Document Title	Classic Platform Release Overview
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	0

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R20-11

Document Change History

Date	Release	Changed by	Description
2020-11-30	R20-11	AUTOSAR Release Management	Release Life Cycle Status: R20-11 is in Evolution, R20-11 supersedes R19-11

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

References

[1] Specification of Operating System
AUTOSAR_SWS_OS

[2] SAE J1939-81 Network Management

[3] SOME/IP Protocol Specification
AUTOSAR_PRS_SOMEIPProtocol

[4] Specification of LIN Interface
AUTOSAR_SWS_LINInterface

[5] OPEN Sleep/Wake-up Specification for Automotive Ethernet
http://www.opensig.org/Automotive-Ethernet-Specifications/

1 Introduction

1.1 Scope of this document

This document provides an overview of the AUTOSAR standard Classic Platform Release R20-11.

1.2 Terminology and Licenses

1.2.1 Terminology statement

AUTOSAR has identified a use of previously common terminology that can be considered oppressive or racist, such as master/slave and black/white list, or in other contexts such as gender or age as harmful connotations. AUTOSAR is currently planning a discussion with all the working groups to replace these terms starting in R21-11. AUTOSAR is committed to provide all specification documents without these terminology in the coming and future releases. Nevertheless, it may take several releases before the terms are completely replaced, as AUTOSAR has to continue its operations and thousands of pages of existing specifications have to be reviewed and updated in parallel.

1.2.2 Usage of W3C XML schema

The AUTOSAR XML Schema requires the XML namespace definition file xml.xsd.

There are several occurrences of the "xml.xsd" file within this release. For all occurrences the W3C license applies which can be found on `https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document`.

License

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this work, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the work or portions thereof, including modifications:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software and Document Short Notice should be included. Notice of any changes or modifications, through a copyright statement on the new code or document such as “This software or document includes material copied from or derived from [title and URI of the W3C document]. Copyright © [YEAR] W3C® (MIT, ERCIM, Keio, Beihang).”

Disclaimers

THIS WORK IS PROVIDED “AS IS,” AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENT WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENT.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the work without specific, written prior permission. Title to copyright in this work will at all times remain with copyright holders.

1.3 AUTOSAR Standards

1.3.1 Introduction

AUTOSAR addresses a wide range of use cases in automotive software development with its standards. These use cases have different requirements and lead to different technical solutions.

Packaging its deliverables into different "standards"

- eases the access to AUTOSAR solutions for users and
- allows AUTOSAR to scale with market needs.

1.3.2 Definition

An AUTOSAR standard is a consistent set of AUTOSAR deliverables, which are released at the same time. AUTOSAR deliverables can, but are not limited to be of the following kinds:

- textual explanations
- textual specifications
- test specifications
- source code
- other formal or semi-formal textual formats (e.g. ARXML, UML models, XML Schemata)

At the time of release, AUTOSAR ensures that dependencies are fulfilled.

1.3.3 Overview on AUTOSAR's Standards

AUTOSAR delivers the following standards:

Standard	Abbreviation
Adaptive Platform	AP
Classic Platform	CP
Foundation	FO

1.3.3.1 Adaptive Platform

The Adaptive Platform is AUTOSAR's solution for high-performance computing ECUs to build safety-related systems for use cases such as highly automated and autonomous driving.

1.3.3.2 Classic Platform

The Classic Platform is AUTOSAR's solution for embedded systems with hard real-time and safety constraints.

1.3.3.3 Foundation

The purpose of the Foundation standard is to enforce interoperability between the AUTOSAR platforms.

Foundation contains the generic artifacts that are common for AP and CP to ensure compatibility between

- Classic- and Adaptive Platform
- Non-AUTOSAR platforms to AUTOSAR platforms

1.3.4 Dependencies between Standards

Each release of Classic and Adaptive Platform relies on a dedicated version of Foundation. The specific dependency is documented in chapter 1.4.6.

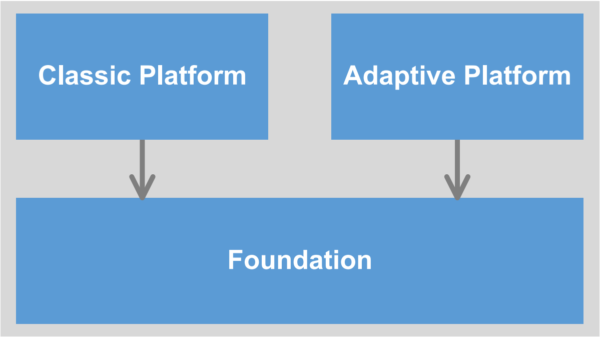


Figure 1.1: Dependencies of AUTOSAR Standards

1.3.5 Dependencies to other Standards

This release of the Classic Platform depends on the standard Foundation in release R20-11, which

- defines protocols implemented by Classic Platform
- contains the project objectives and the common requirements from which the features of the Classic Platform are derived
- contains common specification parts which apply to both, the Adaptive Platform and the Classic Platform.

These dependencies are refined in the trace information of the requirements in the respective specifications.

1.4 Release Numbering and Life Cycle

1.4.1 Platform release number

AUTOSAR applies a four-digit numbering scheme Ryy-mm to identify releases. The identifiers "yy" and "mm" depict the year and month of the release date, e.g. R20-11 for the November 2020 release.

1.4.2 Internal release number

AUTOSAR additionally maintains an internal release number for different purposes (e.g. usage in BSW modules in Classic Platform).
The internal release number is used for all platforms and follows up on the Classic Platform release number. In Adaptive Platform this is newly introduced. In Foundation this leads to a discontinuation of the former numbering pattern (e.g. R1.5.0).
A mapping list between Platform Releases and corresponding internal release num-

bers can be found in chapter 1.4.5. The internal release number uses a three-digit numbering scheme R<major>.<minor>.<revision> to identify releases. Its primary purpose is to identify a release as

- a major release: Valid and draft specification parts may be changed backward incompatibly.
- a minor release: Valid specification parts may only be changed backward compatibly. Draft specification parts may be changed backward incompatibly.
- a revision: Does not contain extensions but only backward compatible bugfixes.

1.4.3 Release life cycle of a major release

Each major release goes through four consecutive steps within its life cycle (examples based on the internal release numbering scheme):

1. Development: Between start of life cycle and the initial release (e.g. R4.0.1)
2. Evolution: Following the initial release with zero, one or several minor releases and/or revisions (e.g. R4.0.2, R4.1.1)
3. Maintenance: No new content is added to a major release but only maintenance of the existing content with zero, one or several revisions (e.g. R3.2.2) is provided
4. Issue Notice: No more revisions but zero, one or several issue notices, i.e. updates of the list of known issues until end of life cycle.

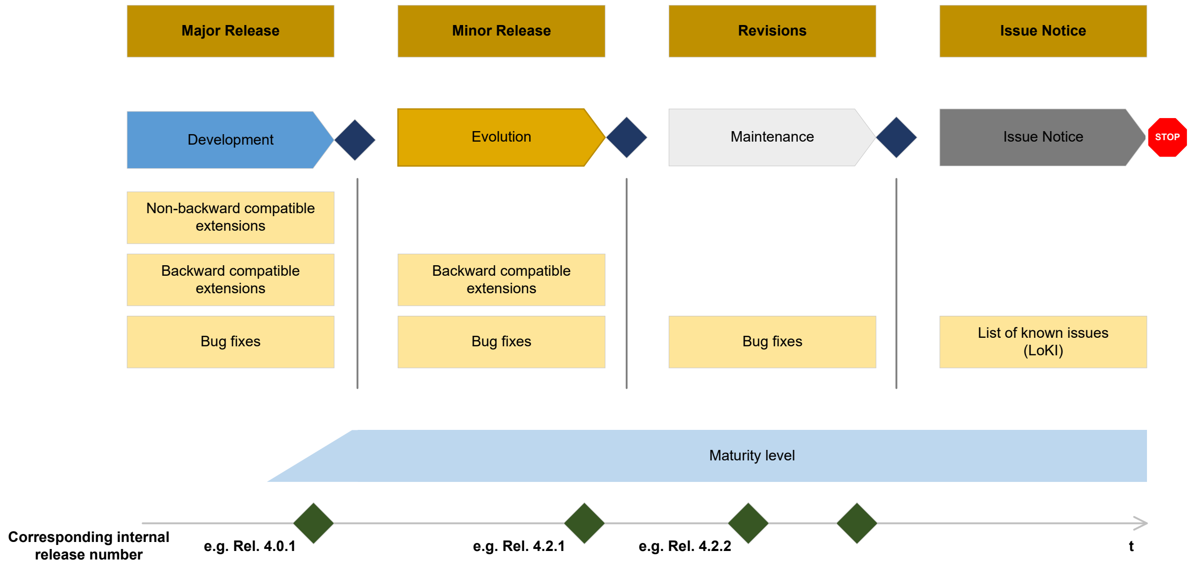


Figure 1.2: Life cycle model of AUTOSAR standards

1.4.4 Life cycle states of specification items and requirements

The life cycle state of a specification item is found after the specification item ID surrounded by curly brackets. The states are:

- {Valid}: This indicates that the related entity is a valid part of the document. This is the default and also applies if no dedicated life cycle status is annotated for the related entity.
- {Draft}: This indicates that the related entity is newly introduced but still experimental. This information is published but is subject to change without backward compatibility guarantee.
- {Obsolete}: This indicates that the related entity is subject to be removed in one of the following releases without further notice.

The life cycle state of a requirement is found in the attribute "type". The states are the same as the specification item states.

1.4.5 Overview of AUTOSAR schema versions and corresponding internal AUTOSAR releases

Schema Version	Platform release	Internal release number
AUTOSAR_00048	R19-11	R4.5.0
AUTOSAR_00049	R20-11	R4.6.0

According to the release life cycle of AUTOSAR the release R20-11 is a minor release.

1.4.6 Overview of AUTOSAR schema versions and corresponding valid AUTOSAR releases

The AUTOSAR schema does not have an impact on the Foundation. The Foundation releases are mentioned for the sake of completeness.

Schema Version	Classic Platform release	Adaptive Platform release	Foundation release
AUTOSAR_00042	R4.3.0	R17-03	R1.1.0
AUTOSAR_00043	R4.3.0	R17-10	R1.2.0
AUTOSAR_00044	R4.3.1	R17-10	R1.3.0
AUTOSAR_00045	R4.3.1	R18-03	R1.4.0
AUTOSAR_00046	R4.4.0	R18-10	R1.5.0
AUTOSAR_00047	R4.4.0	R19-03	R1.5.1

Schema Version	AUTOSAR release
AUTOSAR_00048	R19-11
AUTOSAR_00049	R20-11

1.5 Content of chapters

This document is structured as follows:

- Chapter 1 provides an introduction to AUTOSAR's release strategy and its standardization approach.
- Chapter 2 provides a summary of changes since the previous release of the Classic Platform.
- Chapter 3 contains the overview of specifications comprising the AUTOSAR release R20-11. This chapter is structured according to the clusters of AUTOSAR release R20-11.
- Chapter 4 contains remarks about known technical deficiencies.
- Chapter 5 contains the detailed revision history of all released specifications.

2 Summary of changes

This chapter contains a summary of changes which have been implemented since the previous release R19-11.

2.1 Release R20-11

Several concepts mainly affecting the Classic Platform have been introduced with release R20-11 thereby adding functionalities to the platform (Ethernet Wakeup On Dataline, Vehicle Motion Control Interface, Vehicle Network State Management, 10BASE-T1S, Classic Platform Flexibility, Rework of PNC related ComM and NM handling).
Additionally some concepts target the Classic and Adaptive Platform, strengthening the interaction between the two platforms.
Those concepts are related to security (Intrusion Detection System Manager) and development methodology and mechanisms (Unified Timing and Tracing Approach).

2.1.1 Concepts

2.1.1.1 Introduced Concepts

The following concepts in 2.1.1.1.1 - 2.1.1.1.8 have been introduced.

2.1.1.1.1 10BASE-T1S

This concept introduces the support of Ethernet 10BASE-T1S specified by IEEE802.3cg and enables bus topologies in Ethernet networks. This new extension localized on layers 1 and 2 of the OSI model is to be supported by Classic Platform as well as Adaptive.

2.1.1.1.2 Classic Platform Flexibility

The concept of Classic Platform Flexibility aims to split today's monolithic AUTOSAR Classic Platform binary into several software clusters that can be independently developed, integrated, tested, and programmed.

2.1.1.1.3 Vehicle Motion Control Interface

This concept is for the application interface of AD/ADAS and VMC. It also introduces the layer achitecture between AD/ADAS, VMC and actuators.

2.1.1.1.4 Ethernet Wakeup On Dataline

The concept extends the Ethernet communication stack to use OA TC10 compliant Ethernet hardware (PHY) in combination with existing communication features, e.g. partial network. OA TC10 compliant PHY's provide the possibility to wake up and sleep on dataline and to forward a received wake up to neighbouring PHY's. This support a Ethernet switched network, where a wake up could be propagate across the whole network. The essential properties is to trigger a wake up on the network upon an active communication request (e.g. a PNC request), to trigger a sleep if a communication channel shutdown and to configure a PHY properly regarding the wake up forwarding behaviour.

2.1.1.1.5 Unified Timing and Tracing Approach

The concept elaborates a unified approach for timing analysis, design, validation and verification, including the definition of suitable tracing methods, which can be applied to the Classic and Adaptive platform. This covers to build up a timing reference platform and to extend the timing description to different abstraction levels.

2.1.1.1.6 Vehicle Network State Management

The concept extends the existing PNC coordination algorithm which is based on static routings by the possibility to learn additional routings dynamically. This learning is implemented as a special phase within the PNC algorithm and can be triggered by application or diagnostic.

2.1.1.1.7 Rework of PNC related ComM and NM handling

Concept part 1 extends the network management protocol for partial network functionality to ensure a synchronized PNC shutdown across the partial network topology from the top-level PNC coordinator down to the subordinated PNC nodes. The essential property is a synchronized PNC shutdown even if one or multiple intermediate PNC coordinators are involved. This should avoid abnormalities in the network, e.g. timeout failure of expected data reception on application level or unexpected restart of communication channels of a PNC gateway.

2.1.1.1.8 Intrusion Detection System Manager

The concept “Intrusion Detection System Manager” specifies a framework for an AUTOSAR based Intrusion Detection System (IDS). This includes the BSW components "Intrusion Detection System Manager (IdsM)" and "Adaptive Intrusion Detection System Manager (Adaptive IdsM)". Furthermore extensions of basic software modules are specified to enable reporting of security events to the IdsM. A protocol specification for transmitting qualified security events over the vehicle network was released. The security extract template specified by concept “Intrusion Detection System Manager” allows to model properties of the IDS on system level.

2.1.1.2 Impact of Concepts

The introduced concepts had impact on several specifications. The following table provides a detailed overview.

Please note that some of the specifications are marked by special text formatting:

- Specifications in **bold** font are completely new specifications originating from the particular concept.
- Specifications in *italic* font are affected indirectly as they provide artefacts for the actually impacted specifications.

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Ethernet Wakeup On Dataline	Glossary	Foundation	draft
	System Template	Classic Platform	
	Specification of Ethernet Transceiver Driver		
	Specification of Ethernet Driver		
	Specification of Ethernet Switch Driver		
	Specification of Ethernet State Manager		
	Specification of Ethernet Interface		
	Specification of Communication Manager		
	Specification of Communication Manager		
	Specification of Basic Software Mode Manager		
	Requirements on Mode Management		
	Requirements on Ethernet Support in AUTOSAR		
	Guide to Mode Management		
Unified Timing and Tracing Approach	Specification of AUTOSAR Run-Time Interface	Classic Platform	draft
	Recommended Methods and Practices for Timing Analysis and Design within the AUTOSAR Development Process	Foundation	

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Vehicle Motion Control Interface	**Explanation of Application Interface of AD/ADAS vehicle motion control**	Classic Platform	draft
Vehicle Network State Manager	Specification of Network Management	Adaptive Platform	draft
	Specification of UDP Network Management	Classic Platform	
	Specification of Network Management Interface		
	Specification of FlexRay Network Management		
	Specification of Communication Manager		
	Specification of CAN Network Management		
	System Template		
	Requirements on Network Management		
	Requirements on Mode Management		
	Specification of the AUTOSAR Network Management Protocol	Foundation	
	Requirements on AUTOSAR Network Management		
10BASE-T1S	Specification of Manifest	Adaptive Platform	draft
	Specification of Time Synchronization over Ethernet	Classic Platform	
	Specification of TCP/IP Stack		
	Specification of Network Management Interface		
	Specification of Ethernet Transceiver Driver		
	Specification of Ethernet Switch Driver		
	Specification of Ethernet Interface		
	Specification of Ethernet Driver		
	System Template		
	Requirements on Ethernet Support in AUTOSAR		
	Specification of Socket Adaptor		
Classic Platform Flexibility	Glossary	Foundation	draft
	Main Requirements		
	Requirements on Timing Extensions	Foundation	

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Classic Platform Flexibility	List of Basic Software Modules	Classic Platform	draft
	Specification of Software Cluster Connection module		
	Specification of Timing Extensions		
	System Template		
	Software Component Template		
	Basic Software Module Description Template		
	Specification of RTE Software		
	Specification of Operating System		
	Requirements on Software Cluster Connection module		
	Requirements on Runtime Environment		
	Requirements on System Template		
	Layered Software Architecture		
	Explanation of CP Software Cluster Design And Integration Guideline		
Intrusion Detection System Manager	**Specification of Intrusion Detection System Protocol**	Foundation	partially validated
	Requirements on Intrusion Detection System		
	Requirements on Security Extract Template		
	Security Extract Template		
	Glossary		
	Requirements on Diagnostic Extract Template	Classic Platform	
	Requirements on AUTOSAR Features		
	Specification of Socket Adaptor		
	Layered Software Architecture		
	Specification of Key Manager		
	Specification of Secure Onboard Communication		

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Intrusion Detection System Manager	General Requirements on Basic Software Modules	Classic Platform	partially validated
	General Specification of Basic Software Modules		
	Specification of CAN Driver		
	Specification of CAN Interface		
	Specification of Diagnostic Communication Manager		
	Specification of Diagnostic Event Manager		
	Specification of Ethernet Interface		
	Specification of Intrusion Detection System Manager		
	Specification of NVRAM Manager		
	Specification of TCP/IP Stack		
	Diagnostic Extract Template		
	Software Component Template		
	List of Basic Software Modules		
	Requirements on Manifest Specification	Adaptive Platform	
	Specification of Intrusion Detection System Manager for Adaptive Platform		
	Specification of Communication Management		
	Specification of Cryptography for Adaptive Platform		
	Specification of Manifest		
Rework Of PNC related ComM and NM handling	Specification of the AUTOSAR Network Management Protocol	Foundation	draft
	Requirements on AUTOSAR Network Management		
	Requirements on Mode Management	Classic Platform	
	Requirements on Network Management		
	Specification of CAN Network Management		
	Specification of Communication Manager		

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Rework Of PNC related ComM and NM handling	Specification of Ethernet Driver	Classic Platform	draft
	Specification of FlexRay Network Management		
	Specification of Network Management Interface		
	Specification of UDP Network Management		

Table 2.1: Impact of Concepts

2.1.1.3 Validated Concepts

The following concepts have been validated:

- AUTOSAR Run Time Interface (ARTI)
- BSW Multicore Distribution
- DoIP Extension
- Signal Service Translation
- Non-Volatile Data Handling Enhancements
- Firmware over the Air (FOTA)

2.1.2 Specifications

2.1.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Requirements on Software Cluster Connection module (UID 973, SRS)
- Specification of Software Cluster Connection module (UID 974, SWS)
- Explanation of CP Software Cluster Design And Integration Guideline (UID 975, EXP)
- Specification of Intrusion Detection System Manager (UID 977, SWS)
- Explanation of Application Interface of AD/ADAS vehicle motion control (UID 988, SXP)

In addition to the above listed new specifications, the following documents have been added with R20-11:

- none

2.1.2.2 Migrated Specifications

With this release, the following specifications have been moved from AUTOSAR Classic Platform to the AUTOSAR Foundation standard:

- Recommended Methods and Practices for Timing Analysis and Design within the AUTOSAR Development Process (UID 645, TR)
- Meta Model (UID 59, MMOD)
- XML Schema Production Rules (UID 122, TPS)
- Generic Structure Template (UID 202, TPS)
- Meta Model-generated XML Schema (UID 230, MMOD)
- Standardization Template (UID 535, TPS)
- Requirements on Standardization Template (UID 536, RS)
- Predefined Names in AUTOSAR (UID 600, TR)
- AUTOSAR Miscellaneous Support Files (UID 603, MOD)
- AUTOSAR Feature Model Exchange Format Requirements (UID 605, RS)
- AUTOSAR Feature Model Exchange Format (UID 606, TPS)
- Collection of blueprints for AUTOSAR M1 models (UID 621, MOD)
- Collection of constraints on AUTOSAR M1 models (UID 635, TR)
- Standardized M1 Models used for the Definition of AUTOSAR (UID 636, MOD)
- ARXML Serialization Rules (UID 779, TPS)
- Interoperability of Autosar Tools Supplement (UID 786, TR)
- Supplementary material of the AUTOSAR XML Schema (UID 649, TR)

2.1.2.3 Obsolete Specifications

The following specifications have been set to status "obsolete" in this release:

- Requirements on AUTOSAR Features (UID 294, RS)

2.1.2.4 Removed Specifications

The following specifications have been set to status "removed" in this release and hence are not released anymore:

- General Requirements on Methodology and Templates (UID 604, RS)

2.1.2.5 Reworked Specifications

The following documents have been changed fundamentally in R20-11:

- none

2.1.2.6 Moved Specification parts

The following specification parts have been moved to other documents in R20-11.

- none

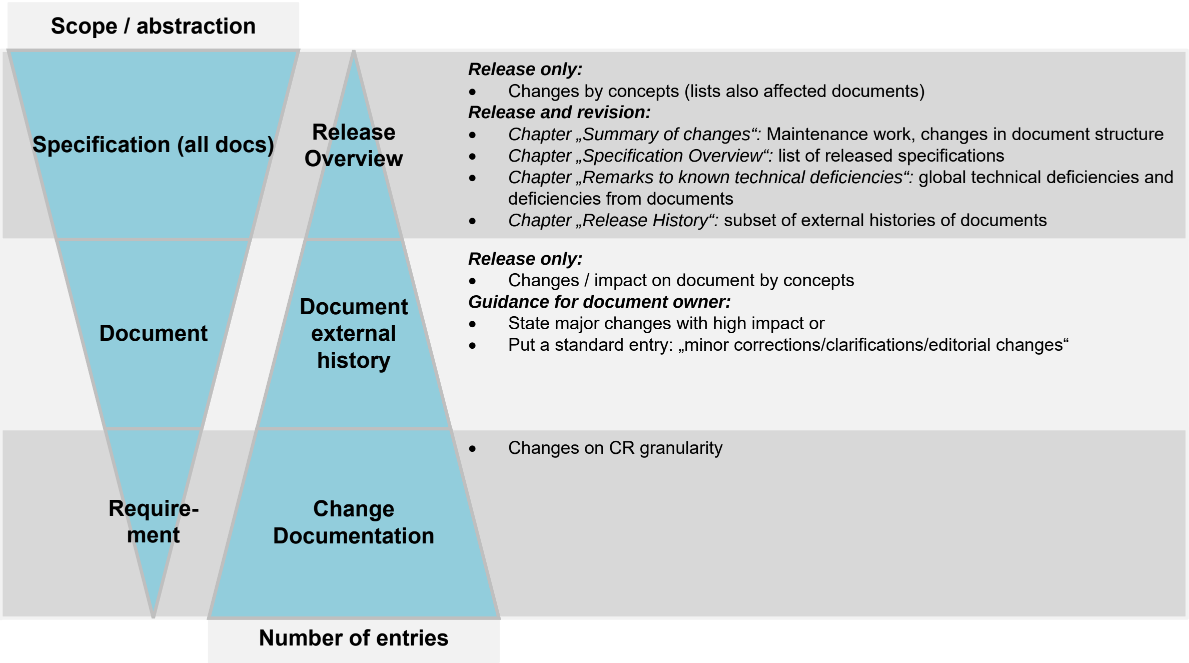
2.1.3 Release Documentation

There are no major changes in the Release Documentation.

2.2 History information in AUTOSAR

The following diagram shows the location of documentation of changes.

The Change Documentation will be available for Adaptive Platform starting with R20-11.



Release only:
- Changes by concepts (lists also affected documents)

Release and revision:
- *Chapter „Summary of changes“:* Maintenance work, changes in document structure
- *Chapter „Specification Overview“:* list of released specifications
- *Chapter „Remarks to known technical deficiencies“:* global technical deficiencies and deficiencies from documents
- *Chapter „Release History“:* subset of external histories of documents

Release only:
- Changes / impact on document by concepts

Guidance for document owner:
- State major changes with high impact or
- Put a standard entry: „minor corrections/clarifications/editorial changes“

- Changes on CR granularity

Figure 2.1: History information in AUTOSAR

3 Specification overview

The published specifications are divided into the clusters

- Release Documentation
- Communication
- Memory
- System Services
- MCAL
- IO
- Libraries
- Diagnostics
- Safety
- BSW General
- General
- Methodology and Templates
- Mode Management
- RTE
- Application Interfaces
- Crypto
- Global Time
- SWArch
- Security

The assignment of the specifications to these clusters is shown below.

Long Name	File Name	Life cycle changes
Release Documentation		
Classic Platform Release Overview	AUTOSAR_TR_ClassicPlatformReleaseOverview	
AUTOSAR Classic Platform Specification Hashes	AUTOSAR_TR_ClassicPlatformSpecificationHashes	
Communication		
General Specification of Transformers	AUTOSAR_ASWS_TransformerGeneral	
Requirements on BSW Modules for SAE J1939	AUTOSAR_SRS_SAEJ1939	
Requirements on Bus Mirroring	AUTOSAR_SRS_BusMirroring	
Requirements on CAN	AUTOSAR_SRS_CAN	
Requirements on Communication	AUTOSAR_SRS_COM	
Requirements on Ethernet Support in AUTOSAR	AUTOSAR_SRS_Ethernet	
Requirements on FlexRay	AUTOSAR_SRS_FlexRay	
Requirements on Gateway	AUTOSAR_SRS_Gateway	
Requirements on I-PDU Multiplexer	AUTOSAR_SRS_IPDUMultiplexer	
Requirements on LIN	AUTOSAR_SRS_LIN	
Requirements on Module XCP	AUTOSAR_SRS_XCP	
Requirements on Network Management	AUTOSAR_SRS_NetworkManagement	
Requirements on Secure Onboard Communication	AUTOSAR_SRS_SecureOnboardCommunication	
Requirements on SPI Handler / Driver	AUTOSAR_SRS_SPIHandlerDriver	
Requirements on Transformer	AUTOSAR_SRS_Transformer	
Requirements on TTCAN	AUTOSAR_SRS_TTCAN	
Requirements on Vehicle-2-X Communication	AUTOSAR_SRS_V2XCommunication	
Specification of a Request Manager for SAE J1939	AUTOSAR_SWS_SAEJ1939RequestManager	
Specification of a Transport Layer for SAE J1939	AUTOSAR_SWS_SAEJ1939TransportLayer	
Specification of Bus Mirroring	AUTOSAR_SWS_BusMirroring	
Specification of CAN Driver	AUTOSAR_SWS_CANDriver	
Specification of CAN Interface	AUTOSAR_SWS_CANInterface	
Specification of CAN Network Management	AUTOSAR_SWS_CANNetworkManagement	
Specification of CAN State Manager	AUTOSAR_SWS_CANStateManager	
Specification of CAN Transceiver Driver	AUTOSAR_SWS_CANTransceiverDriver	
Specification of CAN Transport Layer	AUTOSAR_SWS_CANTransportLayer	
Specification of COM Based Transformer	AUTOSAR_SWS_COMBasedTransformer	
Specification of Communication	AUTOSAR_SWS_COM	
Specification of Diagnostic Log and Trace	AUTOSAR_SWS_DiagnosticLogAndTrace	
Specification of Diagnostic over IP	AUTOSAR_SWS_DiagnosticOverIP	

▽

△

Long Name	File Name	Life cycle changes
Specification of Ethernet Driver	AUTOSAR_SWS_EthernetDriver	
Specification of Ethernet Interface	AUTOSAR_SWS_EthernetInterface	
Specification of Ethernet State Manager	AUTOSAR_SWS_EthernetStateManager	
Specification of Ethernet Switch Driver	AUTOSAR_SWS_EthernetSwitchDriver	
Specification of Ethernet Transceiver Driver	AUTOSAR_SWS_EthernetTransceiverDriver	
Specification of FlexRay AUTOSAR Transport Layer	AUTOSAR_SWS_FlexRayARTransportLayer	
Specification of FlexRay Driver	AUTOSAR_SWS_FlexRayDriver	
Specification of FlexRay Interface	AUTOSAR_SWS_FlexRayInterface	
Specification of FlexRay ISO Transport Layer	AUTOSAR_SWS_FlexRayISOTransportLayer	
Specification of FlexRay Network Management	AUTOSAR_SWS_FlexRayNetworkManagement	
Specification of FlexRay State Manager	AUTOSAR_SWS_FlexRayStateManager	
Specification of FlexRay Transceiver Driver	AUTOSAR_SWS_FlexRayTransceiverDriver	
Specification of I-PDU Multiplexer	AUTOSAR_SWS_IPDUMultiplexer	
Specification of Large Data COM	AUTOSAR_SWS_LargeDataCOM	
Specification of LIN Driver	AUTOSAR_SWS_LINDriver	
Specification of LIN Interface	AUTOSAR_SWS_LINInterface	
Specification of LIN State Manager	AUTOSAR_SWS_LINStateManager	
Specification of LIN Transceiver Driver	AUTOSAR_SWS_LINTransceiverDriver	
Specification of Module E2E Transformer	AUTOSAR_SWS_E2ETransformer	
Specification of Module XCP	AUTOSAR_SWS_XCP	
Specification of Network Management for SAE J1939	AUTOSAR_SWS_SAEJ1939NetworkManagement	
Specification of Network Management Interface	AUTOSAR_SWS_NetworkManagementInterface	
Specification of PDU Router	AUTOSAR_SWS_PDURouter	
Specification of Secure Onboard Communication	AUTOSAR_SWS_SecureOnboardCommunication	
Specification of Service Discovery	AUTOSAR_SWS_ServiceDiscovery	
Specification of Socket Adaptor	AUTOSAR_SWS_SocketAdaptor	
Specification of SOME/IP Transformer	AUTOSAR_SWS_SOMEIPTransformer	
Specification of SPI Handler / Driver	AUTOSAR_SWS_SPIHandlerDriver	
Specification of TCP/IP Stack	AUTOSAR_SWS_TcpIp	
Specification of TTCAN Driver	AUTOSAR_SWS_TTCANDriver	
Specification of TTCAN Interface	AUTOSAR_SWS_TTCANInterface	
Specification of UDP Network Management	AUTOSAR_SWS_UDPNetworkManagement	
Specification of Vehicle-2-X Basic Transport	AUTOSAR_SWS_V2XBasicTransport	

▽

△

Long Name	File Name	Life cycle changes
Specification of Vehicle-2-X Facilities	AUTOSAR_SWS_V2XFacilities	
Specification of Vehicle-2-X Geo Networking	AUTOSAR_SWS_V2XGeoNetworking	
Specification of Vehicle-2-X Management	AUTOSAR_SWS_V2XManagement	
Specification of Wireless Ethernet Driver	AUTOSAR_SWS_WirelessEthernetDriver	
Specification of Wireless Ethernet Transceiver Driver	AUTOSAR_SWS_WirelessEthernetTransceiverDriver	
Specification on SOME/IP Transport Protocol	AUTOSAR_SWS_SOMEIPTransportProtocol	
Memory		
Explanation of Firmware Over-The-Air	AUTOSAR_EXP_FirmwareOverTheAir	
NV Data Handling Guideline	AUTOSAR_EXP_NVDataHandling	
Requirements on EEPROM Driver	AUTOSAR_SRS_EEPROMDriver	
Requirements on Firmware Over-The-Air	AUTOSAR_RS_FirmwareOverTheAir	
Requirements on Flash Driver	AUTOSAR_SRS_FlashDriver	
Requirements on Flash Test	AUTOSAR_SRS_FlashTest	
Requirements on Memory Hardware Abstraction Layer	AUTOSAR_SRS_MemoryHWAbstractionLayer	
Requirements on Memory Services	AUTOSAR_SRS_MemoryServices	
Requirements on RAM Test	AUTOSAR_SRS_RAMTest	
Specification of EEPROM Abstraction	AUTOSAR_SWS_EEPROMAbstraction	
Specification of EEPROM Driver	AUTOSAR_SWS_EEPROMDriver	
Specification of Flash Driver	AUTOSAR_SWS_FlashDriver	
Specification of Flash EEPROM Emulation	AUTOSAR_SWS_FlashEEPROMEmulation	
Specification of Flash Test	AUTOSAR_SWS_FlashTest	
Specification of Memory Abstraction Interface	AUTOSAR_SWS_MemoryAbstractionInterface	
Specification of Memory Mapping	AUTOSAR_SWS_MemoryMapping	
Specification of NVRAM Manager	AUTOSAR_SWS_NVRAMManager	
Specification of RAM Test	AUTOSAR_SWS_RAMTest	
System Services		
Explanation of CP Software Cluster Design And Integration Guideline	AUTOSAR_EXP_CPSwClusterDesignAndIntegrationGuideline	Initial release
Requirements on Free Running Timer	AUTOSAR_SRS_FreeRunningTimer	
Requirements on Function Inhibition Manager	AUTOSAR_SRS_FunctionInhibitionManager	
Requirements on Hardware Test Manager on start up and shutdown	AUTOSAR_SRS_HWTestManager	
Requirements on Operating System	AUTOSAR_SRS_OS	
Requirements on Software Cluster Connection module	AUTOSAR_SRS_SoftwareClusterConnection	Initial release
Requirements on Time Service	AUTOSAR_SRS_TimeService	

▽

△

Long Name	File Name	Life cycle changes
Specification and Integration of Hardware Test Management at start up and shutdown	AUTOSAR_TR_HWTestManagementIntegrationGuide	
Specification of Communication Manager	AUTOSAR_SWS_COMManager	
Specification of Default Error Tracer	AUTOSAR_SWS_DefaultErrorTracer	
Specification of Function Inhibition Manager	AUTOSAR_SWS_FunctionInhibitionManager	
Specification of Hardware Test Manager on start up and shutdown	AUTOSAR_SWS_HWTestManager	
Specification of Operating System	AUTOSAR_SWS_OS	
Specification of Software Cluster Connection module	AUTOSAR_SWS_SoftwareClusterConnection	Initial release
Specification of Time Service	AUTOSAR_SWS_TimeService	
MCAL		
General Requirements on SPAL	AUTOSAR_SRS_SPALGeneral	
Requirements on Core Test	AUTOSAR_SRS_CoreTest	
Requirements on GPT Driver	AUTOSAR_SRS_GPTDriver	
Requirements on MCU Driver	AUTOSAR_SRS_MCUDriver	
Specification of Core Test	AUTOSAR_SWS_CoreTest	
Specification of GPT Driver	AUTOSAR_SWS_GPTDriver	
Specification of MCU Driver	AUTOSAR_SWS_MCUDriver	
IO		
Requirements on ADC Driver	AUTOSAR_SRS_ADCDriver	
Requirements on DIO Driver	AUTOSAR_SRS_DIODriver	
Requirements on I/O Hardware Abstraction	AUTOSAR_SRS_IOHWAbstraction	
Requirements on ICU Driver	AUTOSAR_SRS_ICUDriver	
Requirements on OCU Driver	AUTOSAR_SRS_OCUDriver	
Requirements on Port Driver	AUTOSAR_SRS_PortDriver	
Requirements on PWM Driver	AUTOSAR_SRS_PWMDriver	
Specification of ADC Driver	AUTOSAR_SWS_ADCDriver	
Specification of DIO Driver	AUTOSAR_SWS_DIODriver	
Specification of I/O Hardware Abstraction	AUTOSAR_SWS_IOHardwareAbstraction	
Specification of ICU Driver	AUTOSAR_SWS_ICUDriver	
Specification of OCU Driver	AUTOSAR_SWS_OCUDriver	
Specification of Port Driver	AUTOSAR_SWS_PortDriver	
Specification of PWM Driver	AUTOSAR_SWS_PWMDriver	
Libraries		
Macro Encapsulation of Interpolation Calls	AUTOSAR_EXP_MacroEncapsulationofInterpolationCalls	
Requirements on Libraries	AUTOSAR_SRS_Libraries	
Specification of Basic Software Multicore Library	AUTOSAR_SWS_BSWMulticoreLibrary	
Specification of Bit Handling Routines	AUTOSAR_SWS_BFXLibrary	

▽

△

Long Name	File Name	Life cycle changes
Specification of CRC Routines	AUTOSAR_SWS_CRCLibrary	
Specification of Extended Fixed Point Routines	AUTOSAR_SWS_EFXLibrary	
Specification of Fixed Point Interpolation Routines	AUTOSAR_SWS_IFXLibrary	
Specification of Fixed Point Math Routines	AUTOSAR_SWS_MFXLibrary	
Specification of Floating Point Interpolation Routines	AUTOSAR_SWS_IFLLibrary	
Specification of Floating Point Math Routines	AUTOSAR_SWS_MFLLibrary	
Specification of SW-C End-to-End Communication Protection Library	AUTOSAR_SWS_E2ELibrary	
Diagnostics		
Specification of a Diagnostic Communication Manager for SAE J1939	AUTOSAR_SWS_SAEJ1939 DiagnosticCommunicationManager	
Specification of Diagnostic Communication Manager	AUTOSAR_SWS_ DiagnosticCommunicationManager	
Specification of Diagnostic Event Manager	AUTOSAR_SWS_ DiagnosticEventManager	
Safety		
Overview of Functional Safety Measures in AUTOSAR	AUTOSAR_EXP_ FunctionalSafetyMeasures	
Requirements on Safety Extensions	AUTOSAR_RS_SafetyExtensions	
Requirements on Watchdog Driver	AUTOSAR_SRS_WatchdogDriver	
Safety Use Case Example	AUTOSAR_EXP_SafetyUseCase	
Specification of Safety Extensions	AUTOSAR_TPS_SafetyExtensions	
Specification of Watchdog Driver	AUTOSAR_SWS_WatchdogDriver	
Specification of Watchdog Interface	AUTOSAR_SWS_WatchdogInterface	
Specification of Watchdog Manager	AUTOSAR_SWS_WatchdogManager	
BSW General		
Basic Software UML Model	AUTOSAR_MOD_BSWUMLModel	
Complex Driver design and integration guideline	AUTOSAR_EXP_ CDDDesignAndIntegrationGuideline	
Description of the AUTOSAR standard errors	AUTOSAR_EXP_ErrorDescription	
Explanation of Error Handling on Application Level	AUTOSAR_EXP_ ApplicationLevelErrorHandling	
Explanation of Interrupt Handling within AUTOSAR	AUTOSAR_EXP_ InterruptHandlingExplanation	
General Requirements on Basic Software Modules	AUTOSAR_SRS_BSWGeneral	
General Specification of Basic Software Modules	AUTOSAR_SWS_BSWGeneral	
Guide to BSW Distribution	AUTOSAR_EXP_ BSWDistributionGuide	
List of Basic Software Modules	AUTOSAR_TR_BSWModuleList	
Modeling Guidelines of Basic Software EA UML Model	AUTOSAR_TR_ BSWUMLModelModelingGuide	

▽

△

Long Name	File Name	Life cycle changes
Specification of Communication Stack Types	AUTOSAR_SWS_CommunicationStackTypes	
Specification of Compiler Abstraction	AUTOSAR_SWS_CompilerAbstraction	
Specification of Platform Types	AUTOSAR_SWS_PlatformTypes	
Specification of Standard Types	AUTOSAR_SWS_StandardTypes	
General		
Layered Software Architecture	AUTOSAR_EXP_LayeredSoftwareArchitecture	
Requirements on AUTOSAR Features	AUTOSAR_RS_Features	obsolete
Specification of Bulk NvData Manager	AUTOSAR_SWS_BulkNvDataManager	
Virtual Functional Bus	AUTOSAR_EXP_VFB	
Methodology and Templates		
Basic Software Module Description Template	AUTOSAR_TPS_BSWModuleDescriptionTemplate	
Diagnostic Extract Template	AUTOSAR_TPS_DiagnosticExtractTemplate	
Integration of Franca IDL Software Component Descriptions	AUTOSAR_TR_FrancaIntegration	
Methodology	AUTOSAR_TR_Methodology	
Modeling Show Cases Examples	AUTOSAR_EXP_ModelingShowCases	
Modeling Show Cases Report	AUTOSAR_TR_ModelingShowCases	
Requirements on Basic Software Module Description Template	AUTOSAR_RS_BSWModuleDescriptionTemplate	
Requirements on Diagnostic Extract Template	AUTOSAR_RS_DiagnosticExtractTemplate	
Requirements on ECU Configuration	AUTOSAR_RS_ECUConfiguration	
Requirements on ECU Resource Template	AUTOSAR_RS_ECUResourceTemplate	
Requirements on Software Component Template	AUTOSAR_RS_SoftwareComponentTemplate	
Requirements on System Template	AUTOSAR_RS_SystemTemplate	
Software Component Template	AUTOSAR_TPS_SoftwareComponentTemplate	
Specification of ECU Configuration	AUTOSAR_TPS_ECUConfiguration	
Specification of ECU Configuration Parameters (XML)	AUTOSAR_MOD_ECUConfigurationParameters	
Specification of ECU Resource Template	AUTOSAR_TPS_ECUResourceTemplate	
Specification of Timing Extensions	AUTOSAR_TPS_TimingExtensions	
Supplementary material of general blueprints for AUTOSAR	AUTOSAR_TR_GeneralBlueprintsSupplement	
System Template	AUTOSAR_TPS_SystemTemplate	
Mode Management		
Guide to Mode Management	AUTOSAR_EXP_ModeManagementGuide	
Requirements on Mode Management	AUTOSAR_SRS_ModeManagement	
Specification of Basic Software Mode Manager	AUTOSAR_SWS_BSWModeManager	

▽

△

Long Name	File Name	Life cycle changes
Specification of ECU State Manager	AUTOSAR_SWS_ECUStateManager	
RTE		
Requirements on Runtime Environment	AUTOSAR_SRS_RTE	
Specification of RTE Software	AUTOSAR_SWS_RTE	
Application Interfaces		
Application Design Patterns Catalogue	AUTOSAR_TR_AIDesignPatternsCatalogue	
Application Interface Examples	AUTOSAR_MOD_AISpecificationExamples	
Application Interfaces User Guide	AUTOSAR_EXP_AIUserGuide	obsolete
Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	AUTOSAR_EXP_AIOccupantAndPedestrianSafety	
Explanation of Application Interfaces of the Body and Comfort Domain	AUTOSAR_EXP_AIBodyAndComfort	
Explanation of Application Interfaces of the Chassis Domain	AUTOSAR_EXP_AIChassis	
Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	AUTOSAR_EXP_AIHMIMultimediaAndTelematics	
Explanation of Application Interface of AD/ADAS vehicle motion control	AUTOSAR_EXP_AIADASAndVMC	Initial release
Explanation of Application Interfaces of the Powertrain Engine Domain	AUTOSAR_EXP_AIPowertrain	
Requirements on SW-C and System Modeling	AUTOSAR_RS_SWCModeling	
SW-C and System Modeling Guide	AUTOSAR_TR_SWCModelingGuide	
Unique Names for Documentation, Measurement and Calibration: Modeling and Naming Aspects including Automatic Generation	AUTOSAR_TR_AIMeasurementCalibrationDiagnostics	
XML Specification of Application Interfaces	AUTOSAR_MOD_AISpecification	
Crypto		
Requirements on Crypto Stack	AUTOSAR_SRS_CryptoStack	
Specification of Crypto Driver	AUTOSAR_SWS_CryptoDriver	
Specification of Crypto Interface	AUTOSAR_SWS_CryptoInterface	
Specification of Crypto Service Manager	AUTOSAR_SWS_CryptoServiceManager	
Specification of Key Manager	AUTOSAR_SWS_KeyManager	
Utilization of Crypto Services	AUTOSAR_EXP_UtilizationOfCryptoServices	
Global Time		
Specification of Synchronized Time-Base Manager	AUTOSAR_SWS_SynchronizedTimeBaseManager	
Specification of Time Synchronization over CAN	AUTOSAR_SWS_TimeSyncOverCAN	
Specification of Time Synchronization over Ethernet	AUTOSAR_SWS_TimeSyncOverEthernet	

▽

△

Long Name	File Name	Life cycle changes
Specification of Time Synchronization over FlexRay	AUTOSAR_SWS_ TimeSyncOverFlexRay	
SWArch		
Explanatory Document for usage of AUTOSAR RunTimeInterface	AUTOSAR_EXP_ClassicPlatformARTI	
Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components	AUTOSAR_RS_ ClassicPlatformDebugTraceProfile	
Specification of AUTOSAR Run-Time Interface	AUTOSAR_SWS_ClassicPlatformARTI	
Security		
Specification of Intrusion Detection System Manager	AUTOSAR_SWS_ IntrusionDetectionSystemManager	Initial release

Table 3.1: Specification Overview

4 Remarks to known technical deficiencies

The technical deficiencies per specification are - if applicable - mentioned inside the respective specification in a chapter "Known Limitations" located after the table of contents.

The following technical deficiencies are to be mentioned, where clicking on the section reference will bring you to the respective document:

Document UID	Long Name	Document Type	Section Reference
10	Specification of ADC Driver	SWS	4.19
13	Specification of CAN Network Management	SWS	4.20
26	Specification of FlexRay Driver	SWS	4.31
27	Specification of FlexRay Interface	SWS	4.28
31	Specification of MCU Driver	SWS	4.38
34	Specification of Operating System	SWS	4.8
35	Specification of PDU Router	SWS	4.26
72	Specification of LIN Driver	SWS	4.33
73	Specification of LIN Interface	SWS	4.10
78	Specification of ECU State Manager	SWS	4.7
80	Specification of Watchdog Manager	SWS	4.11
82	Specification of Function Inhibition Manager	SWS	4.12
228	Specification of Network Management Interface	SWS	4.14
253	Specification of CAN State Manager	SWS	4.6
254	Specification of FlexRay State Manager	SWS	4.13
286	Specification of Flash EEPROM Emulation	SWS	4.39
326	Specification of Large Data COM	SWS	4.37
351	Specification of Diagnostic Log and Trace	SWS	4.3
402	Specification of Crypto Service Manager	SWS	4.1 4.2
415	Specification of Ethernet State Manager	SWS	4.36
416	Specification of Socket Adaptor	SWS	4.32
417	Specification of Ethernet Interface	SWS	4.27
420	Specification of Synchronized Time-Base Manager	SWS	4.18
425	Specification of a Transport Layer for SAE J1939	SWS	4.25
428	Specification of SW-C End-to-End Communication Protection Library	SWS	4.29
430	Specification of Ethernet Driver	SWS	4.40
431	Specification of Ethernet Transceiver Driver	SWS	4.41
610	Specification of a Diagnostic Communication Manager for SAE J1939	SWS	4.22
611	Specification of a Request Manager for SAE J1939	SWS	4.24
612	Specification of Network Management for SAE J1939	SWS	4.23
617	Specification of TCP/IP Stack	SWS	4.4
631	Guide to BSW Distribution	EXP	4.9
650	Specification of Module E2E Transformer	SWS	4.30
656	Specification of Ethernet Switch Driver	SWS	4.15

▽

△

Document UID	Long Name	Document Type	Section Reference
674	Specification of Time Synchronization over CAN	SWS	4.17
675	Specification of Time Synchronization over FlexRay	SWS	4.16
798	Specification of Wireless Ethernet Driver	SWS	4.34
799	Specification of Wireless Ethernet Transceiver Driver	SWS	4.35
806	Specification of Crypto Interface	SWS	4.1
807	Specification of Crypto Driver	SWS	4.1
873	Specification of Bus Mirroring	SWS	4.21
907	Specification of Key Manager	SWS	4.1
974	Specification of Software Cluster Connection module	SWS	4.5

Table 4.1: Overview of known technical deficiencies

4.1 Key Management

In AUTOSAR release R19-11, the parts of the specification related to Certificate handling have been moved from Crypto Service Manager (UID 402, SWS), Crypto Interface (UID 806, SWS) and Crypto Driver (UID 807, SWS) to the Key Manager (UID 907, SWS). Details of the changes are described in Classic Platform Release Overview for release R19-11.

However, due to this change, some specific features were removed. The design of the interfaces from Crypto Service Manager allowed to hold and operate certificates in a Hardware Security Module (HSM).

The former interfaces allowed to:

- permanently store the certificate in the HSM
- parse the certificate in the HSM and validate its contents against another certificate that was stored in a key element (which might be held in the HSM as well)
- after parsing a certificate, read back parts of the certificate with `Csm_KeyElementGet()`. The part to be read was indicated by key element IDs (see [SWS_Csm_01022])

With the move of the APIs from Crypto Service Manager to Key Manager , the functionality described above is no longer available since R19-11. Only the storage of certificates in an HSM is still possible with the Key Manager.

This functionality will be reintroduced with upcoming releases. Meanwhile, if this functionality is needed for R20-11 we recommend to use the features specified in CSM from AUTOSAR Release 4.4 aligned with AUTOSAR release R20-11 architecture.

4.2 Specification of Crypto Service Manager (UID 402, SWS)

With AUTOSAR Release 4.4, job-based APIs for key management has been introduced (see chapter 8.3.9: "Cryptographic Primitives and Schemes").
These introduced functions have a parameter for the key identifier. However, they use at the same time an identifier to a job, that itself allows key assignment in the job configuration. Which key to be used, the key identifier configured in the job or the key identifier from the API, is not clear.
This unclear situation will be addressed and discussed inside AUTOSAR and clarification will be provided in an upcoming release.

4.3 Specification of Diagnostic Log and Trace (UID 351, SWS)

VFB Tracing: Currently, VFB Trace only supports the non-verbose mode. I.e., the Dlt module will send out the arguments in a raw format, simply doing a memory copy of the arguments to the trace message.

Note: Currently, the Dlt data type model does NOT support arbitrarily nested complex data types, which AUTOSAR does. So there is no generic way to transform arguments given to the VFB Trace hook functions into Dlt data types needed for the verbose mode.

Also an ASAM Fibex description cannot be generated by the Dlt module as the in-memory representation might not be compliant to the SWCD data type description of the arguments.

4.4 Specification of TCP/IP Stack (UID 617, SWS)

This document does not cover the assignment of UDP or TCP port numbers. There is no reserved space within the IANA assigned number range. Each implementer is responsible for managing the used port numbers.

This document does not cover the management of IP addresses. This might be done dynamically, e.g. by using DHCP, or statically. It is the implementer's responsibility to prevent address conflicts and achieve compliance with IANA address assignments.

This specification does not prescribe a certain physical layer or data rate.

Although a CDD interface is specified, allowing additional upper layer modules, a fan-out of one socket to multiple upper layer modules is not intended to be supported.

The AUTOSAR TLS implementation has the following limitations:

- A TLS implementation shall not support data compression or decompression.
- Session renegotiation shall not be supported.
- No support for secure connection over UDP (e.g. for DTLS)

- No support of FQDN
- No client Hello padding extension IETF RFC7685
- No session hash and extended master secret IETF RFC 7627
- No support for TLS versions lower than 1.2.
- No support for dynamic “downgrading” of a TCP connection with an established TLS connection to a plain TCP connection (without TLS)
- Static TLS connection assignment is bound to the port configuration of the server. Thus, using different TLS settings for different connections (possibly originating from different clients) to the same server port is not possible.

The AUTOSAR IPsec implementation has the following limitations:

- IPsec in “tunnel mode” is not supported right now. Transport mode only.
- IPv6 is not supported
- Multicast is not supported

4.5 Specification of Software Cluster Connection module (UID 974, SWS)

The specification currently supports a limited number of BSW modules. In addition, the available VFB communication features are restricted.

4.6 Specification of CAN State Manager (UID 253, SWS)

The CanSM module can be used for CAN communication only. Its task is to operate with the CanIf module to control one or multiple underlying CAN Controllers and CAN Transceiver Drivers. Other protocols than CAN (i.e. LIN or FlexRay) are not supported.

4.7 Specification of ECU State Manager (UID 78, SWS)

ECUs cannot always be switched off (i.e. zero power consumption).

4.8 Specification of Operating System (UID 34, SWS)

- The core AUTOSAR operating system assumes free access to hardware resources (e.g. registers), which are managed by the OS itself.

- Specific (extended) features of the core operating system extend the requirements on hardware resource (e..g memory protection or timing protection)
- In general hardware failures in the processor are not detected by the operating system.

4.9 Guide to BSW Distribution (UID 631, EXP)

The support for Basic Software Allocation in AUTOSAR is currently limited to backward compatible changes (w.r.t. AUTOSAR 4.0.3). This currently results in the following restrictions, which may not apply to future releases of AUTOSAR:

- There is only one QM BSW partition per core.
- Communication between master and satellites is not standardized.
- BSW functional clusters and their AUTOSAR BSW Cluster Interface are not standardized.

4.10 Specification of LIN Interface (UID 73, SWS)

The LIN Interface module (LinIf) supports ISO 17987 series of standards (which also covers ISO 14229-7:2015, SAE J2602 and the behavior of previous versions of LIN specifications: LIN 2.2, LIN 2.1, LIN 2.0 and LIN 1.3 by LIN Consortium as far as they are identical to ISO 17987, see Annex B of ISO 17987-3 for compatibility information), but with some limitations, e.g.

- Following services are not supported by the LinIf:
  - ConditionalChangeNAD (SID 0xB3, defined in the LIN 2.1 specification; obsolete in ISO 17987-3)
  - DataDump (SID 0xB4, optional in ISO 17987-3)
  - Transmission of Reserved Frames (defined in the LIN 2.1 specification)
- Following services are not supported by the LinIf, for LIN Slave Nodes:
  - ReadByIdentifier with identifier unequal to 0 and 2 (SID 0xB2, mandatory in ISO 17987-3)
  - the Serial Number (defined in the ISO 17987-3, clause 6.2.2). It means that there's no corresponding configuration nor API for accessing Serial Number
  - AutoAddressingSlave (SID 0xB8, optional in ISO 17987-3), Slave node position detection (SID 0xB5, optional in LIN 2.x specification)
- Following services are not supported by the LinIf, for LIN Master Nodes:
  - ReadByIdentifier (SID 0xB2, mandatory in ISO 17987-3)

The Specification of LIN Interface (SWS LinIf) defines the Behavior of LIN Schedule Table Manager which is not defined in the ISO 17987 series of standards. But if LinTp-ScheduleChangeDiag was set to TRUE, simultaneous Schedule Table Switch requests originated from LinTp and from Non-LinTp (BswM or CDD) must be avoided, to prevent premature terminations of diagnostic connections. This issue will be fixed in next release(s).

4.11 Specification of Watchdog Manager (UID 80, SWS)

- There're many long-lasting limitations for the Watchdog Manager module (WdgM). For details, see chap. 4 of SWS WdgM.
- Uptraces: AUTOSAR Foundation documents related to Health Monitoring (which are to be applicable to both AP PHM and CP WdgM) are not referred yet.
- New concept System Health Monitor is not incorporated to CP WdgM yet.
- Behavior of MainFunciton-based Supervision Algorithms (Alive, Timeout part of Deadline) right after Mode Switch requires further clarification.
- Behavior of Partition Restart / Shutdown requires further clarification (incl. restart of statemachines of Global/Local Supervision Status).

4.12 Specification of Function Inhibition Manager (UID 82, SWS)

Timing constrains have to be considered for the whole system. Note that the process and response times strongly depend on the implementation of the FiM module. Hence, if there are explicit needs for faster responses of the FiM than the cycle (time slice of the task) these needs have to be considered by the FiM implementation specifically by the affected application. Special measures have to be implemented by the FiM which are not explicitly specified in this AUTOSAR document, since here, the implementation is - on purpose - not prescribed.

The FiM shall compute the permission of a FID independently of the state of other FIDs.

Interdependencies between FIDs are not supported by the FiM. That means an FID does not influence another FID.

4.13 Specification of FlexRay State Manager (UID 254, SWS)

This specification only defines the straightforward case for starting and stopping the communication on a FlexRay cluster.

For the case of multiple CC of one ECU assigned to one FlexRay cluster some items are left open for the implementation:

- Which CC is used to transmit the wakeup pattern
- Handling of inconsistent POC states in the CCs

4.14 Specification of Network Management Interface (UID 228, SWS)

- The Generic Network Management Interface can only be applied to communication systems that support broadcast communication and 'bus-sleep mode'.
- There is only one instance of the Generic Network Management Interface layer for all NM-Clusters. This instance manages all channels where a NM is used.
- The Generic Network Management Interface shall only include the common modes, definitions and return values of different bus specific NM layers.
- If 10BASE-T1S is used in combination with PLCA media access, then Nm Coordinator functionality are not supported. Note: Consequently, the configuration parameter NmCoordinatorSupportEnabled shall be set to false.

4.15 Specification of Ethernet Switch Driver (UID 656, SWS)

The Ethernet Switch Driver module is only able to handle a single thread of execution. The execution must not be pre-empted by itself. The implementation is limited to 10Mbit/s, 100MBit/s and 1000Mbit/s Ethernet and transceivers connected via (gigabit) Media Independent Interface (xMII). Depending on the Ethernet hardware, it may become necessary that implementations deviate from API specifications in respect to the asynchronous/synchronous behavior. The switch driver does not support the following features:

- MAC-based Ingress Filtering: No filtering options for Ethernet frames based on MAC-addresses is supported

4.16 Specification of Time Synchronization over FlexRay (UID 675, SWS)

- Time Masters, Time Gateways and Time Slaves shall work with a Time Base reference clock with a worst-case accuracy of 2μs.
- The Time Base in the OFS messages is limited to 32 bit, wherefore the maximum supported time value is 4294967295 seconds (232-1).

- “CRC secured” in the context of this document refers to CRC integrity protection mechanism and does not imply that CRC is used as a cybersecurity solution.

4.17 Specification of Time Synchronization over CAN (UID 674, SWS)

- The current version of CanTSyn does not support hardware timestamping capabilities.
  - The first consequence is that the Time Synchronization is less accurate due to Rx-/Tx-ISR latencies and execution time until the Virtual Local Time is retrieved.
  - The second consequence is the need of not nested interrupts in the CAN driver for the Global Time PDUs (i.e., it is strongly recommended not to invoke the TX confirmation and RX indication functions in polling mode).
- The Time Base in the SYNC and OFS messages is limited to 32 bit, wherefore the maximum supported time value is 4294967295 seconds (232-1).
- Time Masters, Time Gateways and Time Slaves shall work with a Time Base reference clock with a worst-case accuracy of 2μs.
- “CRC secured” in the context of this document refers to CRC integrity protection mechanism and does not imply that CRC is used as a cybersecurity solution.

4.18 Specification of Synchronized Time-Base Manager (UID 420, SWS)

- OS ScheduleTable
  The Synchronized Time-Base Manager shall perform the functionality of synchronizing OS ScheduleTables with a respective Synchronized Time Base. However, the StbM considers only the case when the targeted OS ScheduleTable is explicitly synchronized. The implicit synchronization does not affect the StbM, because the synchronization mechanism bypasses the module (for more information about the difference between explicit and implicit synchronization, please refer to [1]). Thus, when talking in the following about synchronization of OS ScheduleTables, always the explicit one is meant.
- Synchronized Time Base Identifier
  The StbMSynchronizedTimeBaseIdentifier range (128 .. 65535) is currently reserved and might still be used by legacy applications (implementing Triggered Customers). The ID range will however be reassigned to new features in the next release. Legacy applications will then no longer be supported.

- Mode switches
  The Synchronized Time-Base Manager does not deal with mode switches during runtime.

- Configuration
  Postbuild configuration of the StbM is limited to enabling or disabling the functionality of a system wide Global Time Master for a Time Base (refer to [ECUC_-StbM_00036]).

- Out of scope
  - Errors, which occurred during Global Time establishment and which are not caused by the module itself (e.g. loss of FlexRay global time is a FlexRay issue and is not an issue of the Synchronized Time-Base Manager).
  - Errors, which occurred during interaction with customers.
    Example: Calling the explicit OS ScheduleTable synchronization may cause an exception, because the delta between the submitted parameter “counterValue” and the OS internal counter is higher than the tolerance range of affected expiry points. Dealing with this exception is an OS issue, not an issue of the Synchronized Time-Base Manager.

4.19 Specification of ADC Driver (UID 10, SWS)

Power State Control APIs are implementable only if the MCAL driver owns the complete underlying HW peripheral i.e. the HW peripheral is not accessed by other MCAL modules.

4.20 Specification of CAN Network Management (UID 13, SWS)

- One channel of CanNm is associated with only one network management cluster in one network. One network management cluster can have only one channel of CanNm in one node.
- One channel of CanNm is associated with only one network within the same ECU.
- CanNm is only applicable for CAN systems.

4.21 Specification of Bus Mirroring (UID 873, SWS)

The Bus Mirroring module cannot be used to influence the traffic on one of the buses configured as a source bus. To ensure this and to avoid loop-back of messages leading to bus overload, the generation tool shall ensure that no bus is connected to the Bus Mirroring module both as source and destination bus (see [SWS_Mirror_00001]).

The Bus Mirroring module is controlled by a diagnostic control application through the dedicated (service) API listed in chapter 8. The control functionality is made accessible to a diagnostic tester by special diagnostic services, which are handled by the DCM and implemented by the diagnostic control application. The DCM provides the necessary security to exclude inadvertent activation of the Bus Mirroring. The Bus Mirroring module does not provide another control interface, and it does not receive control messages on the destination bus.

In general, the Bus Mirroring module does not support source buses that have a larger frame size or more additional information than the destination bus can carry, e.g. CAN-FD to CAN, CAN to LIN, FlexRay to CAN, Ethernet to CAN, or Ethernet to FlexRay. The Bus Mirroring module does not fragment mirrored frames.

The Bus Mirroring module will only mirror traffic that is actually received or transmitted by the bus interface modules. For CAN this means that besides the transmitted frames only those data frames that pass the hardware filter will be mirrored, and that remote frames and error frames will not be mirrored. For LIN, slave-to-slave communication will not be mirrored by a LIN master. And for FlexRay, only transmitted frames and those received frames for which reception buffers are assigned (possibly as a FIFO) will be mirrored.

Another limitation of the mirroring from a FlexRay source bus concerns the reported time stamps and cycles. The Timestamp reported for a FlexRay frame contains the time when the corresponding job list entry was executed. The actual transmission time has to be calculated from the slot ID contained in the reported FrameID. The cycle contained in the reported FrameID is accurate only for received frames and frames transmitted in the static segment. For frames transmitted in the dynamic segment, the reported cycle can be inaccurate because it can happen that a frame cannot be transmitted in the expected cycle, it is then deferred to the next suitable cycle.

A re-serialization of received serialized frames shall not be done by the Bus Mirroring module, because that would require too much resources. Instead, the serialized PDUs shall be routed directly to the destination bus.

The Bus Mirroring module will also not support the forwarding from Ethernet to Ethernet. This use case is already covered by the Port Mirroring feature of the AUTOSAR Ethernet Switch Driver.

4.22 Specification of a Diagnostic Communication Manager for SAE J1939 (UID 610, SWS)

The J1939 Diagnostic Communication Manager implements only a subset of 'Diagnostic messages' as defined in Table 1: Supported DMx messages.

The DM13 does not support “Suspend Signal” “Suspend Duration”.

NACK is not provided for received DMx messages that are not supported or not configured. This restriction mainly affects handling of DM07 and DM13.

4.23 Specification of Network Management for SAE J1939 (UID 612, SWS)

The J1939 Network Management module does not support all features defined in [2], especially:

- Changing the address of a node after reception of CommandedAddress or after an address loss.
- Changing the NAME of a node using the Name Management protocol.
- Detection of address violations by messages other than AddressClaimed.

4.24 Specification of a Request Manager for SAE J1939 (UID 611, SWS)

The J1939 Request Manager only implements Request, Request2, and Acknowledgement PGs. It does not provide support for the Transfer PG.

4.25 Specification of a Transport Layer for SAE J1939 (UID 425, SWS)

The AUTOSAR architecture contains several communication system specific transport layers (J1939Tp, CanTp, FrTp, etc.). All of these modules need to have identical APIs, with the exception of API functions for which the PduR has separate configuration abilities.

The J1939Tp module does not implement the TriggerTransmit API, because it is only needed for time triggered bus architectures.

4.26 Specification of PDU Router (UID 35, SWS)

The PDU Router module does not:

- have mechanisms for signal extraction or conversion,
- have mechanisms for data integrity checking (like checksums).
- change or modify the I-PDU,
- make any PDU payload dependent routing decisions,
- support routing between TP modules and communication interface modules or vice versa,

- support routing of I-PDUs between communication interface modules with rate conversion. (This functionality will be supported in cooperation with an upper layer module, e.g. the Com module).

The PDU Router module supports fan-out of I-PDUs transmitted from a local module (e.g. Com module). There are some limitations if the I-PDU shall be transmitted to more than one destination (i.e. fan-out 1:n; n>1), because the upper layer module is not aware how many destinations there are:

- The PDU Router reports E_OK for a transmit request from an upper layer if at least one destination lower layer reports E_OK.
- The PDU Router gives a transmit confirmation to the upper layer when it receives the last transmit confirmation from destination lower layer.
- The PDU Router returns E_OK for a transmit cancellation requested from the upper layer only if all destination lower layers return E_OK.

If the I-PDU fan-out is performed by the PDU Router, this has further consequences for the Com as upper layer module:

- Update bits will not work
- I-PDU sequence counter will not work
- The Tx confirmation of the communication interface API will be handled in the way that the local module (e.g. Com module) will be informed when the last destination has confirmed the transmission. This means that deadline monitoring is made with respect to the last Tx confirmation (i.e. there is no difference if all the I-PDUs were transmitted successfully or not).
- Starting and stopping of I-PDU groups affects all destinations.

Note that above limitations are not set as requirements since they do not concern functionality provided by the PduR module. But implication of the use of the PduR module will affect these functionalities.

4.27 Specification of Ethernet Interface (UID 417, SWS)

The Ethernet Interface is conceptually able to access one or more Ethernet Driver and one or more Ethernet Transceiver Driver.
It is not possible to transmit data which exceeds the available buffer size of the used Ethernet controller. Longer data has to be transmitted using the Internet Protocol (IP) or Transmission Control Protocol (TCP).

4.28 Specification of FlexRay Interface (UID 27, SWS)

The FlexRay BSW modules are only able to handle a single thread of execution per Cluster. The execution for a particular Cluster must not be pre-empted by itself for the same Cluster. The same applies to the execution of the FlexRay Job List Execution Function.

It is not possible to transmit signals, PDUs, and/or L-SDUs, which exceed the available buffer size of the used FlexRay CC during normal operation. Longer signals, PDUs, and/or L-SDUs have to be transmitted using the FlexRay Transport Protocol.

Note: The FlexRay Interface does not make any PDU payload-dependent routing decisions.

Note: In order for the AUTOSAR FlexRay BSW (FrIf and FlexRay Driver) modules to be able to control a FlexRay CC, this CC must allow for configuring its transmit/receive buffers to support the Cycle Counter Filter Criterion / (Support of Slot/Cycle Muliplexing).

For 2.1 FlexRay Hardware, the following Cycle Counter Filtering is possible

Cycle Number = (B + n * 2R)mod64

with exactly one tuple of values for B and 2R, where:

- Base Cycle **B** $\in$ [0 .. 63]
- Cycle Repetition **2R** ; R $\in$ [0 .. 6]
- Variable **n** = 0 .. 63
- **B < 2R**

For 3.0 FlexRay Hardware, the Cycle Counter Filtering shall be possible as described in FlexRay Communications System Protocol Specification V3.0.

4.29 Specification of SW-C End-to-End Communication Protection Library (UID 428, SWS)

If the E2E Library is invoked from SW-Cs or from E2E Protection Wrapper, then the communication shall be an explicit sender-receiver communication, in 1:1 and 1:N multiplicities.

If the E2E Library is invoked at the level of data elements and 1:N communication model is used and the data elements are sent using more than one I-PDU, then all these I-PDUs shall have the same layout.

For each 1:N sender-receiver relationship the user of AUTOSAR shall define one specific layout to which the data elements that are going to be protected by E2E-Library are mapped for data transmission.

In case a user of AUTOSAR needs protected intra-ECU communication and protected inter-ECU communication to implement a safety-related sender-receiver relationship, the defined inter-ECU communication I-PDU layout shall be used for both transmissions.

If a user of AUTOSAR needs a protected intra-ECU communication to implement a safety-related sender-receiver relationship, then a specific layout (not restricted to the needs of COM I-PDUs) can be defined and used.

Currently AUTOSAR does not provide the functionality to describe and handle more than one layout for the same data element (e.g. within the RTE) by using different protection mechanisms depending on Intra-ECU and Inter-ECU communication. Thus, for a 1:N sender-receiver relationship the user of E2E-Library is responsible to select one appropriate layout for the to be protected data elements. E.g. for a 1:N sender-receiver relationship the COM I-PDU layout can be used for the transmission of data elements protected by E2E-Library to receivers located within and without the ECU.

4.30 Specification of Module E2E Transformer (UID 650, SWS)

The current solution for E2E communication protection is based on some use case constraints. Possibly, they will be reduced or they will be removed in future document versions:

- For sender/receiver communication only cyclic communication of/between Software Components is considered (no event-based)
- Only inter-ECU communication is considered (communication that is exchanged over COM stack)
- Only non-blocking characteristics of queued sender-receiver communication is considered (blocking characteristic for queued communication is not supported)
- No New Data checks are only supported for polling-based Client-Server Communication and Sender-Receiver Communication
- E2E Profiles 1 and 2 do not support Client-Server Communication at all

Please note that these constraints do not necessarily limit the use of the E2E Transformer to use cases meeting these constraints.

Further, the following limitations are known: Error reporting to DEM not yet specified.

Cyclic invocation is needed because the `E2E_Check` is responsible (among others) to detect losses and delays, so in case of loss or delay, the `E2E_Check` is not invoked, resulting with potentially lost error detection. The cyclic invocation of E2E Transformer is ensured by cyclic invocation of corresponding RTE functions by Software Components.

The E2E protection for methods is limited to idempotent functions.

4.31 Specification of FlexRay Driver (UID 26, SWS)

In the dynamic segment of each FlexRay Communication Cycle, a transmit/receive buffer of a FlexRay Communication Controller shall be used only one particular LPdu. This limits the reconfiguration possibilities and thus restricts the number of transmittable (sent and received) LPdus per dynamic segment to the accumulated number (over all CCs on one ECU) of transmit/receive buffers connected to one cluster. This limitation results from the unpredictability of the time of transmission of an LPdu within the dynamic segment. Because of that a point in time for the reconfiguration of a certain buffer for multiple usages within the dynamic segment cannot be predetermined.

4.32 Specification of Socket Adaptor (UID 416, SWS)

The transmission of data using TCP/IP over Ethernet requires about 60 bytes of header information. This implies that for small messages the header overhead may reach an unacceptably high percentage.

To avoid further protocol overhead, the use of a single socket connection per PDU is described here. However, this solution is very resource intensive, particularly if many small PDUs are to be transmitted. One solution described here as an option is to add a small PDU header, containing an ID and length information. This enables transmission of multiple PDUs via one socket connection. Additionaly a resource conservation scheme is included in this specification as an option.

This document does not cover the assignment of UDP or TCP port numbers. There is no reserved space within the IANA assigned number range. Each implementer is responsible for managing the used port numbers.

This document does not cover the management of IP addresses. This might be done dynamically, e.g. by using DHCP, or statically. It is the implementers responsibility to prevent address conflicts and achieve compliance with IANA address assignments.

This specification does not prescribe a certain physical layer or data rate.
SOME/IP Protocol Specification [3] specifies to check Protocol Version prior to the check of Service ID and Method ID to be valid. SoAd does not check Protocol Version. Independent of future version It will always interpret the first 4 bytes as message ID (header ID in this document) and the second 4 bytes as payload length.
SOME/IP Protocol Specification [3] specifies to check Interface Version prior to the check of Method ID to be valid. SoAd does not check Interface Version. SoAd performs routing to configured Message IDs independent of the contained Interface Version on dedicated routing paths.
SOME/IP Protocol Specification [3] specifies to check Service ID and Method ID to be valid. Invalid IDs shall be responded with corresponding error codes.
SoAd checks the SOME/IP message ID (header ID in this document) which is a combination of service ID and method ID. If the ID is not valid the module is not capable to respond with corresponding error codes. Instead, runtime error `SOAD_E_INV_PDUHEADER_ID` is raised.

4.33 Specification of LIN Driver (UID 72, SWS)

Only one LIN channel of an ECU is allowed to connect to a particular LIN cluster. Unless there are unused (not connected) channels in the ECU, the number of LIN channels is equal to the number of LIN clusters.

Driver scope

- One LIN driver provides access to one LIN hardware unit type (simple UART or dedicated LIN hardware) that may consist of several LIN channels.
- For different LIN hardware units a separate LIN driver needs to be implemented. It is up to the implementer to adapt the driver to the different instances of similar LIN channels.
- In case several LIN driver instances (of same or different vendor) are implemented in one ECU the file names, API names, and published parameters must be modified such that no two definitions with the same name are generated. The name shall be extended according to [SRS_BSW_00347] with a Vendor Id (needed to distinguish LIN drivers from different vendors) and a Vendor specific name (needed to distinguish different hardware units implemented by one Vendor): <Module abbreviation>_<Vendor Id>_<Vendor specific name>.

The LIN Interface is responsible for calling the correct function. The necessary information shall be given in an XML file during configuration. See [4] for description how the LIN Interface handles several LIN drivers.

4.34 Specification of Wireless Ethernet Driver (UID 798, SWS)

- It is not possible to transmit data which exceeds the available buffer size of the used controller.
- AUTOSAR supports currently only wireless communication using IEEE 802.11p. Other 802.11 standards (e.g. for infrastructure networks and integration with TCP/IP) can be extended in future releases of the AUTOSAR standard.
- The V2X modules follow the guidance regarding the Day-1 scenarios defined by the Basic System Standards Profile from Car-2-Car-Consortium.
- AUTOSAR R20-11 only focuses on the European version of car-to-car communication as defined by ETSI. Extension to other regions are planned for future releases of the AUTOSAR standard.
- The Microcontroller Abstraction Layer Multi-Core Distribution Concept is implemented as “draft” in this software specification. Refer to chapter 10 for more information.

4.35 Specification of Wireless Ethernet Transceiver Driver (UID 799, SWS)

The Microcontroller Abstraction Layer Multi-Core Distribution Concept is implemented as “draft” in this software specification. Refer to chapter 10 for more information.

4.36 Specification of Ethernet State Manager (UID 415, SWS)

The EthSM can be used for Ethernet communication only. Its dedication is to operate with the EthIf to control one or multiple underlying Ethernet Controllers and Ethernet Transceiver Drivers. Other protocols than Ethernet (i.e. CAN, LIN or FlexRay) are not supported.

The following items are not supported by the current version of this specification:

- Wake on LAN. (Please note, wake-up and sleep on dataline according to the OA TC10 specification (see [5]) is supported)

The actual EthSM requires an IP-based communication stack. To get `FULL_COMMUNICATION` it is necessary to get an active IP communication. In further specifications, an alternative “low level” state machine will be introduced. This state machine only works on driver/transceiver level (without IP communication). This is necessary to realize other communication protocols (e.g. IEEE 1722).

4.37 Specification of Large Data COM (UID 326, SWS)

Efficient COM supports communication of linear opaque byte wise data in a very resource-saving way. It does so by skipping all functionality not required for event based non-cyclic communication.

Efficient COM does not apply any changes like for instance endianness conversion to the data it transports.

Prerequisites for usage of Efficient COM:

- PDU contains only 1 Signal and no ISignalGroup
- The Signal is of type byte array with either fixed or dynamic length
- Transmission mode is either triggered or triggered without repetition
- Transmission mode selection is not used
- No update bit is used
- No minimum delay time is used
- No timeout supervision is used

- No byte order conversion is used
- No Rx/Tx Filtering
- No Signal Invalidation

4.38 Specification of MCU Driver (UID 31, SWS)

In general the activation and configuration of MCU reduced power mode is not mandatory within AUTOSAR standardization.

Enabling/disabling of the ECU or μC power supply is not the task of the MCU driver. This is to be handled by the upper layer.

4.39 Specification of Flash EEPROM Emulation (UID 286, SWS)

The synchronization of a potential parallel access (e.g. BulkNvDataManager) to the underlying flash driver is not part of this AUTOSAR release.

4.40 Specification of Ethernet Driver (UID 430, SWS)

The Ethernet Driver module is only able to handle a single thread of execution. The execution must not be preempted by itself.
It is not possible to transmit data which exceeds the available buffer size of the used controller. Longer data has to be transmitted using the Internet Protocol (IP) or Transmission Control Protocol (TCP).
Depending on the Ethernet hardware, it may become necessary that implementations deviate from API specifications in respect to the asynchronous/synchronous behaviour.

4.41 Specification of Ethernet Transceiver Driver (UID 431, SWS)

The Ethernet Transceiver Driver module is only able to handle a single thread of execution. The execution must not be preempted by itself.

5 Release history

5.1 Release R20-11

Name	Specification history entry
Classic Platform Release Overview	• Release Life Cycle Status: R20-11 is in Evolution, R20-11 supersedes R19-11
Requirements on CAN	• Editorial changes
Requirements on Communication	• No content changes
Requirements on Network Management	• Refined PNC related ComM and NM handling
Requirements on FlexRay	• No content changes
Requirements on Gateway	• No content changes
Requirements on Memory Services	• No content changes
Requirements on Operating System	• Added ARTI requirements
General Requirements on SPAL	• No content changes
Specification of ADC Driver	• Error classification tables updated
Specification of CAN Driver	• Removed Pretended Networking • CanDrv_CONSTR_00512 was added • Updated ECUC_Can_00471 descripton • Add new parameter: CanObjectPayloadLength • A note was added to SWS_Can_00403 • SWS_Can_00222 was changed • Minor corrections / clarifications / editorial changes; • Added Reporting of CAN Error Types chapter. Requirement SWS_Can_91021 was added. • CanEnableSecurityEventReporting container was added
Specification of CAN Interface	• Introduction of Security Events • Rework of Error classification chapters • Removal of pretended networking • Minor corrections/clarifications
Specification of CAN Network Management	• Harmonizing error sections • Partial Networking extensions introduced
Specification of CAN Transport Layer	• Improve Error sections • Clarifications
Specification of Communication	• BSW Multicore Distribution (CONC_643) • minor corrections / clarifications / editorial changes
Specification of CRC Routines	• Improved the structure of the "error sections"
Specification of Default Error Tracer	• Editorial Changes
Specification of Diagnostic Communication Manager	• Incorporation of Concept 671 Intrusion Detection System Manager • Added DcmDspExternalSRDataElementClass • Updated the Error Classification chapter • minor corrections / clarifications /editorial changes; For details please refer to the ChangeDocumentation

▽

△

Name	Specification history entry
Specification of Diagnostic Event Manager	• Referenced DemEventFailureCycleCounterThreshold removed • Remove Dem_J1939DcmClearDTC from SWS_Dem_01101 • Add Dem_GetDTCSelectionResultForClearDTC to the list of API's in SWS_Dem_01253 • Minor corrections / clarifications / editorial changes
Specification of DIO Driver	• Cleaned error classification
Specification of EEPROM Driver	• Editorial Changes
Specification of ICU Driver	• Error table cleanup in Error classification • Removed "7.y Error Detection"' • Moved SWS_Icu_00022 to new requirement number in "8.3 Function definitions"
Specification of Flash Driver	• Editorial changes
Specification of FlexRay Driver	• Reworked chapter "error classification" • Changed exposure of Fr_ConfigType
Specification of FlexRay Interface	• Editorial changes
Specification of FlexRay Network Management	• Chapter 7.12 Error detection removed • Chapter 7.13 Error notification removed. • SWS_FrNm_00056, SWS_FrNm_00057, SWS_FrNm_00051 moved to Chapter 8. • Specification for synchronized PNC shutdown added. • Configurable interfaces moved from Chapter 8.6.2 to 8.6.3 • PRS_NetworkManagementProtocol added as input document.
Specification of GPT Driver	• Delete requirement Sws_Gpt_00270 • Replace requirements defined for each error by global requirement for each error table defined in chapter 7.4. • Move chapter Error detection in chapter 8
Specification of MCU Driver	• Enum and Error related modifications • Editorial Changes
Specification of NVRAM Manager	• InitBlockCallback and ROM block are mutually exclusive • Removal for DET error NVM_E_PARAM_BLOCK_TYPE • NvM partitioning for multi-core
Specification of Operating System	• Updates to ARTI description and configuration • Ioc: correction regarding N:M communication • Minor correction / clarification / editorial changes

▽

△

Name	Specification history entry
Specification of PDU Router	• Description of and requirements related to PduRRoutingPathGroup has been updated • Cancel Transmit for gateways has been clarified • Error classification has been harmonized • Structure of "Error Section" has been improved • "Draft" tags have been removed from Multicore Distribution spec. items
Specification of PWM Driver	• Minor corrections in section Error Classification
Specification of SPI Handler / Driver	• Error sections refactored • New configuration parameters: SWS_Spi_00247, SWS_Spi_00248 • Removed requirements: SWS_Spi_00008, SWS_Spi_00009, SWS_Spi_00010, SWS_Spi_00063 and SWS_Spi_00064 • Chapter 8.2: enumeration types have the values specified
Specification of Watchdog Driver	• Editorial - reshaped "Error classification" chapter
Specification of Port Driver	• Added Enum values for Port_PinDirectionType • Editorial changes
Specification of Watchdog Interface	• Updated the structure and tables of the error sections
Requirements on LIN	• No content changes
General Requirements on Basic Software Modules	• New naming convention (SRS_BSW_00494) • Introduced IDSM concept (SRS_BSW_00488 - SRS_BSW_00493) • Clarification about no return after development error (SRS_BSW_00369) • Updated enum examples with values (SRS_BSW_00441, SRS_BSW_00377)
Specification of I/O Hardware Abstraction	• No content changes
Specification of Platform Types	• Chapter 7.6 "Error classification added" • "VoidPtr" and "ConstVoidPtr" added • Document converted from Word to LaTeX
Specification of Standard Types	• Fixed Design issues with E2E communication protection for methods • Added TransformerError and TransformerForward • Fixed missing Type definitions • Editorial Changes
Specification of Communication Stack Types	• Removed IcomConfigIdType and IcomSwitch_ErrorType from Type definitions
Specification of Compiler Abstraction	• Editorial changes
Layered Software Architecture	• Removed Pretended Networking • Added caveats for E2E Protection Wrapper • Layer Interaction Matrix: Allow Crypto Driver to access Memory Services • Incorporated new concepts for Intrusion Detection System Manager, CP Software Clusters

▽

△

Name	Specification history entry
Virtual Functional Bus	• Added docproperty ConfidentialityInformation • Fixed topmost table on frontpage (removed additional column)
Specification of ECU Resource Template	• No content changes
Software Component Template	• In-place Strategy for Array Data Type • New Service Use Cases • Optimization of Return Type for RTE APIs • minor corrections / clarifications / editorial changes
System Template	• Added support for 10-BASE-T1S • Added support for Software Clusters • Improved RTE Fan-in and RTE Fan-out description • Introduced modeling approach for Service Discovery Service Interfaces on VFB level
Methodology	• Minor corrections and editorial changes
Requirements on Mode Management	• Concept "VNSM (Vehicle Network Stage Manager)" incorporated • Concept "EthernetWakeUpOnDataLine" incorporated • Concept "Rework of PNC related ComM and NM handling" incorporated
Specification of CAN Transceiver Driver	• Modeling of Development Errors, Runtime Errors, and Transient Faults. • SOME/IP transformation props missing is added. • Clean up of APIs with return type void, that specify a return value. • CanTrcv Operation Mode Inconsistencies corrected.
Specification of LIN Driver	• Modified SWS_Lin_00266, Lin_GeneralTypes removed • Editorial change
Specification of LIN Interface	• Corrected behavior of LinTp-originated schedule table switch (with limitations) and LinTp P2 timeout monitoring • Updated the structure and tables of the error sections • Corrected header filename for the imported type LinTrcv_TrcvModeType • Changed Service ID of LinTp_Transmit API • Reformulated negative requirements which are not testable
Specification of FlexRay Transceiver Driver	• Modeling of Development Errors, Runtime Errors, and Transient Faults • Improve the structure of the "error sections" of the SWS documents
Requirements on I/O Hardware Abstraction	• No content changes
Specification of RAM Test	• Updated the structure and tables of the error sections • Editorial changes

▽

△

Name	Specification history entry
Requirements on SPI Handler / Driver	• No content changes
Specification of ECU State Manager	• Corrected broken chapter structure • EcuM_UserType handling improved • Minor content changes, clarifications (multi-core, configuration, values)
Specification of Communication Manager	• Added optional feature for dynamic PNC-to-channel mapping • Added optional handling to transfer kind of communication request (either active or passive) to lower layers • Extend ComM service interface ComM_GetCurrentComMode, to obtain the PNC state of the mapped ComMUser • Added restriction for ComM users according the assignment of managed and managing channels
Specification of Watchdog Manager	• Clarified the meaning of thresholds WdgMDeadlineMin and WdgMDeadlineMax • Updated the structure and tables of the error sections • Editorial/Minor Corrections
Requirements on Function Inhibition Manager	• No content changes
Specification of Function Inhibition Manager	• requirement SWS_Fim_00010 degraded to explanatory description • requirement SWS_Fim_00062 removed
Requirements on Runtime Environment	• Added support for ClassicPlatfromFlexibility [SRS_Rte_00318] -[SRS_Rte_00321] • Added missing requirement for MetaData support [SRS_Rte_00322] • Data filtering on sender side
Specification of RTE Software	• Support for Software Clusters • Optimization of return type for RTE APIs • Minor corrections / clarifications / editorial changes
Requirements on ECU Configuration	• No content changes
Requirements on Basic Software Module Description Template	• No content changes
Specification of ECU Configuration	• Introduced CddModuleId parameter • Removed "Rules for Configuration Editors" chapter • Changed all lower multiplicities in the Ecuc meta-model to 0 and introduced constraints that define at which time which model elements need to be available. For details please refer to the ChangeDocumentation.
Basic Software Module Description Template	• Added Use-Cases • Editorial changes
Requirements on ADC Driver	• No content changes
Requirements on ICU Driver	• No content changes
Requirements on PWM Driver	• No content changes
Requirements on RAM Test	• Editorial changes

▽

△

Name	Specification history entry
Requirements on Memory Hardware Abstraction Layer	• No content changes
Modeling Guidelines of Basic Software EA UML Model	• described modeling of Generic Std_ReturnType Extension
Specification of Memory Mapping	• No content changes
List of Basic Software Modules	• No content changes
Requirements on I-PDU Multiplexer	• No content changes
Specification of I-PDU Multiplexer	• Improved the structure of the Error sections • Clarified IpduMRxDynamicPart lower multiplicity
Requirements on GPT Driver	• No content changes.
Requirements on DIO Driver	• No content changes
Requirements on EEPROM Driver	• No content changes
Requirements on Flash Driver	• No content changes
Requirements on MCU Driver	• No content changes
Requirements on Port Driver	• No content changes
Requirements on Watchdog Driver	• No content changes
SW-C and System Modeling Guide	• No content changes
Requirements on Free Running Timer	• No content changes
Requirements on Software Component Template	• No content changes
Requirements on System Template	• Added Software Cluster requirements; • For details please refer to the • ChangeDocumentation
Specification of Network Management Interface	• Support for synchronized PNC shutdown functionality • Introduction of Dynamic PNC-to-channel-mapping and PNC Learning algorithm • Added limitation regarding Nm Coordinator functionality when using 10BASE-T1S in combination with PLCA media access • Support for passing NM state change to SwC • Editorial changes
Requirements on ECU Resource Template	• No content changes
Specification of CAN State Manager	• Pretended Networking removed • Editorial changes
Specification of FlexRay State Manager	• Improved Error Handling Sections
Specification of LIN State Manager	• Cleanup error sections in chapter 7
Specification of LIN Transceiver Driver	• Cleaned error classification
Requirements on Core Test	• No content changes
Specification of Core Test	• Clean up of Error Classification chapter
Requirements on Flash Test	• No content changes
Specification of Flash Test	• Included Development Errors as artifact
Requirements on SW-C and System Modeling	• No content changes
Explanation of Application Interfaces of the Body and Comfort Domain	• No content changes
Explanation of Application Interfaces of the Powertrain Engine Domain	• Version update, No content changes

▽

△

Name	Specification history entry
Explanation of Application Interfaces of the Chassis Domain	• No content changes
Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	• No content changes
Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	• No content changes
Specification of Memory Abstraction Interface	• Chapter "7.1 Error classification" was reshaped
Specification of Flash EEPROM Emulation	• Fixed inconsistency in the example of SWS_Fee_00100 • Removed FEE_E_INIT_FAILED
Specification of EEPROM Abstraction	• EA_E_INIT_FAILED is removed
Requirements on AUTOSAR Features	• Added Intrusion Detection • Updated ISO 26262 version
Explanation of Interrupt Handling within AUTOSAR	• No content changes
Specification of Basic Software Mode Manager	• Added new notifications BswMNmStateChangeNotification and BswMSwitchAckNotification; Added new action BswMEthIfStartAllPorts • Added support for Ethernet switch port group switching • Removed support for pretended networking • Reinstated the EcuM<->BswM APIs: BswM_EcuM_CurrentState and BswM_EcuM_RequestedState
Requirements on Libraries	• No content changes
Specification of Diagnostic Log and Trace	• Added subcontainer and definition for parameter DltLogLevelThreshold and for DltGeneralNvRAMSupport. • Assigned new ID for Imported Types because of duplicated ID • Minor corrections and bugfixes • Editorial changes
Description of the AUTOSAR standard errors	• No content changes
Explanation of Error Handling on Application Level	• No content changes
Specification of Fixed Point Math Routines	• Chapter 7.1 Error sections updated
Specification of Fixed Point Interpolation Routines	• Chapter 7.1 Error sections were updated
Specification of Floating Point Math Routines	Added: • New function added Mfl_DivLim_f32 with the requirements SWS_Mfl_00844, SWS_Mfl_00845, SWS_Mfl_00846 and SWS_Mfl_00847. Modified: • Updated Chapter 7.1 Error classification.
Specification of Floating Point Interpolation Routines	• Chapter 7.1 Error sections updated
Specification of Bit Handling Routines	• Chapter 7.1 Error sections updated
Specification of Extended Fixed Point Routines	Added : • New function added Efx_Pt1Typ1Calc with the requirements SWS_Efx_00531, SWS_Efx_00532, SWS_Efx_00533, ▽

▽

△

Name	Specification history entry
	△ • SWS_Efx_00534, SWS_Efx_00535 • SWS_Efx_00823, SWS_Efx_00824 SWS_Efx_00825 - Requirements related to saturation has been added for the functions Efx_Add,Efx_Mul,Efx_Div Modified : • Updated Chapter 7.1 Error classification Removed : • Removal of Efx_Cast and Efx_Gt functions
Specification of Crypto Service Manager	• New feature: Save and Restore Context for running crypto operation • New feature: KeyGetStatus and KeySetInvalid • Updated Algorithm Families • Support of Multicore • Removing inconsistency in parameter names. • Editorial changes
Specification of Timing Extensions	• Added support for AUTOSAR Software Clusters and AUTOSAR Classic Platform Flexibility
Specification of Module XCP	• Improve the structure of the "error sections"
Specification of UDP Network Management	• Updates for CONC 641 VSNM • Updates for Light CONC 685 • Minor changes
Specification of Ethernet State Manager	• Reworked the statemachine specification of EthSM • Added handling for request of "COMM_FULL_COMMUNICATION_WITH_WAKEUP_REQUEST" • Cleaned up the document structure
Specification of Socket Adaptor	• Introduction of Security Event reporting • Added limitations for checking SomeIP protocol header. • Added state switch in context of SoAd_ReleaseRemoteAddr()
Specification of Ethernet Interface	• Introduction of Security Event reporting (DRAFT) • Introduction of EthernetWakeupOnDataline (DRAFT) • Introduction of 10BASE-T1S (DRAFT)
Specification of Diagnostic over IP	• Introduction of Security Event reporting (DRAFT) • Introduction of EthernetWakeupOnDataline (DRAFT) • Introduction of 10BASE-T1S (DRAFT)
Requirements on Ethernet Support in AUTOSAR	• Introduction of 10BASE-T1S (DRAFT) • Introduction of Ethernet Wake on data line (DRAFT)

▽

△

Name	Specification history entry
Specification of Synchronized Time-Base Manager	• Update for Time Validation feature – support for time gateways – ring buffer added to pDelay data • New interface Set/GetBusProtocolParam added to access bus specific protocol parameters • Attribute "Variation" of the R-Port MeasurementNotification and DET Error from StbM_BusGetCurrentTime() and StbM_BusSetGlobalTime() APIs corrected • Timesync definitions moved to RS_TimeSync
Specification of a Transport Layer for SAE J1939	• Improved structure of error sections • Replaced error descriptions with generated tables
Requirements on Crypto Stack	• Updated supported algorithms
Specification of SW-C End-to-End Communication Protection Library	• New profiles 4m,7m,08,44 • E2E for methods • Extension of E2E State Machine
Requirements on Module XCP	• No content changes
Specification of Ethernet Driver	• Eth_GeneralTypes removed from module list. • EthGetDropCountApi renamed to EthGetCounterValuesApi • Buffer handling • WakeOnDataLine • Details MII Read/Right for Clause 22
Specification of Ethernet Transceiver Driver	• EthTrcvGetTransceiverModeApi and EthTrcvSetTransceiverModeApi obsolete • Eth_GeneralTypes removed from imported module list • Introduction WakeOnDataLine • Introduction 10BASE-T1S
Specification of TTCAN Driver	• Editorial changes
Specification of TTCAN Interface	• No content changes
Guide to Mode Management	• Concept "EthernetWakeUpOnDataLine" incorporated • Updated PPorts, ProvidedModeDeclarationGroupPrototypes and Configurable ModeSwitchPorts section
Requirements on TTCAN	• No content changes
Application Interfaces User Guide	• Editorial changes • Set Document to obsolete, as methodology is implemented in AI-Tool incl. online help
Unique Names for Documentation, Measurement and Calibration: Modeling and Naming Aspects including Automatic Generation	• No content changes

▽

△

Name	Specification history entry
General Specification of Basic Software Modules	• Incorporation of Intrusion Detection System Manager concept • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Changed Document Status from Final to published
Specification of FlexRay ISO Transport Layer	• Removed SWS_FrTp_01132, SWS_FrTp_01111 • SWS_FrTp_01106 moved to Chapter 7.4 • Structure of Chapter 7.7 changed.
Specification of FlexRay AUTOSAR Transport Layer	• Chapter 7.7 Error detection removed • Chapter 7.8 Error notification removed • SWS_FrArTp_00291 moved to Chapter 8.
Utilization of Crypto Services	• No content changes
Specification of a Diagnostic Communication Manager for SAE J1939	• Unification of DM01 configuration • Improved structure of error sections • Replaced error descriptions with generated tables
Specification of a Request Manager for SAE J1939	• Improved structure of error sections • Replaced error descriptions with generated tables
Specification of Network Management for SAE J1939	• Improved structure of error sections • Replaced error descriptions with generated tables
Requirements on BSW Modules for SAE J1939	• No content changes
Requirements on OCU Driver	• No content changes
Specification of OCU Driver	• Added missing specification for enum values • Updated return value for Ocu_StartChannel • Minor changes in Error Tables
Specification of Service Discovery	• Alignments with Service Discovery Protocol specification • Several minor bugfixes • Editorial changes
Specification of TCP/IP Stack	• Introduction of IdsM (DRAFT) • Minor corrections and clarifications • Editorial changes
Complex Driver design and integration guideline	• Update for Module IDs for non-AUTOSAR BSW modules
Requirements on Time Service	• No content changes
Specification of Time Service	• Included Development Errors and Runtime Errors as artifact
Guide to BSW Distribution	• Added chapter on crypto-stack distribution
Safety Use Case Example	• No content changes
Specification of Module E2E Transformer	• Added Description of Profile 4m and 7m (E2E for methods) • Updated/added drawings of functions • Updated API Specification
Requirements on Secure Onboard Communication	• No content changes

▽

△

Name	Specification history entry
Specification of Secure Onboard Communication	• Added Security Events for IdsM • Added additional freshness value use case • Added separate Mainfunction container for multi core • Minor corrections / clarifications / • editorial changes; For details please refer to the Change Documentation • Changed Document Status from Final to published
Specification of Large Data COM	• Clean up error section • Changed Document Status from Final to published
Specification of Ethernet Switch Driver	• Support for Ethernet wake on data line • Modified description of return values in EthSwt_StoreConfiguration, EthSwt_ResetConfiguration • Updated the types for ReTaggingVlanId and DoubleTaggingVlanId • Fix service IDs for EthSwt_DeletePortMirrorConfiguration, EthSwt_PortLinkStateRequest, EthSwt_GetMaxFIFOBufferFillLevel • Editorial changes
Requirements on Transformer	• Editorial changes
General Specification of Transformers	• Fixed design issues with E2E communication protection for methods • Added Error Codes for E2E • Moved TransformerError and TransformerForward to SWS_StandardTypes • Editorial changes
Specification of SOME/IP Transformer	• Added call/response context to Client Server requirements • Constraint added for data type of length field of variable Strings • Added E_E2E Error to Table 7.11: Return Codes • Requirement added in case unvailability of optional member in the received serialized byte stream • Reworked E2E communication protection for methods • sizeOfStringLengthField introduced for the size of the length field for dynamic length strings • sizeOfArrayLengthField introduced for the size of the length field for variable size arrays • Fixed design issues with E2E communication protection for methods • Editorial Changes
Specification of COM Based Transformer	• No content changes
Integration of Franca IDL Software Component Descriptions	• No content changes
Overview of Functional Safety Measures in AUTOSAR	• No content changes

▽

△

Name	Specification history entry
Requirements on Safety Extensions	• Updated ISO 26262 references to the latest version
Specification of Safety Extensions	• Updated ISO 26262 references to the latest version • Added History of Constraints and Specification Items
Application Design Patterns Catalogue	• Signal quality states introduction • Extension of definition of electrical sensor interface
Diagnostic Extract Template	• Handling of Security Events • minor corrections / clarifications / editorial changes
Specification of Time Synchronization over CAN	• Time Validation updated for gateways • Time out handling of Synchronized and Offset Time messages corrected • Post build variant value corrected for CanTSynGlobalTimeMasterConfirmationHandleId and CanTSynGlobalTimeSlaveHandleId
Specification of Time Synchronization over FlexRay	• Time Validation updated for Time Gateways • Post build variant value corrected for FrTSynGlobalTimeMasterHandleId and FrTSynGlobalTimeSlaveHandleId
Specification of Time Synchronization over Ethernet	• Clarification of Follow_Up information TLV message • Clarification of Safety validation service interface • Sequence Counter specified • Improvement the structure of the Error classification • Clarification of EthTSynPortConfig
Requirements on Diagnostic Extract Template	• Add requirements for security event reporting
Supplementary material of general blueprints for AUTOSAR	• No content changes
Specification of Hardware Test Manager on start up and shutdown	• Chapter 7.3.1 generated from BSW UML model • Chapter 7.1 structure updated
Requirements on Hardware Test Manager on start up and shutdown	• No content changes
Modeling Show Cases Report	• No content changes
Requirements on Vehicle-2-X Communication	• Editorial changes
Specification of Vehicle-2-X Geo Networking	• Editorial changes
Specification of Vehicle-2-X Basic Transport	• Editorial changes
Specification of Vehicle-2-X Facilities	• Corrections of typos in chap. 8 definitions • Editorial Changes
Specification of Vehicle-2-X Management	• Editorial changes
Specification of Wireless Ethernet Driver	• Relaxed requirements on base address and memory alignment • References to the Ethernet Driver substituted by their content.
Specification of Wireless Ethernet Transceiver Driver	• Editorial changes
Specification and Integration of Hardware Test Management at start up and shutdown	• No content changes

▽

△

Name	Specification history entry
Specification of Crypto Interface	• Improve structure of error handling • Improve flexibility of crypto stack for multi core • Clarifications on requirements, API and configuration parameters • Add functionality for key status
Specification of Crypto Driver	• Minor corrections and editorial changes. • Support to save and restore workspace. • Add function to get and invalidate the overall key status. • Harmonize and extend crypto config • Consistent custom configuration for CryptoAlgorithmFamily and -Mode. • Key element handling in NVM.
Macro Encapsulation of Interpolation Calls	• No content changes
Specification on SOME/IP Transport Protocol	• Several minor bugfixes • Editorial changes
NV Data Handling Guideline	• Effective utilization of Block Fragmentation
Specification of Bus Mirroring	• Improved structure of error sections • Replaced error descriptions with generated tables • Multi-partition support finalized • Replaced Mirror_CanIdType and Mirror_FlexRayChannelType by native types
Requirements on Bus Mirroring	• No content changes
Explanatory Document for usage of AUTOSAR RunTimeInterface	• Introduced chapter Example Implementations • Updated chapter Example Configuration with ECUC changes
Specification of Key Manager	• Editorial changes, improve error section • Add security events for IdsM • Detail order of certificate verification • Align functions, parameters and return values for C-API and service interfaces • Signing request reference for CSR
Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components	• Changed document status from draft to valid
Specification of AUTOSAR Run-Time Interface	• Merged EcuC ArtiXxx containers into one Arti container • Added ARTI for RTE • Changed document status from draft to valid
Requirements on Firmware Over-The-Air	• Restructuring of requirements chapters • Rework of update dependencies • Refined Rollback procedure description • Editorial reworks

▽

△

Name	Specification history entry
Explanation of Firmware Over-The-Air	• Rework FOTA internal buffer handling • Refined FOTA state machine • Refined Rollback procedure description • Editorial reworks
Specification of Basic Software Multicore Library	• Improved the structure of the "error sections" of the SWS documents • CONC_643 "BSW Multicore Distribution" finalized
Specification of Bulk NvData Manager	• callback functions renamed • limitation added • specification item prefix adapted • editorial changes
Requirements on Software Cluster Connection module	• Intitial release
Specification of Software Cluster Connection module	• Intitial release
Explanation of CP Software Cluster Design And Integration Guideline	• Intitial release
Specification of Intrusion Detection System Manager	• Introduced policy enforcement in Adaptive Applications • Added IAM for Platform Health Management
Explanation of Application Interface of AD/ADAS vehicle motion control	• Intitial release

Table 5.1: Release History