Document Title	Classic Platform Release Overview
Document Category	Technical report (TR)
Document UID	0
Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Release Life Cycle Status: R26-11 is in Evolution, R26-11 supersedes R25-11

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

1 Introduction 6
1.1 Scope of This Document . . . 6
1.2 Terminology and Licenses . . . 6
1.2.1 Disclaimer Regarding Language in This Release . . . 6
1.2.2 Usage of W3C XML Schema . . . 6
1.3 AUTOSAR Standards . . . 7
1.3.1 Introduction . . . 7
1.3.2 Definition . . . 8
1.3.3 Overview of AUTOSAR's Standards . . . 8
1.3.3.1 Adaptive Platform . . . 8
1.3.3.2 Classic Platform . . . 8
1.3.3.3 Foundation . . . 9
1.3.4 Dependencies Between Standards . . . 9
1.3.5 Precedence of Standards . . . 9
1.3.6 Dependencies to Other Standards . . . 9
1.4 Release Numbering and Life Cycle . . . 10
1.4.1 Platform Release Number . . . 10
1.4.2 Internal Release Number . . . 10
1.4.3 AUTOSAR Schema Versions . . . 10
1.4.4 Release Life Cycle of a Major Release . . . 11
1.4.5 Life Cycle States of Specification Items and Requirements . . . 12
1.5 Content of Chapters . . . 12
2 Summary of Changes in Release R26-11 14
2.1 Concepts . . . 14
2.1.1 Introduced Concepts . . . 14
2.1.1.1 CANsec . . . 14
2.1.1.2 DMA Driver . . . 14
2.1.1.3 Revise MACsec support . . . 15
2.1.1.4 Dynamic PNC for CAN . . . 15
2.1.1.5 PostBuild ConfigClass simplification . . . 16
2.1.1.6 Encryption Support in SecOC . . . 16
2.1.2 Impact of Concepts . . . 16
2.1.3 Validated Concepts . . . 17
2.2 Specifications . . . 18
2.2.1 New Specifications . . . 18
2.2.2 Migrated Specifications . . . 18
2.2.3 Obsolete Specifications . . . 18
2.2.4 Removed Specifications . . . 18
2.2.5 Reworked Specifications . . . 20
2.2.6 Moved Specification Parts . . . 20
2.2.7 Renamed Specifications . . . 20

2.3 Release Documentation 20
3 Specification Overview 21
4 Remarks to Known Technical Deficiencies 30
4.1 CP SWS XCP 30
4.2 CP SWS SOMEIPTransportProtocol 30
4.3 CP SWS TransformerGeneral 31
4.4 CP SWS SOMEIPTransformer 31
4.5 CP SWS ServiceDiscovery 32
4.6 CP SWS SAEJ1939TransportLayer 32
4.7 CP SWS SAEJ1939RequestManager 32
4.8 CP SWS SAEJ1939NetworkManagement 33
4.9 CP SWS SAEJ1939DiagnosticCommunicationManager 33
4.10 CP SWS LSduRouter 33
4.11 CP SWS IEEE1722TransportLayer 34
4.12 CP SWS FlexRayTransceiverDriver 34
4.13 CP SWS FlexRayISOTransportLayer 35
4.14 CP SWS FlexRayDriver 35
4.15 CP SWS EthernetTransceiverDriver 35
4.16 CP SWS CoreTest 36
4.17 CP SWS BusMirroring 36
4.18 CP RS CoreTest 37
4.19 CP EXP BSWDistributionGuide 37
4.20 CP SWS MACsecKeyAgreement 37
5 Release History 39
5.1 Release R26-11 39

References

[1] ISO/IEC Directives, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents
https://www.iso.org/sites/directives/current/part2/index.xhtml

[2] CiA 613-2 version 1.0.0 (DSP) - CAN XL add-on services - Part 2: Security
http://www.can-cia.org

[3] Specification of Bus Mirroring Protocol
AUTOSAR_FO_PRS_BusMirroringProtocol

[4] Specification of Crypto Driver
AUTOSAR_CP_SWS_CryptoDriver

[5] Specification of Timing Extensions for Classic Platform
AUTOSAR_CP_TPS_TimingExtensions

[6] Modeling Guidelines of Basic Software EA UML Model
AUTOSAR_CP_TR_BSWUMLModelModelingGuide

1 Introduction

1.1 Scope of This Document

This document provides an overview of the AUTOSAR standard Classic Platform release R26-11.

1.2 Terminology and Licenses

1.2.1 Disclaimer Regarding Language in This Release

This release may contain terminology or phrasing that, while historically or technically conventional, could be perceived as outdated, insensitive, or discriminatory in modern contexts. AUTOSAR wishes to explicitly state that any such language is unintentional and is not meant to offend, exclude, or imply prejudice against any individual or group.

Such instances typically arise from:

Historical Usage Terms that were once common in technical or industry parlance.

Legacy Systems Descriptions inherited from older systems or standards that predate current inclusive language guidelines.

Technical Specificity Language chosen for precise technical meaning, which may inadvertently carry unintended connotations.

Backward Compatibility and External References In some cases, specific terminologies cannot be altered without breaking backward compatibility with existing systems, codebases, or processes. Additionally, certain terms may be retained to align with established industry standards, external specifications, or third-party documentation, where consistency is critical for interoperability and clear communication across different platforms and organizations.

AUTOSAR is committed to fostering an inclusive environment and regularly reviews our documentation to identify and update language that does not align with our values. Among other guidelines, AUTOSAR is referring to the relevant ISO standard ISO/IEC Directives, Part 2 [1].

If you encounter any language within this document that you find concerning, AUTOSAR encourages you to provide feedback so AUTOSAR can address the feedback in future revisions.

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 be of but are not limited to 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 of 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 and
- 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 1.4.3.

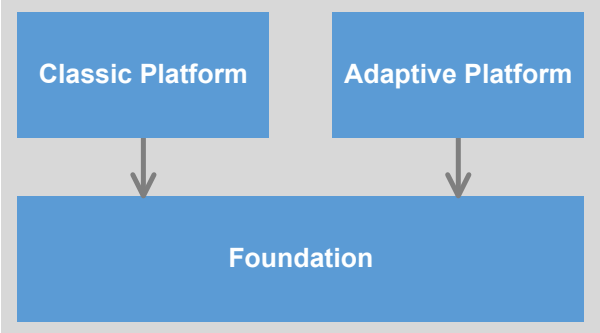


Figure 1.1: Dependencies of AUTOSAR standards

1.3.5 Precedence of Standards

In case of contradictions between AUTOSAR documents, the following precedence applies. For all externally observable communication behavior - such as on-wire message formats and encodings, timing constraints, state machines, and protocol semantics - the Protocol Specifications (PRS) are authoritative.

For software-internal aspects - such as module APIs, configuration parameters, internal state handling, and error reporting that do not affect on-wire behavior - the relevant SWS is authoritative. Where both a PRS and an SWS cover the same subject matter, the PRS shall prevail.

1.3.6 Dependencies to Other Standards

This release of the Classic Platform depends on the standard Foundation in release R26-11, 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, for example, 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. This is newly introduced in the Adaptive Platform. For Foundation, this leads to a discontinuation of the former numbering pattern (e.g., R1.5.0).
A mapping between Platform Releases and corresponding internal release numbers can be found in chapter 1.4.3. 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 AUTOSAR Schema Versions

Until the Releases CP R4.4.0 and AP R19-03, AUTOSAR released the platforms separately where a Foundation release went along with each platform release. Since compatibility between the platforms is essential to be able to have AP and CP ECUs within one vehicle project, an XML schema needs to be available that works with the

different releases. The following table gives an overview about the different schema versions and the corresponding platform releases they can be used for.

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

Table 1.1: Mapping schema version to platform releases (until CP R4.4.0/AP R19-03)

Starting with release R19-11, all platforms are released as one AUTOSAR release and therefore come along with one schema version.

Schema version	Platform release	Internal release number
AUTOSAR_00048	R19-11	R4.5.0
AUTOSAR_00049	R20-11	R4.6.0
AUTOSAR_00050	R21-11	R4.7.0
AUTOSAR_00051	R22-11	R4.8.0
AUTOSAR_00052	R23-11	R4.9.0
AUTOSAR_00053	R24-11	R4.10.0
AUTOSAR_00054	R25-11	R4.11.0
AUTOSAR_00055	**R26-11**	**R4.12.0**

Table 1.2: Mapping schema version to platform release and internal release number (since R19-11)

1.4.4 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 the 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; only maintenance of existing content is provided through zero, one or several revisions (e.g., R3.2.2).
4. Issue Notice: No more revisions but zero, one or several issue notices, i.e., updates of the list of known issues until the end of life cycle.

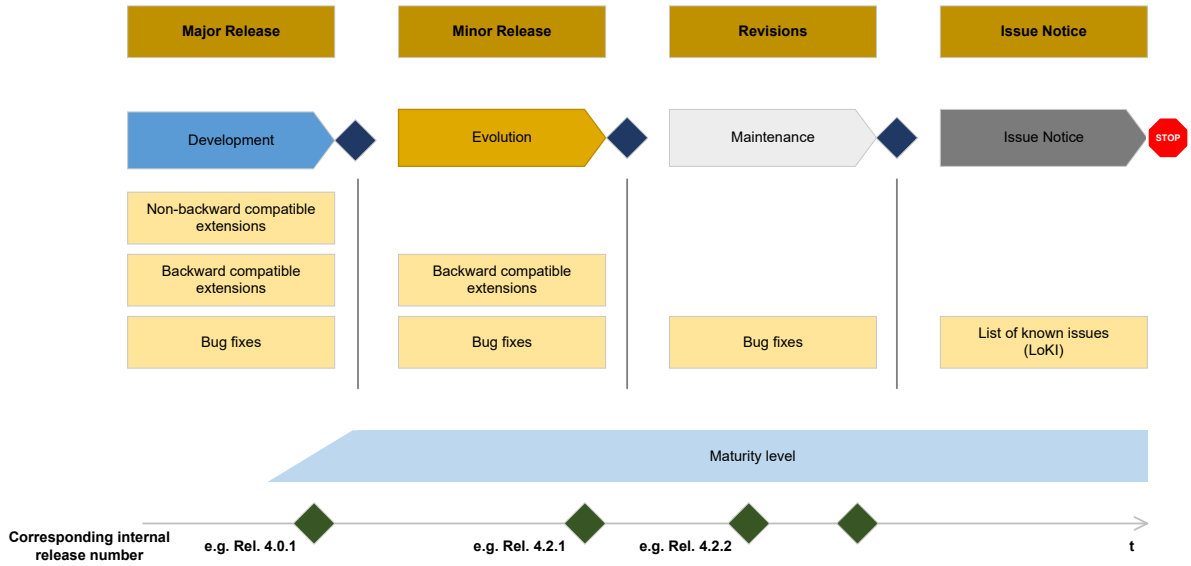


Figure 1.2: Life cycle model of AUTOSAR standards

According to the release life cycle of AUTOSAR the release R26-11 is a minor release.

1.4.5 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 “Status”. The states are the same as the specification item states.

1.5 Content of Chapters

This document is structured as follows:

- Chapter 1 introduces 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 R26-11. This chapter is structured according to the clusters of AUTOSAR release R26-11.
- Chapter 4 contains remarks about known technical deficiencies.
- Chapter 5 contains the detailed revision history of all released specifications.

2 Summary of Changes in Release R26-11

This chapter contains a summary of the changes that have been implemented since the previous release R25-11.

2.1 Concepts

2.1.1 Introduced Concepts

The following concepts in 2.1.1.1–2.1.1.6 have been introduced.

2.1.1.1 CANsec

The CANsec concept implements the secure communication protocol CANsec, as specified in CiA 613-2 [2, CiA 613-2]. As such, it brings the following benefits to Classic AUTOSAR:

- Secure communication on CAN XL (at line speed with HW acceleration)
- Secure-by-default, unsecured (bypassed) messages need to be explicitly defined
- Secure communication at OSI layer 2, which allows securing all higher-level protocols
- Built upon AUTOSAR MACsec concepts, i.e., similar configuration and secure network concepts

2.1.1.2 DMA Driver

The concept introduces a standardized MCAL driver for Direct Memory Access (DMA) controllers. Until now, DMA usage on Classic Platform ECUs was implemented either in vendor-specific drivers or privately inside individual peripheral drivers, which made DMA-based data movement hard to reuse and port.

The new DMA Driver (module Dma) provides a hardware-independent interface to configure and control DMA transfers between memory and peripherals with reduced CPU load. Main features are:

- software-triggered (polling or interrupt) and hardware-triggered channels,
- Transfer Control Descriptors (TCDs) with inner and outer loops and transfer-element width adaptation,
- scatter-gather descriptor chaining,
- runtime update of transfer parameters and channel-to-TCD mapping,

- suspend/resume, status query and notification callbacks for completion, transfer errors and hardware errors,
- partition assignment of channels and TCDs for multi-partition systems,
- standardized extension points for hardware-specific configuration (e.g. CRC calculation, circular buffers).

Peripherals with a private DMA engine (e.g. Ethernet controllers) remain under the control of their own drivers.

The concept introduces a standardized DMA Driver for the Classic Platform. In the Foundation, it adds the optional feature `FO_TR_FEAT_00284` (Direct_Memory_Access, child of the `FO_TR_FEAT_00266` MCU abstraction) to the feature model and the term *Direct Memory Access* to the glossary.

2.1.1.3 Revise MACsec support

The concept focuses on specification gaps and ambiguous requirements regarding the MACsec support in the communication stack of AUTOSAR Classic Platform:

- Replace default values at <Eth|EthTrcv|EthSwt>_MacSecResetRxSc
- Constrain MACsec support at host and uplink_port for point to point connection
- Refine specification regarding ICV calculation
- Clarify interpretation for multiple configured bypass rules
- Refine configuration of destination MAC address at MKA to exchange MKPDUs
- Add constraint that rejects configurations where frame preemption and MACsec support is enabled at the same time

2.1.1.4 Dynamic PNC for CAN

Dynamic PNC for CAN
The concept improves the flexibility of Partial Networking in CAN by enabling the PNC assignment of an ECU to be changed without reprogramming the ECU software. Instead, only the configuration parameters related to Partial Networking are updated using the Post-Build configuration mechanism.

Initial configuration
The initial ECU configuration shall provide the PNC-related resources required for possible future changes of the PNC assignment. Partial Networking must be enabled already in the initial configuration. The initial configuration of the PNCs must plan, in addition to the PNCs actually used, some spare PNCs that can be used in the future. Adding PNCs To modify the PNC assignment, the PNC-related configuration parameters affected by the new assignment are updated using the Post-Build configuration

mechanism. This allows the PNC configuration to be changed without modifying the ECU software.

Restrictions
This concept applies only to Classic Platform.

The changes required to change the PNC assignment are restricted to the Partial Networking Byte Masks and do not include changes of the PNC ID or the PNC Bit Vector Length.

2.1.1.5 PostBuild ConfigClass simplification

The concept removes the assignment of configuration classes to configuration times in the Ecu configuration tables and simplifies the chapter 10 of SWS documents. The assignment is replaced by a simpler approach that introduces four boolean flags Post-Build Selectable Multiplicity, Post-Build Selectable Value, Post-Build Loadable Multiplicity, Post-Build Loadable Value to chapter 10 tables.

2.1.1.6 Encryption Support in SecOC

The Secure Onboard Communication functionality has been extended to optionally protect the confidentiality of selected I-PDUs in addition to their authenticity and freshness. Authenticated Encryption with Associated Data (AEAD) can be configured as an alternative to authentication-only protection, sufficient payload capacity of the underlaying communication channel provided. The SecOC processing, configuration, and protocol specifications were updated to support encryption and decryption through the Crypto Service Manager and to introduce dedicated nonce-management interfaces. The corresponding specifications for the AUTOSAR Classic and Adaptive Platforms were updated while retaining backward compatibility with existing authentication-only configurations.

2.1.2 Impact of Concepts

The introduced concepts have an 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.
- If specifications or models like the BSW UML model or the ECUC model are only indirectly affected because they just provide artifacts for other specifications, they are not listed here.

Concept Name	Specification Name	Affected Standard	Concept Lifecycle
CANsec	CP SWS BSWModemanager	Classic Platform, Foundation	valid
	CP SWS CANDriver		
	CP SWS CANInterface		
	CP SWS CANXLDriver		
	CP SWS MACsecKeyAgreement		
	FO TR Features		
	FO TR Glossary		
DMA Driver	**CP SWS DMADriver**	Classic Platform, Foundation	draft
	CP EXP LayeredSoftwareArchitecture		
	CP SWS BSWGeneral		
	FO TR Features		
	FO TR Glossary		
Revise MACsec support	CP SWS EthernetDriver	Classic Platform	valid
	CP SWS EthernetInterface		
	CP SWS EthernetSwitchDriver		
	CP SWS EthernetTransceiverDriver		
	CP SWS MACsecKeyAgreement		
Dynamic PNC for CAN	CP SWS NetworkManagementInterface	Classic Platform	valid
	CP EXP ModeManagementGuide		
PostBuild ConfigClass simplification	CP TPS EcuConfiguration	Classic Platform, Foundation	valid
	CP SWS BSWGeneral		
	FO MMOD MetaModel		
Encryption Support in SecOC	CP TPS SoftwareComponentTemplate	Classic Platform, Adaptive Platform, Foundation	valid
	CP TPS SystemTemplate		
	CP SWS CryptoServiceManager		
	AP TPS ManifestSpecification		
	AP SWS CommunicationManagement		
	CP SWS SecureOnboardCommunication		
	FO PRS SecOCProtocol		
	FO EXP SecurityOverview		
	FO TR Features		

Table 2.1: Impact of concepts

2.1.3 Validated Concepts

The following concepts have been validated:

- "Classic Platform Flexibility", only the R21-11 feature set of "Classic Platform Flexibility" has been validated

2.2 Specifications

2.2.1 New Specifications

The following new specifications have been introduced via concepts:

- CP SWS DMADriver (Specification of DMA Driver, UID 1148)

In addition to the above listed new specifications, the following documents have been added with this release:

- none

2.2.2 Migrated Specifications

With this release, the following specification has been moved from or to Classic Platform:

- none

2.2.3 Obsolete Specifications

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

- none

2.2.4 Removed Specifications

The following specifications have been removed from the AUTOSAR standard:

- CP RS ChineseV2XCommunication
  (Requirements on Chinese Vehicle-2-X Communication, UID 989)
- CP RS HWTestManager
  (Requirements on Hardware Test Manager on start up and shutdown, UID 760)
- CP RS V2XCommunication
  (Requirements on Vehicle-2-X Communication, UID 791)
- CP SWS CellularV2XDriver
  (Specification of Cellular Vehicle-2-X Driver, UID 1030)
- CP SWS ChineseV2XManagement
  (Specification of Chinese Vehicle-2-X Management, UID 1031)

- CP SWS ChineseV2XMessage
  (Specification of Chinese Vehicle-2-X Message, UID 990)
- CP SWS ChineseV2XNetwork
  (Specification of Chinese Vehicle-2-X Network, UID 991)
- CP SWS ChineseV2XSecurity
  (Specification of Chinese Vehicle-2-X Security, UID 1032)
- CP SWS FlexRayARTransportLayer
  (Specification of FlexRay AUTOSAR Transport Layer, UID 601)
- CP SWS HWTestManager
  (Specification of Hardware Test Manager on start up and shutdown, UID 703)
- CP SWS V2XBasicTransport
  (Specification of Vehicle-2-X Basic Transport, UID 794)
- CP SWS V2XDataManager
  (Specification of Vehicle-2-X Data Manager, UID 1023)
- CP SWS V2XFacilities
  (Specification of Vehicle-2-X Facilities, UID 795)
- CP SWS V2XGeoNetworking
  (Specification of Vehicle-2-X Geo Networking, UID 793)
- CP SWS V2XManagement
  (Specification of Vehicle-2-X Management, UID 796)
- CP SWS WirelessEthernetDriver
  (Specification of Wireless Ethernet Driver, UID 798)
- CP SWS WirelessEthernetTransceiverDriver
  (Specification of Wireless Ethernet Transceiver Driver, UID 799)
- CP TR HWTestManagementIntegrationGuide
  (Specification and Integration of Hardware Test Management at start up and shutdown, UID 804)
- CP RS LIN
  (Requirements on LIN, UID 42)
- CP RS SAEJ1939
  (Requirements on BSW Modules for SAE J1939, UID 613)
- CP RS Cryptostack
  (Requirements on Crypto Stack, UID 426)
- CP RS Gateway
  (Requirements on Gateway, UID 6)

- CP RS DebugTraceProfile
  (Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components, UID 916)
- CP RS COM
  (Requirements on Communication, UID 2)
- CP RS ChargingManager
  (Requirements on Charging Manager, UID 1096)
- CP RS CAN
  (Requirements on CAN, UID 1)
- CP RS BusMirroring
  (Requirements on Bus Mirroring, UID 874)
- CP RS IPDUMultiplexer
  (Requirements on I-PDU Multiplexer, UID 181)
- CP RS XCP
  (Requirements on Module XCP, UID 429)
- CP RS Libraries
  (Requirements on Libraries, UID 314)

2.2.5 Reworked Specifications

The following specifications have been changed fundamentally in this release:

- none

2.2.6 Moved Specification Parts

The following specification parts have been moved to other documents in this release:

- none

2.2.7 Renamed Specifications

The following specifications have been renamed in this release:

- CP SWS TransformerGeneral (General Specification of Transformers, UID 658)

2.3 Release Documentation

There are no major changes in the release documentation.

3 Specification Overview

The published specifications are divided into the clusters:

- ReleaseDocumentation
- ApplicationInterfaces
- BSWGeneral
- Communication
- Crypto
- Diagnostics
- General
- GlobalTime
- IO
- Libraries
- MCAL
- Memory
- MethodologyAndTemplates
- ModeManagement
- RTE
- SWArch
- Safety
- Security
- SystemServices

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

Specification Name	Long Name	File Name	Life cycle changes
ReleaseDocumentation			
CP TR ReleaseOverview	Classic Platform Release Overview	AUTOSAR_CP_TR_Release Overview	
CP TR SpecificationHashes	AUTOSAR Classic Platform Specification Hashes	AUTOSAR_CP_TR_ SpecificationHashes	
ApplicationInterfaces			
CP EXP AIADASAndVMC	Explanation of Application Interface of AD/ADAS vehicle motion control	AUTOSAR_CP_EXP_ AIADASAndVMC	



△

Specification Name	Long Name	File Name	Life cycle changes
CP EXP AIBodyAndComfort	Explanation of Application Interfaces of the Body and Comfort Domain	AUTOSAR_CP_EXP_AIBody AndComfort	
CP EXP AIChassis	Explanation of Application Interfaces of the Chassis Domain	AUTOSAR_CP_EXP_ AIChassis	
CP EXP AIHMIMultimediaAnd Telematics	Explanation of Application Interfaces of the HMI, Multimedia and Telematics Domain	AUTOSAR_CP_EXP_ AIHMIMultimediaAnd Telematics	
CP EXP AIOccupantAnd PedestrianSafety	Explanation of Application Interfaces of Occupant and Pedestrian Safety Systems Domain	AUTOSAR_CP_EXP_ AIOccupantAndPedestrian Safety	
CP EXP AIPowertrain	Explanation of Application Interfaces of the Powertrain Engine Domain	AUTOSAR_CP_EXP_ AIPowertrain	
CP EXP AIUserGuide	Application Interfaces User Guide	AUTOSAR_CP_EXP_AIUser Guide	
CP MOD AISpecification	XML Specification of Application Interfaces	AUTOSAR_CP_MOD_ AISpecification	
CP MOD AISpecification Examples	Application Interface Examples	AUTOSAR_CP_MOD_ AISpecificationExamples	
CP RS SWCModeling	Requirements on SW-C and System Modeling	AUTOSAR_CP_RS_ SWCModeling	
CP TR AIDesignPatterns Catalogue	Application Design Patterns Catalogue	AUTOSAR_CP_TR_AIDesign PatternsCatalogue	
CP TR AIMeasurement CalibrationDiagnostics	Modeling and Naming Aspects for Documentation, Measurement, and Calibration	AUTOSAR_CP_TR_ AIMeasurementCalibration Diagnostics	
CP TR SWCModelingGuide	SW-C and System Modeling Guide	AUTOSAR_CP_TR_ SWCModelingGuide	
BSWGeneral			
CP EXP ApplicationLevelError Handling	Explanation of Error Handling on Application Level	AUTOSAR_CP_EXP_ ApplicationLevelErrorHandling	
CP EXP BSWDistributionGuide	Guide to BSW Distribution	AUTOSAR_CP_EXP_ BSWDistributionGuide	
CP EXP CDDDesignAnd IntegrationGuideline	Complex Driver design and integration guideline	AUTOSAR_CP_EXP_ CDDDesignAndIntegration Guideline	
CP EXP ErrorDescription	Description of the AUTOSAR standard errors	AUTOSAR_CP_EXP_Error Description	
CP MOD BSWUMLModel	Basic Software UML Model	AUTOSAR_CP_MOD_ BSWUMLModel	
CP RS BSWGeneral	General Requirements on Basic Software Modules	AUTOSAR_CP_RS_ BSWGeneral	
CP SWS BSWGeneral	General Specification of Basic Software Modules	AUTOSAR_CP_SWS_ BSWGeneral	
CP SWS CommunicationStack Types	Specification of Communication Stack Types	AUTOSAR_CP_SWS_ CommunicationStackTypes	
CP SWS PlatformTypes	Specification of Platform Types for Classic Platform	AUTOSAR_CP_SWS_Platform Types	
CP SWS StandardTypes	Specification of Standard Types	AUTOSAR_CP_SWS_ StandardTypes	
CP TR BSWUMLModel ModelingGuide	Modeling Guidelines of Basic Software EA UML Model	AUTOSAR_CP_TR_ BSWUMLModelModelingGuide	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
Communication			
CP RS Ethernet	Requirements on Ethernet Support in AUTOSAR	AUTOSAR_CP_RS_Ethernet	
CP RS FlexRay	Requirements on FlexRay	AUTOSAR_CP_RS_FlexRay	
CP RS I2CDriver	Requirements on I2C Driver	AUTOSAR_CP_RS_I2CDriver	
CP RS SPIHandlerDriver	Requirements on SPI Handler/Driver	AUTOSAR_CP_RS_SPIHandlerDriver	
CP RS Transformer	Requirements on Transformer	AUTOSAR_CP_RS_Transformer	
CP SWS BusMirroring	Specification of Bus Mirroring	AUTOSAR_CP_SWS_BusMirroring	
CP SWS CANDriver	Specification of CAN Driver	AUTOSAR_CP_SWS_CANDriver	
CP SWS CANInterface	Specification of CAN Interface	AUTOSAR_CP_SWS_CANInterface	
CP SWS CANNetwork Management	Specification of CAN Network Management	AUTOSAR_CP_SWS_CANNetworkManagement	
CP SWS CANStateManager	Specification of CAN State Manager	AUTOSAR_CP_SWS_CANStateManager	
CP SWS CANTransceiverDriver	Specification of CAN Transceiver Driver	AUTOSAR_CP_SWS_CANTransceiverDriver	
CP SWS CANTransportLayer	Specification of CAN Transport Layer	AUTOSAR_CP_SWS_CANTransportLayer	
CP SWS CANXLDriver	Specification for CAN XL Driver	AUTOSAR_CP_SWS_CANXLDriver	
CP SWS CANXLTransceiver Driver	Specification of CAN XL Transceiver Driver	AUTOSAR_CP_SWS_CANXLTransceiverDriver	
CP SWS COM	Specification of Communication	AUTOSAR_CP_SWS_COM	
CP SWS COMBased Transformer	Specification of COM Based Transformer	AUTOSAR_CP_SWS_COMBasedTransformer	
CP SWS DataDistribution Service	Specification of Data Distribution Service for Classic Platform	AUTOSAR_CP_SWS_DataDistributionService	
CP SWS DataDistribution ServiceTransformer	Specification of Data Distribution Service Transformer	AUTOSAR_CP_SWS_DataDistributionServiceTransformer	
CP SWS DiagnosticLogAnd Trace	Specification of Diagnostic Log and Trace	AUTOSAR_CP_SWS_DiagnosticLogAndTrace	
CP SWS DiagnosticOverIP	Specification of Diagnostic over IP	AUTOSAR_CP_SWS_DiagnosticOverIP	
CP SWS E2ETransformer	Specification of Module E2E Transformer	AUTOSAR_CP_SWS_E2ETransformer	
CP SWS EthernetDriver	Specification of Ethernet Driver	AUTOSAR_CP_SWS_EthernetDriver	
CP SWS EthernetInterface	Specification of Ethernet Interface	AUTOSAR_CP_SWS_EthernetInterface	
CP SWS EthernetState Manager	Specification of Ethernet State Manager	AUTOSAR_CP_SWS_EthernetStateManager	
CP SWS EthernetSwitchDriver	Specification of Ethernet Switch Driver	AUTOSAR_CP_SWS_EthernetSwitchDriver	
CP SWS EthernetTransceiver Driver	Specification of Ethernet Transceiver Driver	AUTOSAR_CP_SWS_EthernetTransceiverDriver	
CP SWS FlexRayDriver	Specification of FlexRay Driver	AUTOSAR_CP_SWS_FlexRayDriver	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
CP SWS FlexRayISOTransport Layer	Specification of FlexRay ISO Transport Layer	AUTOSAR_CP_SWS_FlexRay ISOTransportLayer	
CP SWS FlexRayInterface	Specification of FlexRay Interface	AUTOSAR_CP_SWS_FlexRay Interface	
CP SWS FlexRayNetwork Management	Specification of FlexRay Network Management	AUTOSAR_CP_SWS_FlexRay NetworkManagement	
CP SWS FlexRayState Manager	Specification of FlexRay State Manager	AUTOSAR_CP_SWS_FlexRay StateManager	
CP SWS FlexRayTransceiver Driver	Specification of FlexRay Transceiver Driver	AUTOSAR_CP_SWS_FlexRay TransceiverDriver	
CP SWS I2CDriver	Specification of I2C Driver	AUTOSAR_CP_SWS_I2 CDriver	
CP SWS IEEE1722Transport Layer	Specification of IEEE1722 Transport Protocol Module	AUTOSAR_CP_SWS_ IEEE1722TransportLayer	
CP SWS IPDUMultiplexer	Specification of I-PDU Multiplexer	AUTOSAR_CP_SWS_ IPDUMultiplexer	
CP SWS ISO15118Charging	Specification of ISO15118 Charging	AUTOSAR_CP_SWS_ ISO15118Charging	
CP SWS LINDriver	Specification of LIN Driver	AUTOSAR_CP_SWS_ LINDriver	
CP SWS LINInterface	Specification of LIN Interface	AUTOSAR_CP_SWS_ LINInterface	
CP SWS LINStateManager	Specification of LIN State Manager	AUTOSAR_CP_SWS_LINState Manager	
CP SWS LINTransceiverDriver	Specification of LIN Transceiver Driver	AUTOSAR_CP_SWS_ LINTransceiverDriver	
CP SWS LSduRouter	Specification of Linklayer Sdu Routing Module	AUTOSAR_CP_SWS_LSdu Router	
CP SWS MACsecKey Agreement	Specification of MACsec Key Agreement	AUTOSAR_CP_SWS_MACsec KeyAgreement	
CP SWS NetworkManagement Interface	Specification of Network Management Interface	AUTOSAR_CP_SWS_Network ManagementInterface	
CP SWS PDURouter	Specification of PDU Router	AUTOSAR_CP_SWS_ PDURouter	
CP SWS SAEJ1939Functional SafetyCommProtocol	Specification of a Functional Safety Communication Protocol Handler for SAE J1939	AUTOSAR_CP_SWS_ SAEJ1939FunctionalSafety CommProtocol	
CP SWS SAEJ1939Network Management	Specification of Network Management for SAE J1939	AUTOSAR_CP_SWS_ SAEJ1939Network Management	
CP SWS SAEJ1939Request Manager	Specification of a Request Manager for SAE J1939	AUTOSAR_CP_SWS_ SAEJ1939RequestManager	
CP SWS SAEJ1939Transport Layer	Specification of a Transport Layer for SAE J1939	AUTOSAR_CP_SWS_ SAEJ1939TransportLayer	
CP SWS SOMEIPTransformer	Specification of SOME/IP Transformer	AUTOSAR_CP_SWS_ SOMEIPTransformer	
CP SWS SOMEIPTransport Protocol	Specification on SOME/IP Transport Protocol	AUTOSAR_CP_SWS_ SOMEIPTransportProtocol	
CP SWS SPIHandlerDriver	Specification of SPI Handler/Driver	AUTOSAR_CP_SWS_ SPIHandlerDriver	
CP SWS SecureOnboard Communication	Specification of Secure Onboard Communication	AUTOSAR_CP_SWS_Secure OnboardCommunication	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
CP SWS ServiceDiscovery	Specification of Service Discovery	AUTOSAR_CP_SWS_Service Discovery	
CP SWS SocketAdaptor	Specification of Socket Adaptor	AUTOSAR_CP_SWS_Socket Adaptor	
CP SWS TcpIp	Specification of TCP/IP Stack	AUTOSAR_CP_SWS_TcpIp	
CP SWS TransformerGeneral	General Specification of Transformers	AUTOSAR_CP_SWS_ TransformerGeneral	
CP SWS UDPNetwork Management	Specification of UDP Network Management	AUTOSAR_CP_SWS_ UDPNetworkManagement	
CP SWS VDPCMRemote	Specification of VDP Communication Module Remote	AUTOSAR_CP_SWS_ VDPCMRemote	
CP SWS XCP	Specification of Module XCP	AUTOSAR_CP_SWS_XCP	
Crypto			
CP EXP UtilizationOfCrypto Services	Utilization of Crypto Services	AUTOSAR_CP_EXP_ UtilizationOfCryptoServices	
CP SWS CryptoDriver	Specification of Crypto Driver	AUTOSAR_CP_SWS_Crypto Driver	
CP SWS CryptoInterface	Specification of Crypto Interface	AUTOSAR_CP_SWS_Crypto Interface	
CP SWS CryptoService Manager	Specification of Crypto Service Manager	AUTOSAR_CP_SWS_Crypto ServiceManager	
CP SWS KeyManager	Specification of Key Manager	AUTOSAR_CP_SWS_Key Manager	
Diagnostics			
CP SWS Diagnostic CommunicationManager	Specification of Diagnostic Communication Manager	AUTOSAR_CP_SWS_ DiagnosticCommunication Manager	
CP SWS DiagnosticEvent Manager	Specification of Diagnostic Event Manager	AUTOSAR_CP_SWS_ DiagnosticEventManager	
CP SWS SAEJ1939Diagnostic CommunicationManager	Specification of a Diagnostic Communication Manager for SAE J1939	AUTOSAR_CP_SWS_ SAEJ1939Diagnostic CommunicationManager	
General			
CP EXP LayeredSoftware Architecture	Layered Software Architecture	AUTOSAR_CP_EXP_Layered SoftwareArchitecture	
CP SWS BulkNvDataManager	Specification of Bulk NvData Manager	AUTOSAR_CP_SWS_BulkNv DataManager	
CP TR VFB	Virtual Functional Bus	AUTOSAR_CP_TR_VFB	
GlobalTime			
CP SWS SynchronizedTime BaseManager	Specification of Synchronized Time-Base Manager	AUTOSAR_CP_SWS_ SynchronizedTimeBase Manager	
CP SWS TimeSyncOverCAN	Specification of Time Synchronization over CAN	AUTOSAR_CP_SWS_Time SyncOverCAN	
CP SWS TimeSyncOver Ethernet	Specification of Time Synchronization over Ethernet	AUTOSAR_CP_SWS_Time SyncOverEthernet	
CP SWS TimeSyncOverFlex Ray	Specification of Time Synchronization over FlexRay	AUTOSAR_CP_SWS_Time SyncOverFlexRay	
IO			
CP RS ADCDriver	Requirements on ADC Driver	AUTOSAR_CP_RS_ADCDriver	
CP RS DIODriver	Requirements on DIO Driver	AUTOSAR_CP_RS_DIODriver	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
CP RS ICUDriver	Requirements on ICU Driver	AUTOSAR_CP_RS_ICUDriver	
CP RS IOHWAbstraction	Requirements on I/O Hardware Abstraction	AUTOSAR_CP_RS_ IOHWAbstraction	
CP RS OCUDriver	Requirements on OCU Driver	AUTOSAR_CP_RS_ OCUDriver	
CP RS PWMDriver	Requirements on PWM Driver	AUTOSAR_CP_RS_ PWMDriver	
CP RS PortDriver	Requirements on Port Driver	AUTOSAR_CP_RS_PortDriver	
CP SWS ADCDriver	Specification of ADC Driver	AUTOSAR_CP_SWS_ ADCDriver	
CP SWS DIODriver	Specification of DIO Driver	AUTOSAR_CP_SWS_ DIODriver	
CP SWS ICUDriver	Specification of ICU Driver	AUTOSAR_CP_SWS_ ICUDriver	
CP SWS IOHardware Abstraction	Specification of I/O Hardware Abstraction	AUTOSAR_CP_SWS_ IOHardwareAbstraction	
CP SWS OCUDriver	Specification of OCU Driver	AUTOSAR_CP_SWS_ OCUDriver	
CP SWS PWMDriver	Specification of PWM Driver	AUTOSAR_CP_SWS_ PWMDriver	
CP SWS PortDriver	Specification of Port Driver	AUTOSAR_CP_SWS_Port Driver	
Libraries			
CP EXP MacroEncapsulationof InterpolationCalls	Macro Encapsulation of Interpolation Calls	AUTOSAR_CP_EXP_Macro EncapsulationofInterpolation Calls	
CP SWS BFXLibrary	Specification of Bit Handling Library	AUTOSAR_CP_SWS_ BFXLibrary	
CP SWS BSWMulticoreLibrary	Specification of Basic Software Multicore Library	AUTOSAR_CP_SWS_ BSWMulticoreLibrary	
CP SWS CRCLibrary	Specification of CRC Library	AUTOSAR_CP_SWS_ CRCLibrary	
CP SWS E2ELibrary	Specification of SW-C End-to-End Communication Protection Library	AUTOSAR_CP_SWS_E2 ELibrary	
CP SWS EFXLibrary	Specification of Extended Fixed Point Library	AUTOSAR_CP_SWS_ EFXLibrary	
CP SWS IFLLibrary	Specification of Floating Point Interpolation Library	AUTOSAR_CP_SWS_ IFLLibrary	
CP SWS IFXLibrary	Specification of Fixed Point Interpolation Library	AUTOSAR_CP_SWS_ IFXLibrary	
CP SWS MFLLibrary	Specification of Floating Point Math Library	AUTOSAR_CP_SWS_ MFLLibrary	
CP SWS MFXLibrary	Specification of Fixed Point Math Library	AUTOSAR_CP_SWS_ MFXLibrary	
CP SWS MSFLibrary	Specification of MSFLibrary	AUTOSAR_CP_SWS_ MSFLibrary	
MCAL			
CP RS CoreTest	Requirements on Core Test	AUTOSAR_CP_RS_CoreTest	
CP RS GPTDriver	Requirements on GPT Driver	AUTOSAR_CP_RS_GPTDriver	
CP RS MCUDriver	Requirements on MCU Driver	AUTOSAR_CP_RS_ MCUDriver	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
CP RS SPALGeneral	General Requirements on SPAL	AUTOSAR_CP_RS_SPALGeneral	
CP SWS CoreTest	Specification of Core Test	AUTOSAR_CP_SWS_CoreTest	
CP SWS DMADriver	Specification of DMA Driver	AUTOSAR_CP_SWS_DMADriver	initial release
CP SWS GPTDriver	Specification of GPT Driver	AUTOSAR_CP_SWS_GPTDriver	
CP SWS MCUDriver	Specification of MCU Driver	AUTOSAR_CP_SWS_MCUDriver	
Memory			
CP EXP FirmwareOverTheAir	Explanation of Firmware Over-The-Air	AUTOSAR_CP_EXP_FirmwareOverTheAir	
CP EXP NVDataHandling	NV Data Handling Guideline	AUTOSAR_CP_EXP_NVDataHandling	
CP RS FirmwareOverTheAir	Requirements on Firmware Over-The-Air	AUTOSAR_CP_RS_FirmwareOverTheAir	
CP RS FlashTest	Requirements on Flash Test	AUTOSAR_CP_RS_FlashTest	
CP RS MemoryHWAbstraction Layer	Requirements on Memory Hardware Abstraction Layer	AUTOSAR_CP_RS_MemoryHWAbstractionLayer	
CP RS MemoryServices	Requirements on Memory Services	AUTOSAR_CP_RS_MemoryServices	
CP RS RAMTest	Requirements on RAM Test	AUTOSAR_CP_RS_RAMTest	
CP SWS EEPROMAbstraction	Specification of EEPROM Abstraction	AUTOSAR_CP_SWS_EEPROMAbstraction	
CP SWS Flash EEPROMEmulation	Specification of Flash EEPROM Emulation	AUTOSAR_CP_SWS_FlashEEPROMEmulation	
CP SWS FlashTest	Specification of Flash Test	AUTOSAR_CP_SWS_FlashTest	
CP SWS MemoryAbstraction Interface	Specification of Memory Abstraction Interface	AUTOSAR_CP_SWS_MemoryAbstractionInterface	
CP SWS MemoryAccess	Specification of Memory Access	AUTOSAR_CP_SWS_MemoryAccess	
CP SWS MemoryDriver	Specification of Memory Driver	AUTOSAR_CP_SWS_MemoryDriver	
CP SWS MemoryMapping	Specification of Memory Mapping	AUTOSAR_CP_SWS_MemoryMapping	
CP SWS NVRAMManager	Specification of NVRAM Manager	AUTOSAR_CP_SWS_NVRAMManager	
CP SWS RAMTest	Specification of RAM Test	AUTOSAR_CP_SWS_RAMTest	
MethodologyAndTemplates			
CP EXP ModelingShowCases	Modeling Show Cases Examples	AUTOSAR_CP_EXP_ModelingShowCases	
CP MOD ECUConfiguration Parameters	Specification of ECU Configuration Parameters (XML)	AUTOSAR_CP_MOD_ECUConfigurationParameters	
CP TPS BSWModule DescriptionTemplate	Basic Software Module Description Template	AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate	
CP TPS DiagnosticExtract Template	Diagnostic Extract Template	AUTOSAR_CP_TPS_DiagnosticExtractTemplate	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
CP TPS ECUConfiguration	Specification of ECU Configuration	AUTOSAR_CP_TPS_ ECUConfiguration	
CP TPS ECUResource Template	Specification of ECU Resource Template	AUTOSAR_CP_TPS_ ECUResourceTemplate	
CP TPS SoftwareComponent Template	Software Component Template	AUTOSAR_CP_TPS_Software ComponentTemplate	
CP TPS SystemTemplate	System Template	AUTOSAR_CP_TPS_System Template	
CP TPS TimingExtensions	Specification of Timing Extensions for Classic Platform	AUTOSAR_CP_TPS_Timing Extensions	
CP TR GeneralBlueprints Supplement	Supplementary material of general blueprints for AUTOSAR	AUTOSAR_CP_TR_General BlueprintsSupplement	
CP TR Methodology	Methodology for Classic Platform	AUTOSAR_CP_TR_ Methodology	
CP TR ModelingShowCases	Modeling Show Cases Report	AUTOSAR_CP_TR_Modeling ShowCases	
CP TR WorkflowExample	Technical Report on Supplementary Material of Workflow Example	AUTOSAR_CP_TR_Workflow Example	
ModeManagement			
CP EXP ModeManagement Guide	Guide to Mode Management	AUTOSAR_CP_EXP_Mode ManagementGuide	
CP RS ModeManagement	Requirements on Mode Management	AUTOSAR_CP_RS_Mode Management	
CP SWS BSWModeManager	Specification of Basic Software Mode Manager	AUTOSAR_CP_SWS_ BSWModeManager	
CP SWS ECUStateManager	Specification of ECU State Manager	AUTOSAR_CP_SWS_ ECUStateManager	
RTE			
CP RS RTE	Requirements on Runtime Environment	AUTOSAR_CP_RS_RTE	
CP SWS RTE	Specification of RTE Software	AUTOSAR_CP_SWS_RTE	
SWArch			
CP EXP ARTI	Explanatory Document for Usage of AUTOSAR Run-Time Interface	AUTOSAR_CP_EXP_ARTI	
CP SWS ARTI	Specification of AUTOSAR Run-Time Interface	AUTOSAR_CP_SWS_ARTI	
Safety			
CP EXP FunctionalSafety Measures	Overview of Functional Safety Measures in AUTOSAR	AUTOSAR_CP_EXP_ FunctionalSafetyMeasures	
CP EXP SafetyUseCase	Safety Use Case Example	AUTOSAR_CP_EXP_Safety UseCase	
CP RS WatchdogDriver	Requirements on Watchdog Driver	AUTOSAR_CP_RS_Watchdog Driver	
CP SWS WatchdogDriver	Specification of Watchdog Driver	AUTOSAR_CP_SWS_ WatchdogDriver	
CP SWS WatchdogInterface	Specification of Watchdog Interface	AUTOSAR_CP_SWS_ WatchdogInterface	
CP SWS WatchdogManager	Specification of Watchdog Manager	AUTOSAR_CP_SWS_ WatchdogManager	

▽

△

Specification Name	Long Name	File Name	Life cycle changes
Security			
CP SWS Firewall	Specification of Firewall for Classic Platform	AUTOSAR_CP_SWS_Firewall	
CP SWS IntrusionDetection SystemManager	Specification of Intrusion Detection System Manager	AUTOSAR_CP_SWS_Intrusion DetectionSystemManager	
SystemServices			
CP EXP SwClusterDesignAnd IntegrationGuideline	Explanation of Software Cluster Design And Integration Guideline for Classic Platform	AUTOSAR_CP_EXP_Sw ClusterDesignAndIntegration Guideline	
CP RS FunctionInhibition Manager	Requirements on Function Inhibition Manager	AUTOSAR_CP_RS_Function InhibitionManager	
CP RS OS	Requirements on Operating System	AUTOSAR_CP_RS_OS	
CP RS SoftwareCluster Connection	Requirements on Software Cluster Connection module	AUTOSAR_CP_RS_Software ClusterConnection	
CP RS TimeService	Requirements on Time Service	AUTOSAR_CP_RS_Time Service	
CP SWS COMManager	Specification of Communication Manager	AUTOSAR_CP_SWS_ COMManager	
CP SWS DefaultErrorTracer	Specification of Default Error Tracer	AUTOSAR_CP_SWS_Default ErrorTracer	
CP SWS FunctionInhibition Manager	Specification of Function Inhibition Manager	AUTOSAR_CP_SWS_Function InhibitionManager	
CP SWS OS	Specification of Operating System	AUTOSAR_CP_SWS_OS	
CP SWS SoftwareCluster Connection	Specification of Software Cluster Connection module	AUTOSAR_CP_SWS_Software ClusterConnection	
CP SWS TimeService	Specification of Time Service	AUTOSAR_CP_SWS_Time Service	

Table 3.1: Specification overview

4 Remarks to Known Technical Deficiencies

This section and its subsections provide an overview of several important technical deficiencies of the specification. All known technical deficiencies and limitations are detailed in the individual documents.

Document UID	Document name	Document type	Section reference
26	CP SWS FlexRayDriver	SWS	4.14
74	CP SWS FlexRayTransceiverDriver	SWS	4.12
258	CP RS CoreTest	RS	4.18
259	CP SWS CoreTest	SWS	4.16
412	CP SWS XCP	SWS	4.1
425	CP SWS SAEJ1939TransportLayer	SWS	4.6
431	CP SWS EthernetTransceiverDriver	SWS	4.15
589	CP SWS FlexRayISOTransportLayer	SWS	4.13
610	CP SWS SAEJ1939DiagnosticCommunicationManager	SWS	4.9
611	CP SWS SAEJ1939RequestManager	SWS	4.7
612	CP SWS SAEJ1939NetworkManagement	SWS	4.8
616	CP SWS ServiceDiscovery	SWS	4.5
631	CP EXP BSWDistributionGuide	EXP	4.19
658	CP SWS TransformerGeneral	SWS	4.3
660	CP SWS SOMEIPTransformer	SWS	4.4
809	CP SWS SOMEIPTransportProtocol	SWS	4.2
873	CP SWS BusMirroring	SWS	4.17
1066	CP SWS MACsecKeyAgreement	SWS	4.20
1093	CP SWS IEEE1722TransportLayer	SWS	4.11
1094	CP SWS LSduRouter	SWS	4.10

Table 4.1: Overview of known technical deficiencies

4.1 CP SWS XCP

- The SET_DAQ_ID command according to the XCP CAN Transport Layer Specification is not part of the AUTOSAR XCP module
- Currently, the AUTOSAR RTE does not offer APIs for direct communication with XCP
- NAX is only configurable through the ASAM configuration file A2L.

4.2 CP SWS SOMEIPTransportProtocol

- The SOME/IP TP is a simple protocol to segment SOME/IP messages. It does not implement retry mechanism nor does it reordering of received SOME/IP segments.

- These limitations are intended to spare runtime and memory resources on receiver side.

4.3 CP SWS TransformerGeneral

Both data transformation and communication itself are very extensive fields and can get quite complex because a lot of use cases and scenarios are theoretically possible. Because these have a big impact on the functionality of transformer (especially in the RTE), this diversity makes it necessary to impose a few restrictions and assumptions to the transformers.

If the transformation targets primarily the serialization of large complex data elements, it is most efficient when the transformation is used for communication over busses with large PDU sizes (e.g. Ethernet). If busses with small PDU size are used (e.g CAN), the byte array produced by the serializer would have to be spanned over multiple PDUs which is possible but inefficient.

Subject to transformation are the data elements (VariableDataPrototypes) of ports typed with SenderReceiverInterfaces, the operations (ClientServerOperations) of ports typed with ClientServerInterfaces and non-queued external trigger events of ports typed with TriggerInterfaces with swImplPolicy not set to queued.

This imposes the majority of restrictions and is therefore the most important contraint! As a consequence of this decision, it is not possible to transform whole PDUs. The reason for this is the fact that inside the RTE (where the transformation happens) there exist no PDUs because these are built inside the Com module.

Nonetheless, it is still possible to aggregate multiple transformed data elements of Sender/Receiver-Communication into one large PDU inside Com (each transformed data element is visible within Com as an ISignal). But in this case, all data elements/ISignals contained in this PDU are transformed independently from each other, each including its own header (if the transformation adds headers). As a consequence of this, it is not possible to transform data structures where the data structure's subelements are produced by different data elements of different PPortPrototypes/SWCs.

The length of the transformer chains is not limited by the solutions chosen within this concept. But to enable a memory efficient configuration and implementation, the maximum length is artificially limited to 255 because current use cases see a maximum chain length of 3.

4.4 CP SWS SOMEIPTransformer

The SOME/IP transformer doesn't implement the whole SOME/IP protocol:

- a part is implemented by SWS Service Discovery
- a part is implemented by SWS Socket Adaptor

- a part is currently not implemented in AUTOSAR.
- The processing order of header fields in AUTOSAR CP deviates from the processing order defined in SOME/IP Protocol specification This deviation is caused by the layered architecture of AUTOSAR CP.

4.5 CP SWS ServiceDiscovery

- Although the AUTOSAR SD is able to respond to wildcard requests (ANY) for Service ID, Instance ID, Major Version, and Minor Version, this module is only able to send wildcard finds for Minor Version.
- Load Balancing Option (Priority field and Weight field) can be configured for the Offer Services. However, the Client does not evaluate these fields.
- The specification does not support setting the Discardable flag of a SOME/IP entry option to 1 and reacting to the reception of an (unknown/unsupported) option with the Discardable flag set to 1.
- The specification does not support that IPv4/IPv6 SD endpoint options are included in any SOME/IP-SD entry of a transmitted SOME/IP-SD message.
- SOME/IP-ACL contents will be the IP addresses of the allowed communication partners, so this feature will not be applicable in case of dynamic IP via DHCP is used.
- Capability records that are received from CP side will not be evaluated on AP side

4.6 CP SWS SAEJ1939TransportLayer

- 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.

4.7 CP SWS SAEJ1939RequestManager

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

4.8 CP SWS SAEJ1939NetworkManagement

The J1939 Network Management module does not support all features defined in SAE J1939-81, 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.

4.9 CP SWS SAEJ1939DiagnosticCommunicationManager

- The J1939 Diagnostic Communication Manager implements only the subset of 'Diagnostic Messages' defined in [SWS_J1939Dcm_00238].
- The DM13 does not support "Suspend Signal" and "Suspend Duration".
- NACK is not provided for received DMx messages that are not supported or not configured. This restriction mainly affects handling of DM07, DM13, and DM22.

4.10 CP SWS LSduRouter

The L-SDU Router module does not:

- have mechanisms for signal extraction or conversion,
- have mechanisms for data integrity checking (like checksums),
- change or modify the L-SDU,
- make any L-SDU payload dependent routing decisions,
- support routing of I-PDUs between Communication Interface modules with rate conversion.

Limitations on supported functionality

- The L-SDU Router module supports fan-out of L-SDUs transmitted from a local module (e.g. IEEE1722Tp) to more than one destinations. There are some limitations if the L-SDU shall be transmitted to more than one destination (fan-out 1:n; n>1), because the upper layer module is not aware how many destinations there are:
  - The L-SDU Router reports E_OK for a Transmit request from an upper layer if at least one destination lower layer reports E_OK.
  - The L-SDU Router gives a TxConfirmation to the upper layer when it receives the last TxConfirmation from destination lower layer.

  - The L-SDU Router returns E_OK for a ReleaseRxBuffer requested from the upper layer only if all destination lower layers return E_OK.

If the L-SDU fan-out is performed by the L-SDU Router, this has further consequences for upper layer module (e.g. IEEE1722Tp module):

- The TxConfirmation of the Communication Interface API will be handled in the way that the local module (e.g. IEEE1722Tp module) will be informed when the last destination has confirmed the transmission.

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

4.11 CP SWS IEEE1722TransportLayer

The IEEE1722Tp module support a subset of the AVTP stream data subtypes specified by [1, IEEE1722]:

- audio and video streaming: AAF, RVF, 61883_IIDC
- distribution of a generated clock rate provided by a media clock: CRF
- transport of encapsulated bus frames (ACF_CAN, ACF_CAN_BRIEF and ACF_LIN) via an ACF-stream, where the time-synchronous TSCF or the non-timesynchronousNTSCF AVTP stream data subtype is used in the ACF-header.

The IEEE1722Tp module is responsible to forward 1722Tp streams from the lower layers to stream data consumers, and from stream data producer to the lower layers. The time synchronous handling of the transported data with respect to the given AVTP presentation time is in responsibility of the according stream data consumer. Thus, the IEEE1722Tp module cannot ensure time synchronous handling with the accuracy of the AVTP presentation time in units of nanoseconds.

An ACF-stream with ACF-header set to TSCF (time-synchronous control format) carries an AVTP presentation time. The AVTP presentation time is given in units of nanoseconds. The IEEE1722Tp module can only perform a forwarding of bus frames with a resolution accuracy of the main function period (e.g. 5 ms). Please note, synchronicity of forwarded bus frames across multiple bus cluster highly depends on the surrounding infrastructur and software implementation, e.g. internal data processing, accuracy of the synchronized global time, busload.

4.12 CP SWS FlexRayTransceiverDriver

[SWS_FrTrcv_00231]
Upstream requirements: SRS_Fr_05138

The FlexRay Transceiver Driver shall use the APIs of underlying DIO drivers synchronously.

[SWS_FrTrcv_00433]
The FlexRay Transceiver Driver should use the APIs of underlying SPI drivers synchronously if possible and asynchronously where required.

[SWS_FrTrcv_00441]
The FlexRay transceiver requires a LEVEL 2, Enhanced (Synchronous/Asynchronous) SPI Handler/Driver

[SWS_FrTrcv_00238]
Upstream requirements: SRS_Fr_05152
The FlexRay Transceiver Driver shall handle the transceiver-specific timing requirements internally.

4.13 CP SWS FlexRayISOTransportLayer

The AUTOSAR architecture defines communication system specific transport protocol layers (FrTp, CanTp, LinTp etc.). Thus, the FlexRay Transport Protocol layer (FrTp) only covers FlexRay transport protocol specifics. The FlexRay Transport Protocol has an interface to a single underlying Linklayer SDU Router and a single upper PDU Router. All reported DTCs are set as tested (notTestedThisCycle = 0) with each start-up. This would result in a toggling DTC for permanent issues, increasing the Occurrence and/or OperationCycle Counters (if configured) with each power cycle.

4.14 CP SWS FlexRayDriver

[SWS_Fr_00449]
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.

4.15 CP SWS EthernetTransceiverDriver

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

4.16 CP SWS CoreTest

A Core test module implementation might be limited to be executed during power-up/start-up time where core resources are not shared among different active AUTOSAR related software tasks or hardware-entities (e.g. IRQ-controller, DMA, Cache, MMU/MPU and MemoryIF)
OR
might be limited to test resources which are not shared during runtime software execution (e.g. ALU and CPU-registers). This is overall automotive system architecture dependent and cannot be covered in a MCAL Core Test SWS specification.

There must be a managing entity or architecture available who manages tasks like 'hardware-resource-access-managing' due to the inability of a MCAL-driver to handle such tasks on its own.

4.17 CP SWS BusMirroring

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 bus 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 XL to CAN FD, CAN FD to CAN, CAN to LIN, FlexRay to CAN or CAN FD, 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 re-

ported 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.

4.18 CP RS CoreTest

Due to 4.1.3.1 and 4.1.3.2, a Core test implementation might be limited to be executed during power-up/start-up time where core resources are not shared among different active application tasks or entities (e.g. IRQ-controller, DMA) -OR- might be limited to test resources with are not shared during runtime (e.g. CPU itself).

4.19 CP EXP BSWDistributionGuide

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:

- Communication between master and satellites is not standardized.
- BSW functional clusters and their AUTOSAR BSW Cluster Interface are not standardized.

4.20 CP SWS MACsecKeyAgreement

The AUTOSAR MACsec implementation currently has the following limitations:

- Only participants authentication based on Connectivity Association pre-shared keys (CAKs) is supported (EAP-TLS, EAP-IKEv2, and other variants are not supported).
- Only MACsec between direct peers is supported (e.g. Point-to-Point configurations).

- Point-to-Multipoint configurations are not supported.
- In-service upgrades with EAPoL-MKA frames are not supported.
- Temporary suspension of MKA operation is not supported.
- MACsec Cipher Suites is the only currently supported EAPoL-Announcement TLV. The following EAPoL Announcements are currently not required:
  - Access Information (TLV Type 111)
  - Key Management Domain (TLV Type 113)
  - NID (TLV Type 114).
- Dynamic Key Server election based on Key Server priority is not supported (roles are set per configuration and fixed).

5 Release History

5.1 Release R26-11

Specification Name	Specification history entry
CP EXP AIADASAndVMC	• Freeze document
CP EXP AIBodyAndComfort	• No content changes
CP EXP AIChassis	• Splitting of the former chapter 2 into new chapters 2, 3 , and 4 • Editorial corrections
CP EXP AIHMIMultimediaAndTelematics	• No content changes
CP EXP AIOccupantAndPedestrianSafety	• No content changes
CP EXP AIPowertrain	• No content changes
CP EXP AIUserGuide	• Complete rework of all chapters - Part 2 • Milestone propagation added • Editorial corrections
CP EXP ARTI	• Removed `ArtiOsMessageContainer` • Changed name of referenced document *ASAM ARTI* to *ASAM TDF* • Editorial changes
CP EXP ApplicationLevelErrorHandling	• No content changes
CP EXP BSWDistributionGuide	• Updated description of large data path from LdCom to Com • Added constraint for asynchronous DMA copy operations • Added hints for configuring the BswM per partition and multi-core use of BSW modules
CP EXP CDDDesignAndIntegrationGuideline	• No content changes
CP EXP ErrorDescription	• No content changes
CP EXP FirmwareOverTheAir	• No content changes
CP EXP FunctionalSafetyMeasures	• Removed RS Features and RS TimingExtensions
CP EXP LayeredSoftwareArchitecture	• Removed several modules: HWTestManager, V2X modules (including their drivers), FlexRayARTransportLayer • Smaller clarifications and corrections (terminology, updates to diagrams)
CP EXP MacroEncapsulationofInterpolation Calls	• No content changes
CP EXP ModeManagementGuide	• Added Dynamic PNC for CAN Support. • Editorial changes.
CP EXP NVDataHandling	• No content changes
CP EXP SafetyUseCase	• Editorial changes
CP EXP SwClusterDesignAndIntegration Guideline	• Minor corrections / clarifications / editorial changes
CP EXP UtilizationOfCryptoServices	• No content changes
CP RS ADCDriver	• No content changes
CP RS BSWGeneral	• Added requirement for handling of corrupted NV data ([SRS_BSW_00497]) • Added requirement for double initialization errors ([SRS_BSW_00498]) • Updated MISRA reference to MISRA C 2025 ([SRS_BSW_00007]) • Removed requirements for extended production errors
CP RS CoreTest	• No content changes



△

Specification Name	Specification history entry
CP RS DIODriver	• Editorial changes
CP RS Ethernet	• Removal of V2X support • Set requirement statuses to valid
CP RS FirmwareOverTheAir	• No content changes
CP RS FlashTest	• No content changes
CP RS FlexRay	• No content changes
CP RS FunctionInhibitionManager	• New mechanism in FIM added to track and store the specific diagnostic events that cause the inhibition of a functionality (FID). Events are stored in the order of detection. • Editorial Changes
CP RS GPTDriver	• No content changes
CP RS I2CDriver	• No content changes
CP RS ICUDriver	• No content changes
CP RS IOHWAbstraction	• No content changes
CP RS MCUDriver	• No content changes
CP RS MemoryHWAbstractionLayer	• No content changes
CP RS MemoryServices	• No content changes
CP RS ModeManagement	• Dropped use case for alarm clock support • Set watchdog related requirements to obsolete • Editorial changes
CP RS OCUDriver	• No content changes
CP RS OS	• No content changes
CP RS PWMDriver	• No content changes
CP RS PortDriver	• No content changes
CP RS RAMTest	• No content changes
CP RS RTE	• Editorial changes
CP RS SPALGeneral	• No content changes
CP RS SPIHandlerDriver	• No content changes
CP RS SWCModeling	• No content changes
CP RS SoftwareClusterConnection	• No content changes
CP RS TimeService	• No content changes
CP RS Transformer	• No content changes
CP RS WatchdogDriver	• No content changes
CP SWS ADCDriver	• Adc_GetVersionInfo API removed • Chapter 8 restructured according new template • Interface cleanup Adc_Externals.h introduced • ECUC configuration parameter updates
CP SWS ARTI	• Removed GetVersionInfo API • Added more descriptive information • Removed references to requirement specification • Corrected ARXML examples

▽

△

Specification Name	Specification history entry
CP SWS BFXLibrary	• Removed API Bfx_GetVersionInfo • Removed SWS_Bfx_00206, SWS_Bfx_00207 and SWS_Bfx_00208 • Corrected Sync/Async and Reentrancy of Bfx_ShiftBitSat • Editorial changes
CP SWS BSWGeneral	• Simplification of Multiplicity Configuration Class and Value Configuration Class • Support for DMA transfer (draft) • Support for MISRA C:2025 • Removal of version check and extended production errors
CP SWS BSWModeManager	• Removed support for some RTE lifecycle APIs (e.g. BswMRteStart, BswMPartitionRestarted, BswMCoreHaltMode ...) • New diagram showing BswM Service Ports • Editorial Changes
CP SWS BSWMulticoreLibrary	• General cleanup of library requirements
CP SWS BulkNvDataManager	• Removed extended production errors • Removed BndM_GetVersionInfo API • Editorial changes
CP SWS BusMirroring	• Moved protocol definition to [3, FO PRS Bus Mirroring Protocol] • Improved description of `Mirror_TxConfirmation` and clarified transmission deadline timing • Removed `Mirror_GetVersionInfo` and `MirrorVersionInfoApi` and multi-partition related specification • Adapted to current template and improved EcuC tables, removed references to RS Bus Mirroring
CP SWS CANDriver	• Move EcuC Parameter [ECUC_Can_00485] "CanFdPaddingValue" from container CanHardwareObject [ECUC_Can_00324] to Can Controller [ECUC_Can_00354] • Remove Chapeter 8.3.1.2 related to API Can_GetVersionInfo [SWS_Can_00224] and [SWS_Can_00177] • Remove Chapters 10.2.12 CanXLGeneral [ECUC_Can_00524] 10.2.13 CanXLController [ECUC_Can_00499] 10.2.14 Can XLHardwareObject [ECUC_Can_00526] 10.2.15 CanXLHwFilter [ECUC_Can_00528] 10.2.16 CanXLBaudrateConfig [ECUC_Can_00512] 10.2.17 CanXLEthEgressFifo [ECUC_Can_00502] 10.2.18 CanXLEthIngressFifo [ECUC_Can_00509] • Remove of RS Can Specifications and RS_Can_Xxx requirements. • Introduce CANsec
CP SWS CANInterface	• CANsec Support • Support of zero-copy operation • Editorial changes
CP SWS CANNetworkManagement	• Remove Version Info • RS NM Removal • Configuration Classes and Variants simplified • Editorial Changes
CP SWS CANStateManager	• Removed GetVersionInfo-API • Removed extended production errors • Restructured chapter 8 (API specification)

▽

△

Specification Name	Specification history entry
CP SWS CANTransceiverDriver	• Removed GetVersionInfo API • Changed CANTRCV_E_BUS_ERROR to a production error and removed the chapter "Extended Production Errors" • Specified waiting for the configured CanTrcvWaitTime after hardware mode change requests • Restructured chapter 8 (API specification)
CP SWS CANTransportLayer	• Changed the extended production errors into runtime errors • Removed the configuration parameter CanTpVersionInfoApi of the CanTp_GetVersionInfo API
CP SWS CANXLDriver	• CANsec Support • Support of zero-copy operation • Editorial changes
CP SWS CANXLTransceiverDriver	• Removal of tracing to CP RS CAN requirements • Editorial changes
CP SWS COM	• Support DMA transfer approach • Removed ComIPduGroupGroupRef • Removed GetVersionInfo API • Minor corrections / clarifications / editorial changes
CP SWS COMBasedTransformer	• Removed ComXf_GetVersionInfo API
CP SWS COMManager	• Removed extended production errors • Removed ComM_GetVersionInfo API • Harmonized term for PNC request bit vector • Clarified handling for corrupted NV data
CP SWS CRCLibrary	• Removed GetVersionInfo API • Minor corrections / clarifications / editorial changes
CP SWS CommunicationStackTypes	• `PduLengthType` and `PduIdType` are no longer configurable • Removed references to RS COM • Removed section "Extended Production Errors"
CP SWS CoreTest	• Removed the CorTst_GetVersionInfo API and its configuration parameter. • Classified CORTST_E_CORE_FAILURE as a production error and updated the related reporting requirements. • Aligned the API chapter structure with the updated SWS template. • Updated configuration information and traceability.
CP SWS CryptoDriver	• Add IdsM_ReportSecurityEvent to Optional interfaces [4] 8.5.3. • Update `CryptoPrimitives` containers in Chapter 10.2.15. • Update `CryptoPrimitive` and its sub-containers in Chapter 10.2.16. • Add `CryptoPrimitiveTypes` to Crypto configurations chapter 10.2.1. • Add `CryptoPrimitiveSettings` to Crypto configurations 10.2.1. • Restructure in Functional Specification Chapter [4] 7. • Update API Specification Chapter [4] 8. • Update Members per each crypto service in [SWS_Crypto_00071]. • Update Parameters for each crypto service in [SWS_Crypto_00073].

▽

△

Specification Name	Specification history entry
CP SWS CryptoInterface	• Restructure of crypto channels and primitives across the whole crypto stack. • Remove GetVersionInfo API and configuration parameters.
CP SWS CryptoServiceManager	• Restructure and streamline the configuration model • Remove queue-based job processing model (removal of CsmQueue container and related parameters) • Remove the Csm_GetVersionInfo API and its associated configuration parameter • Editorial changes
CP SWS DIODriver	• Removed GetVersionInfo API
CP SWS DMADriver	• Initial release
CP SWS DataDistributionService	• Added DMA copy support
CP SWS DataDistributionServiceTransformer	• Remove GetVersionInfo API
CP SWS DefaultErrorTracer	• ecuc expressions changed to ARMQL. • GetVersionInfo removed. • Editorial Changes.
CP SWS DiagnosticCommunicationManager	• Introduced optional challenge for the server and client in bidirectional authentication • Introduced optional NRC overwrite after certificate verification • API Dcm_GetVersionInfo removed • Restructured Chapter 8 regarding Function Definitions and Callback notifications • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
CP SWS DiagnosticEventManager	• Added J1939 individual-DTC clearing (DM22) and severity reporting (DM01); corrected DTC-specific lamp status (DM31). • Added OBD DTC internal data elements and clarified fault detection counter data. • Clarified PTO handling and added PID $1E reporting. • Removed event combination on retrieval and the TestFailed status-bit storage option. • Updated API, callback and notification specifications, including removal of obsolete version-information references. • Added and clarified configuration constraints for OBD relevance, event availability and diagnostic data elements. • Minor corrections and editorial changes.
CP SWS DiagnosticLogAndTrace	• Specified additional values of Std_ReturnType in chapter 8 • Removed GetVersionInfo-API • Dlt for VFB tracing chapter was corrected • Minor corrections • Editorial changes
CP SWS DiagnosticOverIP	• Alignment of connection handling and message processing with ISO 13400-2 • Editorial changes

▽

△

Specification Name	Specification history entry
CP SWS E2ELibrary	• Removed references to RS Libraries • Removed 'Extended Production Errors' chapter • Aligned chapters with CP SWS Template • Removed the <Mip>_GetVersionInfo API from chapter 8 and also reference in other parts of the document to this functionality • In SWS_E2E_91068, replaced init value
CP SWS E2ETransformer	• Aligned chapters with CP SWS Template • Updated references from CP ASWS TransformerGeneral to CP SWS TransformerGeneral • Removed the <Mip>_GetVersionInfo API from chapter 8 and related references • Changed e2eCrcError and e2eCounterError from production to runtime errors
CP SWS ECUStateManager	• Removed "Alarm Clock" feature (API, configuration and service interface) • Changed variations in service descriptions to ARMQL • Removed GetVersionInfo API • Update of configuration hash handling • Minor content changes, clarifications
CP SWS EEPROMAbstraction	• Removed Ea_GetVersionInfo and associated configuration item • Editorial changes
CP SWS EFXLibrary	• Removed API function Efx_GetVersionInfo • Changed Service ID of Efx_Div_u64s64_s64 to resolve a Service ID collision • Restructured chapter 8 "API specification"; general sections moved to chapter 7; removed chapter "Extended Production Errors" • Removed references to Requirements on Libraries
CP SWS EthernetDriver	• Added scheduled functions Eth_MainFunctionRx and Eth_Main FunctionTx • Added assignment of Ethernet controllers and ingress/egress queues to main functions and partitions • Replaced interrupt configuration by polling configuration per ingress/ egress queue • Removed Eth_Receive, Eth_TxConfirmation and the ingress queue handler function • Removed EthEcucPartitionRef and Eth_RxStatusType • Replaced parameter Priority by QueueIdx in Eth_ProvideTxBuffer and Eth_ImmediateTransmit • Moved egress queue sorting to EthIf and removed the related runtime errors ETH_E_UNKNOWN_EGRESS_PRIORITY and ETH_E_NO_ MATCHING_EGRESS_QUEUE_IDENTIFIED • Reworked transmission request with direct data provision: added API parameter AsyncBuffers, replacing the configuration of hardware supported data transfer • Restricted the buffer index returned by Eth_ProvideTxBuffer to a compact value range to allow efficient buffer lookup • Specified transmit confirmation handling when the controller is set to ETH_MODE_DOWN • Added Eth_SetAndGetPhcCorrection to support the Virtual Free-Running Clock (VFRC) of EthTSyn

▽

▽

△

Specification Name	Specification history entry
	△ • Reclassified extended production errors (e.g. ETH_E_CRC) as runtime errors • Added runtime error ETH_E_TRANSMISSION_ETHERNET_FRAME_FAILED • Added development error ETH_E_INV_QUEUE_IDX • Removed development error ETH_E_HW_SUPPORTED_DATA_TRANSFER_REJECTED • Eth_MacSecResetRxSc deletes the receive secure channel instead of resetting it to default values • Frame preemption shall not be enabled together with MACsec • Added configuration parameter EthCtrlRequiredAlignment for the memory alignment of source buffers used in hardware supported data transfers • Added configuration parameter EthCtrlDuplexMode for the duplex mode of 10BASE-T1S Ethernet controllers • Removed controller mode ETH_MODE_ACTIVE_TX_OFFLINE • Removed API Eth_GetVersionInfo • Removed API Eth_GetIngressTimeStamp
CP SWS EthernetInterface	• Added the BSW Distribution Improvements for Ethernet • Added the support of distributing EthIfControllers and Ingress/Egress queues to different OS partitions • Added new develoment error ETHIF_E_INV_QUEUE_IDX • Added new Runtime error ETHIF_E_NO_MATCHING_EGRESS_QUEUE • Added parameter QueueIdx to EthIf_RxIndication and EthIf_TxConfirmation • Added IdsM_ReportSecurityEvent as an optional interface • Added new constraint to Exclude host port and uplink ports from MACsec protection on a point to point connection • Added Virtual Free Running Clock Specification • Added Eth_PhcCorrectionType and Eth_PhcCorrectionResultType as Imported Types • Added Eth_SetAndGetPhcCorrection as Option interfaces • Added EthIf_SetAndGetPhcCorrection as Triggers correction of the PHC used as reference time for the given controller and clock unit, and retrieves the performed correction • Added contraints to ensure that the same PduBufferAlignment is used by the connected PDUs • Added definition for the parameters EthIfGetArlTableAPI, EthIfStoreConfigurationApi, EthIfResetConfigurationApi • Added CtrlIdx as input parameter of EthIf_SwtEthTxAdaptBufferLength to support multiple switch drivers • Updated the Evaluation of transmission request with direct data provision to use the new parameter EthIfFrameTxPduDeferredForwarding • Update Reentrancy for EthIf_GetCurrentTimeTuple to support the calls from different context • Update Reentrancy for Transceiver-related APIs to support different TrcvIdx context • Remove V2X references from Ethernet Interface ▽

▽

△

Specification Name	Specification history entry
	△ • Removed Eth_DataType from Imported Type and replaced Eth_Data Type by uint8 in affected APIs • Removed chapters: – Communication control – Extended Production Errors – Callback notification • Editorial changes
CP SWS EthernetStateManager	• COMM_SILENT_COMMUNICATION removed • GetVersionInfo() removed
CP SWS EthernetSwitchDriver	• Further revised MacSec support • Eth_DataType, GetVersionInfo() removed • Bugfixes (types, VLAN handling, etc.)
CP SWS EthernetTransceiverDriver	• Removed EthTrcv_GetVersionInfo API • Corrected reentrancy of EthTrcv APIs • Removed Extended Production Errors • Bugfixes
CP SWS Firewall	• Removed GetVersionInfo API • IdsM related interface and imported types are listed again • Removed upwards tracing to `FO_RS_Firewall` • Removed extended production errors and restructured chapter 8
CP SWS FlashEEPROMEmulation	• Removed Fee_GetVersionInfo and associated configuration item • Editorial changes
CP SWS FlashTest	• ECUC items; restructured • ECUC items; Loadable and selectable post build values introduced • ECUC items; Pre-compile option removed in case of several allowed configuration classes • Extendended production error changed to production error • API function FlsTst GetVersionInfo removed
CP SWS FlexRayDriver	• Changed `FR_E_LPDU_SLOTSTATUS` to a production error and `FR_E_CTRL_TESTRESULT` to a runtime error • Fixed return type of `Fr_CheckTxLPduStatus()` • Removed `Fr_GetVersionInfo()` and `FrVersionInfoApi` • Adapted to current template and improved EcuC tables
CP SWS FlexRayISOTransportLayer	• Deleted GetVersionInfo API • Reclassifcation of Ext. Prod. Errors
CP SWS FlexRayInterface	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation • Editorial layout change of chapter 10 tables
CP SWS FlexRayNetworkManagement	• Remove Version Info • RS NM Removal • Configuration Classes and Variants simplified • Editorial Changes
CP SWS FlexRayStateManager	• Remove Version Info • Configuration Classes and Variants simplified • Editorial Changes

▽

△

Specification Name	Specification history entry
CP SWS FlexRayTransceiverDriver	• VersionInfo API removed • Extended Prod. Errors reclassified
CP SWS FunctionInhibitionManager	• New mechanism in FIM added to track and store the specific diagnostic events that cause the inhibition of a functionality (FID). Events are stored in the order of detection. • Data Type for CS Interface FunctionInhibitionCallback corrected to Fi M_FidStatusChangeType • Functionality FiM_GetVersionInfo completely removed • Editorial changes: Added missing links, Restructure of Chapter 8 regarding Function Definitions and Callback notifications, Removed Chapter 7.3.4 Extende Production Errors
CP SWS GPTDriver	• Removed GetVersionInfo API • Aligned the API chapter structure with the updated SWS template • Changed the extended production errors into production errors
CP SWS I2CDriver	• Specification items and constraints promoted from draft to valid • API I2C_GetVersionInfo and configuration parameter I2CVersionInfo Api removed • Data type I2C_DataPtrType removed; I2C_SetupEB uses uint8 pointers • Editorial fixes
CP SWS ICUDriver	• Removed the `Icu_GetVersionInfo` API and the `IcuGetVersion InfoApi` configuration parameter. • Updated the error classification and API chapter structure. • Clarified ECU configuration table information and updated diagram formatting. • Corrected requirement traceability and non-applicable upstream references.
CP SWS IEEE1722TransportLayer	• API cleanup • RS dependency removed • Editorial changes
CP SWS IFLLibrary	• Removed API function Ifl_GetVersionInfo • Restructured chapter 8 "API specification"; interpolation routine definitions and general sections moved to chapter 7 • Removed chapter "Extended Production Errors" • Removed references to Requirements on Libraries
CP SWS IFXLibrary	• Removed API Ifx_GetVersionInfo • Removed general library requirements • Corrected Service IDs of Ifx_DPSearch and Ifx_IpoCub and the description of the cuboid interpolation • Editorial changes
CP SWS IOHardwareAbstraction	• Removed the `IoHwAb_GetVersionInfo` API. • Updated notification interface header files. • Updated the API and error-classification chapter structure. • Corrected requirement traceability and editorial references.
CP SWS IPDUMultiplexer	• Added support for DMA transfer approach • Removed IpduM_GetVersionInfo() • Removed PduR bypass option • Editorial changes

▽

△

Specification Name	Specification history entry
CP SWS ISO15118Charging	• RS dependency removed • Editorial changes
CP SWS IntrusionDetectionSystemManager	• Introduced new Security Event IdsM Statistics • Added Post Build Selectable Support • Tracing to Requirements Specification IntrusionDetectionSystem removed
CP SWS KeyManager	• Added optional IdsM interface. • Removed Extended Production Errors chapter. • Added support for retrieval of content in case of unverified certificates. • Restructured chapter 8.
CP SWS LINDriver	• Removed GetVersionInfo API and parameters • Added for `Lin_MainFunction` supporting polling mode • Editorial Changes
CP SWS LINInterface	• Resolved inconsistencies between ComM, LinSM and LinIf (both Master and Slave nodes) • Removed GetVersionInfo API and parameters • Changed the error category for Extended Production Errors to Runtime Errors • Added [SWS_LinIf_CONSTR_00903] for clarification. • Editorial changes
CP SWS LINStateManager	• Resolved inconsistencies between ComM, LinSM and LinIf (both Master and Slave nodes) • Removed GetVersionInfo API and parameters • Editorial Changes
CP SWS LINTransceiverDriver	• Removed GetVersionInfo API
CP SWS LSduRouter	• Removed version check API • Removed extended production errors • Removed uptraces to SRS_GTW_* requirements
CP SWS MACsecKeyAgreement	• Support of CANsec key agreement: new callback functions, the Mka_CanSecConfigType datatype and related configuration parameters • New requirements and constraints for ICV calculation at transmission side and ICV validation at reception side, including handling of the MAC source address via meta data • IdsM security event reporting: IdsM_ReportSecurityEvent added as optional interface, IdsM imported types added, and security event context data definitions revised • Revision of the ECUC configuration model and editorial changes
CP SWS MCUDriver	• Aligned chapter 8 structure with the CP SWS template: – Moved content of chapter "Callback notifications" to chapter "Function definitions" and removed chapter "Callback notifications" – Added subchapter "Callout interfaces" to chapter "Expected interfaces" – Removed API `Mcu_GetVersionInfo` and its references • Removed chapter "Extended Production Errors"; [SWS_Mcu_00053] `MCU_E_CLOCK_FAILURE` is now specified as Production Error

▽

△

Specification Name	Specification history entry
CP SWS MFLLibrary	• Removed API Mfl_GetVersionInfo and the obsolete or deprecated APIs Mfl_arcSin_f32, Mfl_arcCos_f32, Mfl_arcTan_f32, Mfl_RampCalc Switch_f32, Mfl_DeadTime_f32_f32, Mfl_DebounceSetparam and Mfl_CtrlSetLimit • Removed general library requirements • Corrected API definitions: parameter directions of Mfl_PCalc and Mfl_ PIDTyp2Calc, value ranges, out-of-bound write notes and Service ID of Mfl_Cos_s32_f32 • Editorial changes
CP SWS MFXLibrary	• Removed API function Mfx_GetVersionInfo • Restructured chapter 8 "API specification"; general sections moved to chapter 7 • Removed chapter "Extended Production Errors" • Removed references to Requirements on Libraries
CP SWS MSFLibrary	• Removal of Msf_GetVersionInfo API and chapter "Extended Production Errors"
CP SWS MemoryAbstractionInterface	• Removed MemIf_GetVersionInfo and associated configuration item • Editorial changes
CP SWS MemoryAccess	• Restructured the API specification chapter • Removed the Extended Production Errors chapter
CP SWS MemoryDriver	• Added support for external memory using I2C • Added configuration parameter to indicate if the driver is used in the HSM or in the host • Removed API Mem_GetVersionInfo and the extended production errors • Editorial changes
CP SWS MemoryMapping	• Removal of DebugTraceProfile
CP SWS NVRAMManager	• Removal of NvM_GetVersionInfo • Updates to error reporting • Restructuring of the API specification chapter • Editorial corrections
CP SWS NetworkManagementInterface	• Added support for Dynamic PNC for CAN • Removed Nm_GetVersionInfo() API • Removed RS NM references • Editorial changes
CP SWS OCUDriver	• Removed GetVersionInfo API • Editorial changes
CP SWS OS	• Removal of draft state from spec items/APIs • Several ARTI updates • Minor correction / clarification / editorial changes
CP SWS PDURouter	• Removed RS Gateway references • Added support for DMA transfer approach • Removed PduR_GetVersionInfo() API • Editorial changes

▽

△

Specification Name	Specification history entry
CP SWS PWMDriver	• ECUC items; restructured • ECUC items; Loadable and selectable post build values introduced • ECUC items; Pre-compile option removed in case of several allowed configuration classes • API function PWM GetVersionInfo removed
CP SWS PlatformTypes	• Definition of TRUE/FALSE changed to avoid MISRA violations.
CP SWS PortDriver	• Editorial changes • Removed [ECUC_Port_00133], [SWS_Port_00143] and [SWS_Port_00225]
CP SWS RAMTest	• ECUC items; restructured • ECUC items; Loadable and selectable post build values introduced • ECUC items; Pre-compile option removed in case of several allowed configuration classes • Extendended production errors changed to production errors • API function RamTst GetVersionInfo removed
CP SWS RTE	• Rapid prototyping: support of direct buffer access method removed • Various corrections / clarifications/ editorial changes
CP SWS SAEJ1939DiagnosticCommunication Manager	• Support for `DM22` • DTCs of `DM01` are now sorted by priority • Introduced generic message handling override for supported DMx messages • Removed `J1939DcmSubElementInImplDataElementInstance` and `J1939DcmSubElementInImplDataElementInstanceRef` and updated `DataCondition` accordingly • Removed `J1939Dcm_GetVersionInfo` and `J1939DcmVersionInfoApi`, adapted to current template and improved EcuC tables
CP SWS SAEJ1939FunctionalSafetyComm Protocol	• Add parameters and functions to control enabling/disabling functional safety processing. • Remove J1939Fscp_GetVersionInfo • Reorganize Chapter 8
CP SWS SAEJ1939NetworkManagement	• Changed `J1939NM_E_ADDRESS_LOST` to a production error • Improved description of `J1939Nm_TxConfirmation` • Removed `J1939Nm_GetVersionInfo` and `J1939NmVersionInfoApi` • Adapted to current template and improved EcuC tables, removed references to RS SAE J1939
CP SWS SAEJ1939RequestManager	• Improved description of `J1939Rm_TxConfirmation` • Removed `J1939Rm_GetVersionInfo` and `J1939RmVersionInfoApi` • Adapted to current template and improved EcuC tables, removed references to RS SAE J1939
CP SWS SAEJ1939TransportLayer	• Fixed transmission diagrams • Improved descriptions of `J1939Tp_TxConfirmation()` and `PduR_J1939TpRxIndication()` • Removed `J1939Tp_GetVersionInfo()` and `J1939TpVersionInfoApi` • Adapted to current template and improved EcuC tables, removed references to RS SAE J1939

▽

△

Specification Name	Specification history entry
CP SWS SOMEIPTransformer	• Added security-event reporting and context-data definitions for SOME/IP processing errors, along with new rules for string length-field insertion, length-field typing, serialized string payload sizing, and maximum variable-array elements. • Updated fixed-length string serialization and deserialization behavior, including BOM handling, length calculation, padding, and configuration-dependent defaults. • Refined development-error classification, security-event context definitions, and imported datatype/API specifications for improved consistency with related modules.
CP SWS SOMEIPTransportProtocol	• Added explicit reception handling for unsegmented SOME/IP messages and single-segment TP messages when all configured receive N-SDUs are occupied. • Clarified reporting of receive-resource limitations and the associated runtime condition. • Refined ECUC container definitions, imported datatypes, development-error classification, reception processing, and N-SDU state handling. • Removed obsolete configuration and behavioral specifications to align the document with the updated SOME/IP-TP behavior.
CP SWS SPIHandlerDriver	• Removed GetVersionInfo API • Aligned the API chapter structure with the updated SWS template • Changed the extended production errors into production errors
CP SWS SecureOnboardCommunication	• Added Authenticated Encryption of Confidential I-PDUs • Added asynchronous copying of I-PDUs into the local buffer • Added option to confirm the transmission to the upper layer immediately • Minor corrections / clarifications / editorial changes; For details please refer to the Change Documentation
CP SWS ServiceDiscovery	• Added security-event handling for unavailable services, including standardized context data for unauthorized offers, unavailable services, malformed SOME/IP-SD PDUs, unauthorized subscriptions, and unauthorized method invocations. • Updated security-event configuration and reporting definitions, including the security-event reference container and related IdsM integration details. • Refined Service Discovery behavior for malformed message validation, eventgroup subscription handling, connection readiness checks, ACL enforcement, runtime error reporting, and mandatory interface definitions. • Removed obsolete configuration and specification content related to legacy development-error handling, timing parameters, and superseded service-state behavior.
CP SWS SocketAdaptor	• Added support for secure UDP communication using DTLS • Added support for DMA-based data transfer • Removed the configuration switch for the version information API • Restructured the API specification and removed obsolete chapters
CP SWS SoftwareClusterConnection	• Set requirement statuses to valid • Clarify implementation aspects of Proxy Modules • Corrections and editorial changes
CP SWS StandardTypes	• Std_VersionInfoType removed.

▽

△

Specification Name	Specification history entry
CP SWS SynchronizedTimeBaseManager	• Virtual Free-running Clock (VFRC) added • APIs reentrancy attribute udpated • Chapter 8 restructured • GetVersionInfo APIs and Multicore Distribution requirements removed
CP SWS TcpIp	• Introduce IKEv2 post-quantum preshared-key support according to IETF RFC 8784 • Define TCP/IP reception and transmission buffer lifecycle, including synchronous and asynchronous copying • Add PDU buffer alignment, immediate transmission acknowledgments, UDP/TCP transmit header pointers, and PDU-state error handling
CP SWS TimeService	• Removed GetVersionInfo API • Aligned the API chapter structure with the updated SWS template • Changed the extended production errors into production errors
CP SWS TimeSyncOverCAN	• Chapter 8 restructured • CanTSyn_GetVersionInfo removed
CP SWS TimeSyncOverEthernet	• Virtual Free Running Clock added • Rate deviation granularity improved • APIs reentrancy attribute udpated • Chapter 8 restructured • GetVersionInfo APIs and Multicore Distribution requirements removed
CP SWS TimeSyncOverFlexRay	• Chapter 8 restructured • FrTSyn_GetVersionInfo removed
CP SWS TransformerGeneral	• Changed document type from ASWS to SWS • Removed the <Mip>_GetVersionInfo API • Removed Extended Production Errors • Improved Security Events for Service-Oriented Communication • Removed possibility to pass transformer specific transaction handles
CP SWS UDPNetworkManagement	• Removal of version info API • Removal of RS NM • Simplification of configuration classes and variants • Bug fixes and editorial changes
CP SWS VDPCMRemote	• Removed function *VdpCmR_GetVersionInfo()*. • Restructured chapter 8 *API specification* according to new AUTOSAR template
CP SWS WatchdogDriver	• Removed GetVersionInfo API. • Changed the error classification from Extended Production Errors to Runtime Errors.
CP SWS WatchdogInterface	• Removed GetVersionInfo API and parameters • Editorial changes
CP SWS WatchdogManager	• Removed GetVersionInfo API and parameters • Changed Dem Events from Post-Build to Pre-Compile. • Editorial changes
CP SWS XCP	• Support DMA transfer approach • Removal of RS dependency • Editorial changes
CP TPS BSWModuleDescriptionTemplate	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

▽

△

Specification Name	Specification history entry
CP TPS DiagnosticExtractTemplate	• Retire unused encodings • minor corrections / clarifications / editorial changes
CP TPS ECUConfiguration	• PostBuild ConfigClass simplification [CONC 739] • Simplify Uri Reference (remove destinationUriNestingContract) • Editorial: Move chapter on service needs from appendix to main text • Editorial: Introduce imposition times for constraints • Minor changes
CP TPS ECUResourceTemplate	• No content changes
CP TPS SoftwareComponentTemplate	• Remove V2X Support • Improve description of record layouts • Improve description of data dependencies • various corrections / clarifications / editorial changes
CP TPS SystemTemplate	• removed FrArTp • Added support for Connectivity Association and Secure Channel for MACsec and CANsec • Added support for TimeBase Cloning • removed SenderReceiverCompositeElementToSignalMapping
CP TPS TimingExtensions	• See appendix [5] G.19
CP TR AIDesignPatternsCatalogue	• No content changes
CP TR AIMeasurementCalibrationDiagnostics	• No content changes
CP TR BSWUMLModelModelingGuide	• Updated taged lifecycle information of model elements ([TR_BSWMG_00702]) • Removed modeling approach of “abstract interfaces” for configuration dependent interface classes ([TR_BSWMG_00917]) • Described modeling of diagrams as specification items ([6] 2.12.4) • Minor clarifications
CP TR GeneralBlueprintsSupplement	• Update list of delivered files
CP TR Methodology	• Expand explanation of extracts and add additional extracts (e.g. Security Extract)
CP TR ModelingShowCases	• No content changes
CP TR ReleaseOverview	• Release Life Cycle Status: R26-11 is in Evolution, R26-11 supersedes R25-11
CP TR SWCModelingGuide	• Updated MISRA reference to MISRA C2025
CP TR VFB	• Editorial changes: converted bibliography entries of AUTOSAR documents from STD-CAT-DocName to STD_CAT_DocName
CP TR WorkflowExample	• Implement second iteration including gateway for CAN / ETH

Table 5.1: Overview of specification changes in R26-11