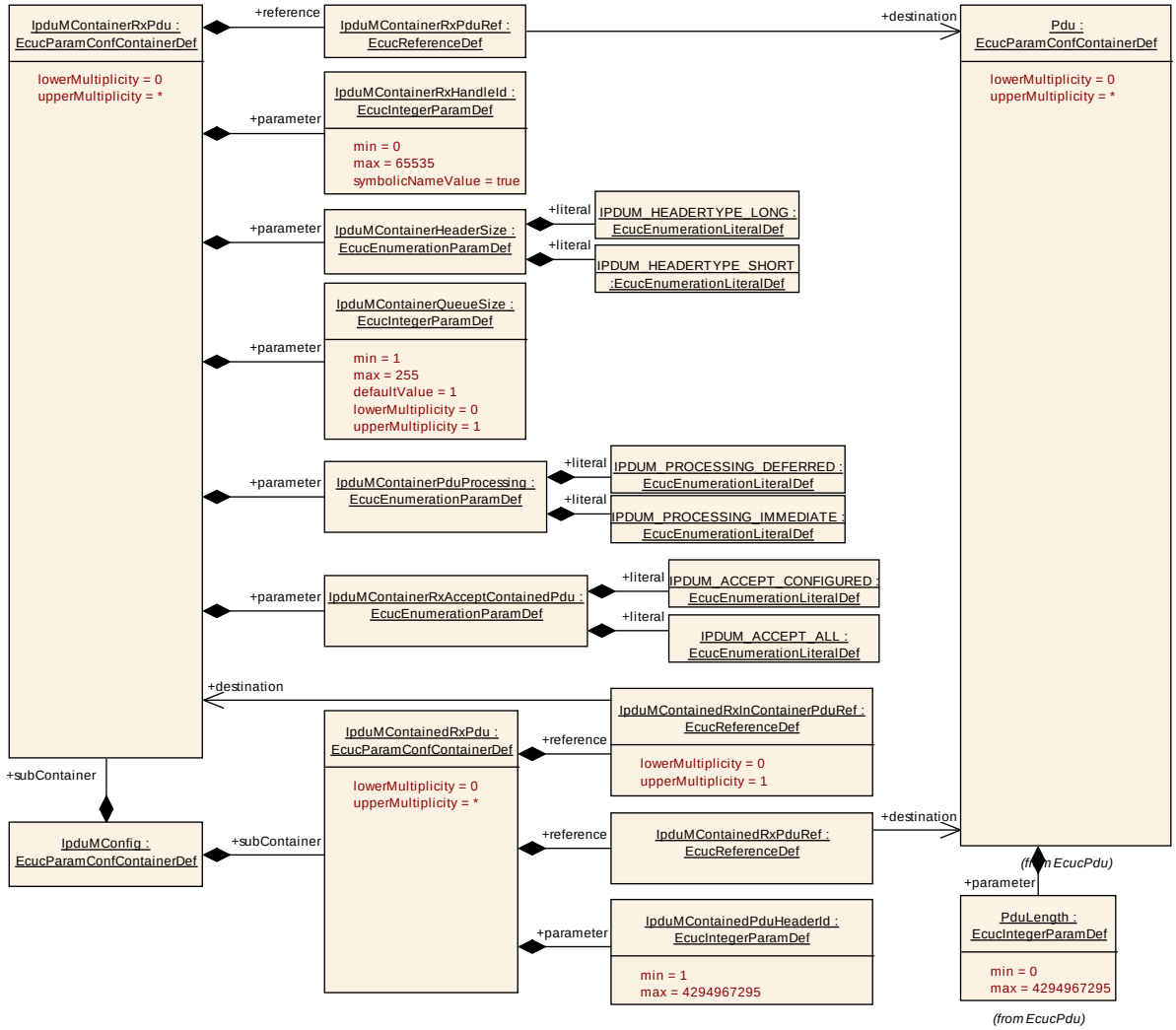

Figure 12 Configuration Overview RxContainer

10.2.19 IpduMContainerRxPdu

SWS Item	**ECUC_IpduM_00188 :**		
Container Name	IpduMContainerRxPdu		
Description	Configuration of a receiver ContainerPdu which may collect several ContainedPdus.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00183 :**	
Name	IpduMContainerHeaderSize	
Description	Defines the layout of the header information (header id and length).	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	IPDUM_HEADERTYPE_LONG	Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit

	IPDUM_HEADERTYPE_SHORT		Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00184 :**		
Name	IpduMContainerPduProcessing		
Description	Defines whether the handling of this ContainerPdu shall be done in the context of the caller (IMMEDIATE) or in the next IpduM main function (DEFERRED).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_PROCESSING_DEFERRED	--	
	IPDUM_PROCESSING_IMMEDIATE	--	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00185 :**		
Name	IpduMContainerQueueSize		
Description	Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default value	1		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00186 :**	
Name	IpduMContainerRxAcceptContainedPdu	
Description	Defines for the received IpduMRxContainerPdu whether the list of referencing IpduMRxContainedPdus (via the reference IpduMRxContainedPduContainerRef) is a closed set.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	IPDUM_ACCEPT_ALL	The IpduMRxContainedPdus which are referencing this IpduMRxContainerPdu are expected inside this IpduMRxContainerPdu, but there may also occur

			other Pdus inside this IpduMRxContainerPdu as well. This also supports the case where no IpduMRxContainedPdu references the IpduMRxContainerPdu.
	IPDUM_ACCEPT_CONFIGURED		Only the IpduMRxContainedPdus which are referencing this IpduMRxContainerPdu are expected inside this IpduMRxContainerPdu.
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00187 :**		
Name	IpduMContainerRxHandleId		
Description	Handle Id used by the PduR for RxIndication.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00189 :**		
Name	IpduMContainerRxPduRef		
Description	Reference to the Pdu which represents the container and is used for reception.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.20 IpduMContainedRxPdu

SWS Item	**ECUC_IpduM_00174 :**		
Container Name	IpduMContainedRxPdu		
Description	Configuration of a received contained Pdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00172 :**		
Name	IpduMContainedPduHeaderId		
Description	Header Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00173 :**		
Name	IpduMContainedRxInContainerPduRef		
Description	Optional reference to a container Pdu this contained Pdu may be transported in. The reference may be omitted in case IpduMContainerRxAcceptContainedPdu=IPDUM_ACCEPT_ALL.		
Multiplicity	0..1		
Type	Reference to [IpduMContainerRxPdu]		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00175 :**		
Name	IpduMContainedRxPduRef		
Description	Reference to the Pdu which represents this ContainedPdu and is used for reception indication.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

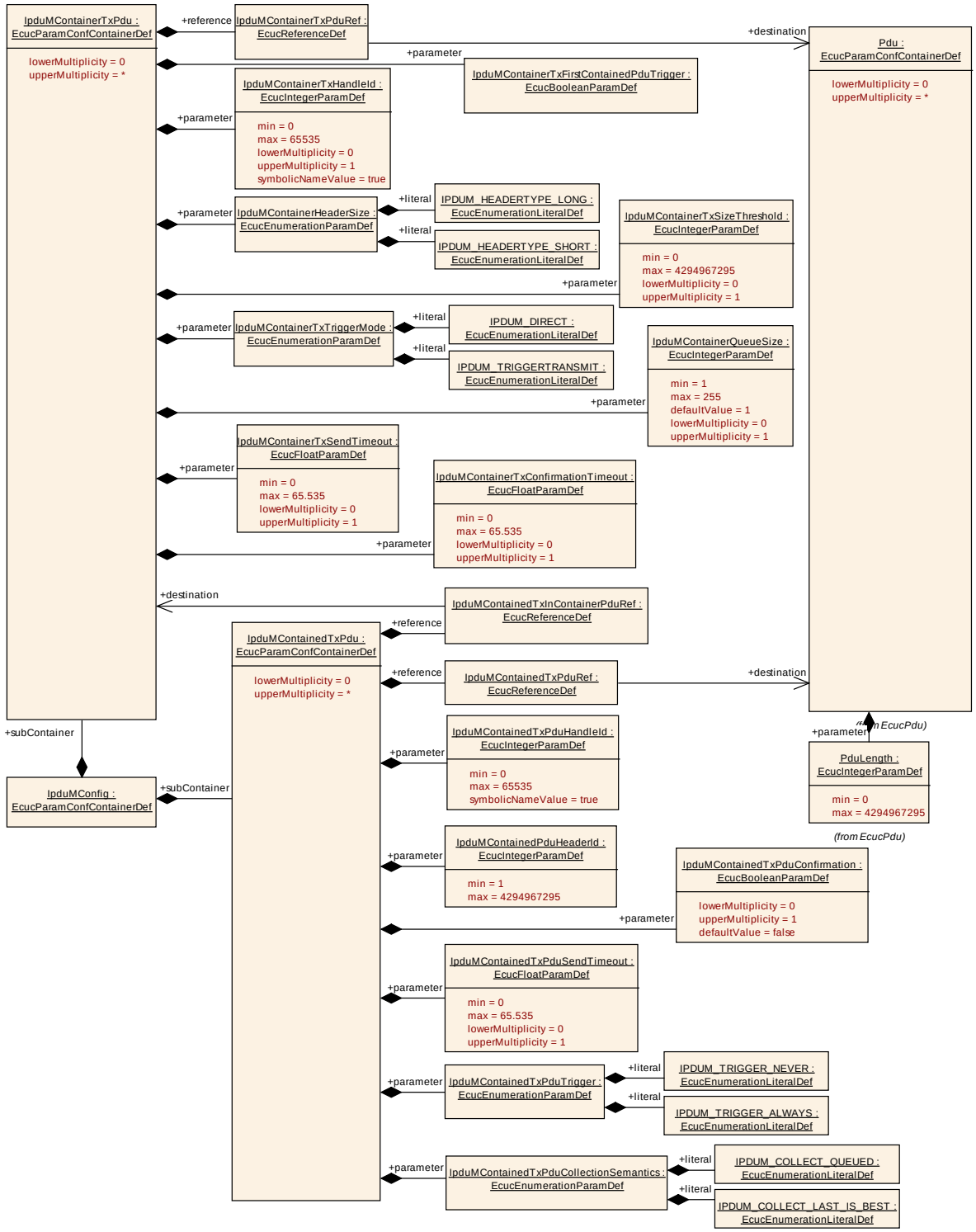

Figure 13 Configuration Overview TxContainer

10.2.21 IpduMContainerTxPdu

SWS Item	**ECUC_IpduM_00192 :**
Container Name	IpduMContainerTxPdu
Description	Configuration of a transmitted container Pdu.
Post-Build Variant Multiplicity	true

Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00183 :**		
Name	IpduMContainerHeaderSize		
Description	Defines the layout of the header information (header id and length).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_HEADERTYPE_LONG	Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
	IPDUM_HEADERTYPE_SHORT	Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00185 :**		
Name	IpduMContainerQueueSize		
Description	Defines a local queue for handling of each ContainerPdu. Defined in number of instances of this ContainerPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 255		
Default value	1		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00190 :**		
Name	IpduMContainerTxConfirmationTimeout		
Description	This timeout (in seconds) defines the timeout period for monitoring the reception of the TxConfirmation. It is not used when an I-PDU is requested using the trigger transmit API. If this Parameter is omitted, the default value shall be used.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. 65.535		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME

	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00199 :**		
Name	IpduMContainerTxFirstContainedPduTrigger		
Description	Defines if the transmission of this IpduMContainerTxPdu shall be requested right after the first IpduMContainedTxPdu was put into it.		
Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00191 :**		
Name	IpduMContainerTxHandleId		
Description	Handle Id used by the PduR for TxConfirmation and for TriggerTransmit of the ContainerPdu.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00194 :**		
Name	IpduMContainerTxSendTimeout		
Description	When this timeout expires the ContainerPdu is triggered for sending. The respective timer is started when the first Pdu is put into the ContainerPdu. Defined in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. 65.535		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00195 :**		
Name	IpduMContainerTxSizeThreshold		
Description	Defines the size threshold in bytes which, when exceeded, triggers the sending of the ContainerPdu although the maxium Pdu size (PduLength parameter of Pdu object) has not been reached yet.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00196 :**		
Name	IpduMContainerTxTriggerMode		
Description	Defines whether this ContainerPdu is fetched via trigger transmit.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_DIRECT	The IpduM sends this ContainerPdu when this ContainerPdu is triggered.	
	IPDUM_TRIGGERTRANSMIT	This ContainerPdu is stored in the IpduM and fetched via trigger transmit.	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00193 :**		
Name	IpduMContainerTxPduRef		
Description	Reference to the Pdu which represents the container and is used for transmission.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.2.22 IpduMContainedTxPdu

SWS Item	**ECUC_IpduM_00177 :**		
Container Name	IpduMContainedTxPdu		
Description	Configuration of a sender ContainedPdu.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

SWS Item	**ECUC_IpduM_00172 :**		
Name	IpduMContainedPduHeaderId		
Description	Header Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	1 .. 4294967295		
Default value	--		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00198 :**		
Name	IpduMContainedTxPduCollectionSemantics		
Description	Defines whether this IpduMContainedTxPdu shall be collected using a last-is-best or queued semantics.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_COLLECT_LAST_IS_BEST	The IpduMContainedTxPdu data will be fetched via TriggerTransmit just before the transmission executes.	
	IPDUM_COLLECT_QUEUED	The IpduMContainedTxPdu data will instantly be stored to the IpduMContainerTxPdu in the context of the Transmit API.	
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00178 :**
Name	IpduMContainedTxPduConfirmation
Description	This Parameter determines whether for this contained I-PDU a TxConfirmation shall be provided. If set to TRUE a TxConfirmation is issued. It is not used when an I-PDU is requested using the trigger transmit API. If this Parameter is omitted, the default value shall be used.
Multiplicity	0..1
Type	EcucBooleanParamDef
Default value	false

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

SWS Item	**ECUC_IpduM_00179 :**		
Name	IpduMContainedTxPduHandleId		
Description	Handle Id of the ContainedPdu.		
Multiplicity	1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Pre-compile time	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / Dependency	scope: ECU		

SWS Item	**ECUC_IpduM_00181 :**		
Name	IpduMContainedTxPduSendTimeout		
Description	Defines a ContainedPdu specific sender timeout which can reduce the ContainerPdu timer when this ContainedPdu is put inside the Contai-nerPdu. Defined in seconds.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. 65.535		
Default value	--		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00182 :**	
Name	IpduMContainedTxPduTrigger	
Description	Defines whether this Pdu triggers the sending of the ContainerPdu.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	IPDUM_TRIGGER_ALWAYS	This Pdu directly triggers the sending of the ContainerPdu.
	IPDUM_TRIGGER_NEVER	This Pdu does not triggers the sending of the ContainerPdu (other trigger criteria might still trigger sending of the ContainerPdu).
Post-Build Variant	true	

Value			
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00176 :**		
Name	IpduMContainedTxInContainerPduRef		
Description	Reference to the container Pdu which this contained Pdu shall be collected in.		
Multiplicity	1		
Type	Reference to [IpduMContainerTxPdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

SWS Item	**ECUC_IpduM_00180 :**		
Name	IpduMContainedTxPduRef		
Description	Reference to the Pdu which represents this ContainedPdu and is used for transmission.		
Multiplicity	1		
Type	Reference to [Pdu]		
Post-Build Variant Value	true		
Value Configuration Class	Pre-compile time	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

No Included Containers

10.3 Published Information

For details refer to the Chapter 10.3 Published Information in SWS_BSWGeneral.

10.3.1 IpduMPublishedInformation

SWS Item	**ECUC_IpduM_00141 :**
Container Name	IpduMPublishedInformation
Description	Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.
Configuration Parameters	

SWS Item	**ECUC_IpduM_00142 :**
Name	IpduMRxDirectComInvocation
Description	If set to TRUE the COM invocation optimization as defined in IPDUM140 is implemented.

Multiplicity	1		
Type	EcucBooleanParamDef		
Default value	--		
Post-Build Variant Value	false		
Value Configuration Class	Published Information	X	All Variants
Scope / Dependency	scope: local		

No Included Containers

10.4 Configuration Rules

10.4.1 Selector Field

[SWS_IpduM_00011] ⌈The number of values used of the selector field, i.e. values used to distinguish between different I-PDU layouts, does not have to be the whole range of possible values.⌋ (SRS_IpduM_02803)

Example: The size of a selector field with 3 bits leads to 2^3 possible selector field values; it shall be allowed to use only an arbitrary subset of these values. The used subset needs no to be contiguous.

10.4.2 Byte Order

The byte order of all segments and the selector field of a multiplexed I-PDU is restricted to be the same, see ECUC_IpduM_00162. Any necessary byte order conversion shall be handled within the COM module. The multiplexed I-PDUs in COM and IpduM have to be configured consistently to have the same endianness.

[SWS_IpduM_00166] ⌈The endianness of signals of the de-multiplexed I-PDUs configured in COM must match the endianness of the corresponding multiplexed I-PDU in IpduM as configured per IpduMByteOrder (ECUC_IpduM_00162). ⌋ (SRS_IpduM_02816, SRS_IpduM_02817)

The above configuration rule also restricts all COM signals of a multiplexed attribute to have the same endianness.

10.4.3 Multiple PDU to Container Mapping

[SWS_IpduM_00197] ⌈A TxConfirmation for a contained I-PDU shall only be configurable if for the corresponding Container PDU has a IpduMContainerTxConfirmationTimeout is configured.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00218] ⌈If the IpduMContainerTxSendTimeout is omitted all IpduMContainedTxPdu have to provide a IpduMContainedTxPduSendTimeout.⌋ (SRS_IpduM_02820)

[SWS_IpduM_00219] [IpduM shall reject configurations in which the transmit properties (see ECUC_IpduM_00198: IpduMContainedTxPduCollectionSemantics) of the contained I-PDUs which are assigned to a specific Container PDU are mixed. A Container PDU shall contain either solely I-PDUs with IPDUM_COLLECT_LAST_IS_BEST or solely I-PDUs with IPDUM_COLLECT_QUEUED semantic.] (SRS_IpduM_02821)

Container PDUs that have only I-PDUs assigned with IPDUM_COLLECT_LAST_IS_BEST semantic to can be realized buffer efficiently.

11 Not applicable requirements

[SWS_IpduM_00999] [These requirements are not applicable to this specification.] (SRS_BSW_00171, SRS_BSW_00375, SRS_BSW_00437, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00427, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00336, SRS_BSW_00339, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00386, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00314, SRS_BSW_00377)