Document Title	Adaptive Platform Release Overview
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	782

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

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

Disclaimer

Table of Contents

1 Introduction 6
1.1 Scope of this document . . . 6
1.2 Terminology and Licenses . . . 6
1.2.1 Terminology statement . . . 6
1.2.2 Usage of W3C XML schema . . . 6
1.3 AUTOSAR Standards . . . 7
1.3.1 Introduction . . . 7
1.3.2 Definition . . . 7
1.3.3 Overview on AUTOSAR's Standards . . . 8
1.3.3.1 Adaptive Platform . . . 8
1.3.3.2 Classic Platform . . . 8
1.3.3.3 Foundation . . . 8
1.3.4 Dependencies between Standards . . . 8
1.3.5 Dependencies to other Standards . . . 9
1.4 Release Numbering and Life Cycle . . . 9
1.4.1 Platform release number . . . 9
1.4.2 Internal release number . . . 9
1.4.3 Release life cycle of a major release . . . 10
1.4.4 Life cycle states of specification items and requirements . . . 11
1.4.5 Overview of AUTOSAR schema versions and corresponding internal AUTOSAR releases . . . 11
1.4.6 Overview of AUTOSAR schema versions and corresponding valid AUTOSAR releases . . . 11
1.5 Introduction to the Adaptive Platform . . . 12
1.5.1 Release strategy . . . 12
1.5.2 Parallel validation of specification via implementation . . . 12
1.5.3 Specification depth . . . 13
1.6 Content of chapters . . . 13
2 Summary of changes 15
2.1 Release R20-11 . . . 15
2.1.1 Concepts . . . 15
2.1.1.1 Introduced Concepts . . . 15
2.1.1.2 Impact of Concepts . . . 17
2.1.1.3 Validated Concepts . . . 21
2.1.2 Specifications . . . 21
2.1.2.1 New Specifications . . . 21
2.1.2.2 Migrated Specifications . . . 21
2.1.2.3 Obsolete Specifications . . . 22
2.1.2.4 Removed Specifications . . . 22
2.1.2.5 Reworked Specifications . . . 22
2.1.2.6 Moved Specification parts . . . 22
2.1.3 Release Documentation . . . 22
2.2 History information in AUTOSAR . . . 22

3 Specification overview 24

4 Remarks to known technical deficiencies 27

4.1 Specification of Communication Management (UID 717, SWS) 27
4.2 Specification of Execution Management (UID 721, SWS) 28
4.3 Specification of Diagnostics (UID 723, SWS) 28
4.4 Specification of Platform Health Management for Adaptive Platform (UID 851, SWS) 30
4.5 Specification of Persistency (UID 858, SWS) 30
4.6 Specification of Cryptography for Adaptive Platform (UID 883, SWS . . 30
4.7 Specification of Update and Configuration Management (UID 888, SWS) 31
4.8 Specification of Network Management (UID 898, SWS) 32
4.9 Specification of Identity and Access Management (UID 900, SWS) . . 32
4.10 Specification of the Adaptive Core (UID 903, SWS) 32

5 Release history 33

5.1 Release R20-11 33

References

[1] Explanation of ara::com API
AUTOSAR_EXP_ARAComAPI

[2] E2E Protocol Specification
AUTOSAR_PRS_E2EProtocol

[3] SOME/IP Protocol Specification
AUTOSAR_PRS_SOMEIPProtocol

[4] Specification of Manifest
AUTOSAR_TPS_ManifestSpecification

[5] Specification of Execution Management
AUTOSAR_SWS_ExecutionManagement

[6] Requirements on Execution Management
AUTOSAR_RS_ExecutionManagement

[7] Unified diagnostic services (UDS) – Part 1: Application layer (Release 2020-02)
http://www.iso.org

[8] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2013-03)
http://www.iso.org

1 Introduction

1.1 Scope of this document

This document provides an overview of the AUTOSAR standard Adaptive Platform release R20-11.

1.2 Terminology and Licenses

1.2.1 Terminology statement

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

1.2.2 Usage of W3C XML schema

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

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

License

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

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

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

1.3 AUTOSAR Standards

1.3.1 Introduction

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

Packaging its deliverables into different "standards"

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

1.3.2 Definition

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

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

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

1.3.3 Overview on AUTOSAR's Standards

AUTOSAR delivers the following standards:

Standard	Abbreviation
Adaptive Platform	AP
Classic Platform	CP
Foundation	FO

1.3.3.1 Adaptive Platform

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

1.3.3.2 Classic Platform

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

1.3.3.3 Foundation

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

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

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

1.3.4 Dependencies between Standards

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

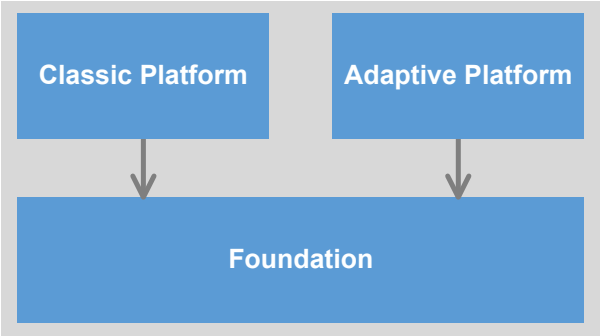

Figure 1.1: Dependencies of AUTOSAR Standards

1.3.5 Dependencies to other Standards

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

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

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

1.4 Release Numbering and Life Cycle

1.4.1 Platform release number

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

1.4.2 Internal release number

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

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

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

1.4.3 Release life cycle of a major release

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

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

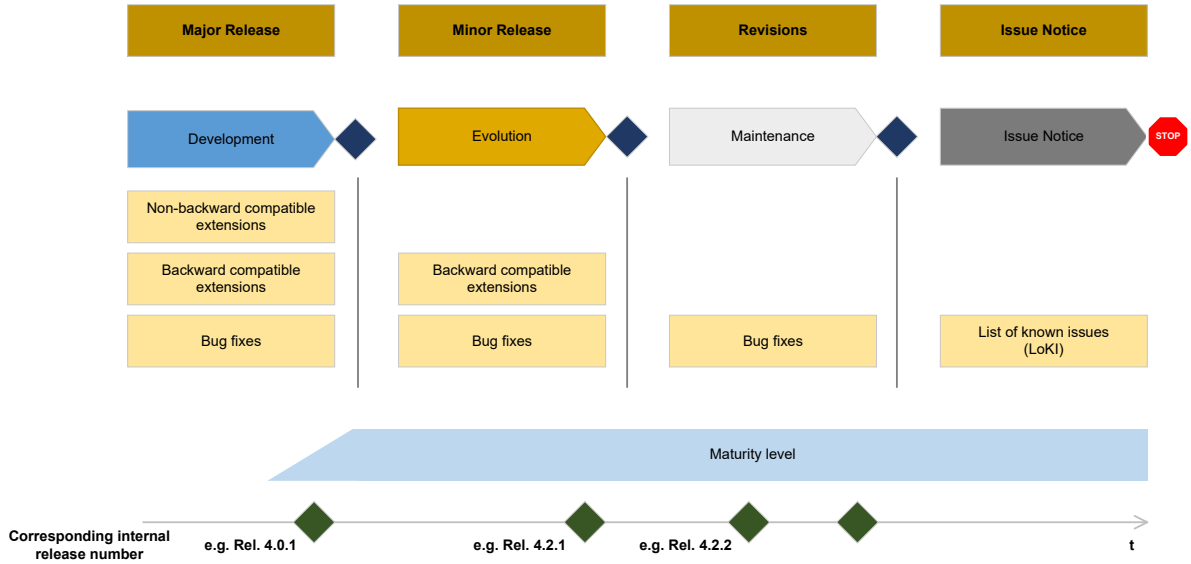

Figure 1.2: Life cycle model of AUTOSAR standards

1.4.4 Life cycle states of specification items and requirements

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

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

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

1.4.5 Overview of AUTOSAR schema versions and corresponding internal AUTOSAR releases

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

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

1.4.6 Overview of AUTOSAR schema versions and corresponding valid AUTOSAR releases

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

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

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

1.5 Introduction to the Adaptive Platform

The AUTOSAR Adaptive Platform is the standardized platform for microprocessor-based ECUs supporting use cases like highly automated driving as well as high speed on-board and off-board communication.

The Adaptive Platform differs in a number of aspects from the standardization approach of the Classic Platform:

- Parallel validation of specification via software implementation
- Specification of functional clusters instead of modules

1.5.1 Release strategy

The Adaptive Platform has changed its life cycle state to "Evolution" according to AUTOSAR's life cycle model for its standards (as depicted in chapter 1.4.3). Since R19-11, AUTOSAR releases the Adaptive Platform together with the Classic Platform and Foundation in a yearly cycle. The life cycle state "Evolution" implies that users of the Adaptive Platform have a guarantee on backward compatibility for certain parts of the specifications. The differentiation is handled by the life cycle state of the requirements and specification items according to chapter 1.4.4.

1.5.2 Parallel validation of specification via implementation

The Adaptive Platform is partially validated through an AUTOSAR-internal implementation: the Adaptive Platform Demonstrator. This Demonstrator is available to all the partners and can provide further details to understand the underlying concepts of the Adaptive Platform. The Adaptive Platform Demonstrator is an exemplary implementation of the Adaptive Platform specifications. All further usage based on the Demonstrator (e.g. in series development) will become the responsibility of the respective partner. For legal constraints see the dedicated paragraphs in the Development Agreement.

For the current releases, the Demonstrator software implementation has undergone only informal reviews with no strict quality assurance. AUTOSAR is increasing the quality assurance significantly to ensure the quality criteria given by the project.

The Demonstrator comes with traceability up to the specifications to document the validation aspect.

Additionally AUTOSAR develops System Test specifications and implementation to support the test of the demonstrator implementation against the AUTOSAR requirements. These tests are also part of the release.

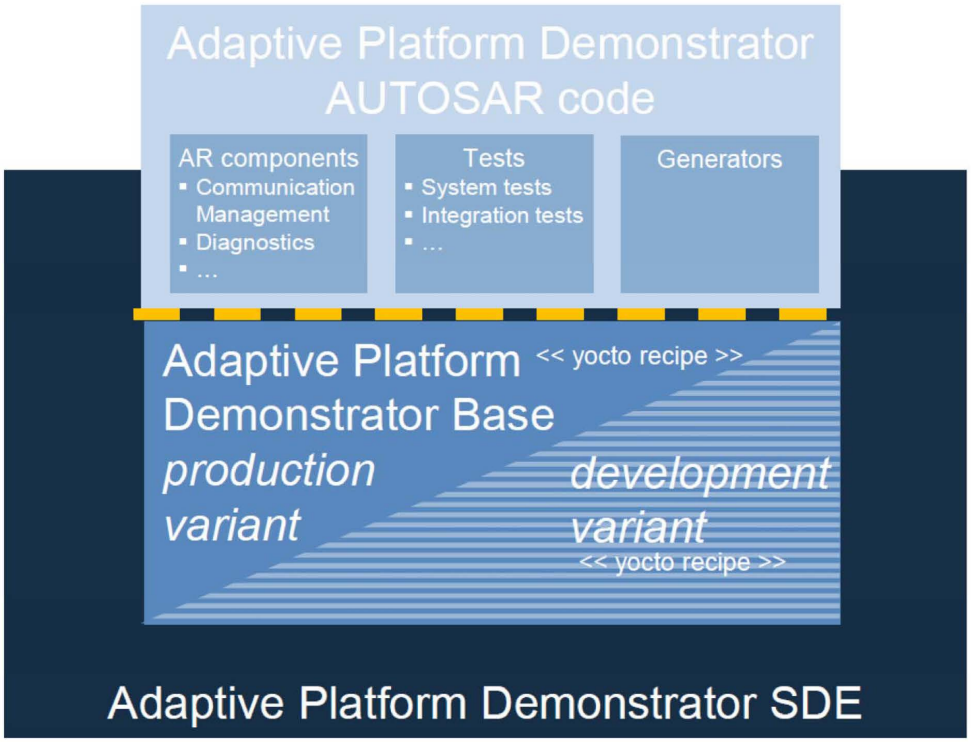

Figure 1.3: Overview of the AUTOSAR Adaptive Platform Demonstrator

1.5.3 Specification depth

Based on the development history of the Classic Platform, AUTOSAR has decided to specify functional clusters instead of a specific software architecture to provide the implementers with options to find efficient solutions for the standardized features.

1.6 Content of chapters

This document is structured as follows:

- Chapter 1 provides an introduction to AUTOSAR's release strategy, the Adaptive Platform and its standardization approach.
- Chapter 2 provides a summary of changes since the previous release of the Adaptive Platform.

- Chapter 3 contains the overview of specifications comprising the release R20-11. This chapter is structured according to the clusters of AUTOSAR release R20-11.
- Chapter 4 contains remarks about known technical deficiencies.
- Chapter 5 contains the detailed release history of all released specifications.

2 Summary of changes

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

2.1 Release R20-11

Several concepts affecting the Adaptive Platform have been introduced with release R20-11 thereby adding functionality to the platform.
Those concepts are e.g. related to security (Integration of IAM, Crypto API, SCREIAM) or communication (10BASE-T1S, ara Communication Groups).

Additionally some concepts target the Classic and Adaptive Platform, strengthening the interaction between the two platforms.
Those concepts are related to security (Intrusion Detection System Manager) and development methodology and mechanisms (Unified Timing and Tracing Approach).

2.1.1 Concepts

2.1.1.1 Introduced Concepts

The following concepts in 2.1.1.1.1 - 2.1.1.1.10 have been introduced.

2.1.1.1.1 Vehicle Network State Management

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

2.1.1.1.2 Integration of IAM

AUTOSAR AP's Identity and Access Management is applied to Functional Clusters by the concept "Integration of IAM". For each Functional Cluster that implements an API under access control, according requirements and modelling elements are introduced by concepts parts. In the release R20-11 the concept part "Platform Health Management" is introduced.
Requirements deducted from an analysis of attack scenario in the domain of Platform Health Management have been elaborated and specified. Further concept parts will be introduced in future releases.

2.1.1.1.3 Crypto API

The concept “CryptoAPI” describes the functionality and the configuration for the Adaptive Functional Cluster Cryptography (FC Crypto) and its API (CryptoAPI), which is part of the AUTOSAR Adaptive Platform Foundation. The goal of this concept is to standardize a software API which lists cryptographic services and provides an interface for AUTOSAR applications. This concept provides a solution to allow the OEM handling encrypted and / or signed data and validating ECUs, communication partners, or services. Therefore, the FC Crypto offers applications and other Adaptive AUTOSAR Functional Cluster a standardized interface which provides operations for cryptographic and related calculations. These operations include cryptographic operations, key management, and certificate handling. The standardized interface is exposed by the CryptoAPI.

2.1.1.1.4 RS Safety

The concept of “RS Safety” aims to provide safety requirements for the AUTOSAR Adaptive Platform within a requirement specification (RS) document: RS Safety. Providing safety requirements in this form allows the derivation and detailing of safety requirements from RS Main in a generic fashion: as Functional Safety Requirements (FSRs), and targeting the platform and the respective functional clusters as Technical Safety Requirements (TSRs). The TSRs can then be traced to from the requirement specifications of functional clusters towards RS Safety.

2.1.1.1.5 10BASE-T1S

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

2.1.1.1.6 AD Sensor Interfaces

This concept is to provide well-defined sensor service interfaces as Adaptive Platform Services, which is compliant to ISO 23150 “Road vehicles - Data communication between sensors and data fusion unit for automated driving functions - Logical interface”. The reflected sensors are Radar, Lidar, Camera and Ultra-Sonic Sensors.

2.1.1.1.7 Intrusion Detection System Manager

The concept “Intrusion Detection System Manager” specifies a framework for an AUTOSAR based Intrusion Detection System (IDS). This includes the BSW compo-

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

2.1.1.1.8 Deterministic Synchronization

The concept extends the synchronization behaviors for redundant execution using DeterministicClient APIs on Adaptive Platform. It allows several redundant processes running with synchronized cycles, which also enables the possibility of integrating redundant execution with additional Software Lockstep functionality.

2.1.1.1.9 Static Configuration of Remote ECU Identity and Access Management

The concept limits the impact of compromised communicating nodes running the AUTOSAR Adaptive Platform. The goal is to prevent an attacker, who gains control of an ECU, from going beyond the ECU’s intended functionality. This paradigm is realized by SCREIAM through provisioning of access control checks on the receiver side, based on the identity of the communicating remote peers and pre-defined policies.

2.1.1.1.10 ara Communication Groups

The concept allows a set of Processes to implement a synchronized behavior. This is achieved by enabling a server within such a group of Processes to broadcast messages to all other Processes(clients) within the group and receive their answers to this request. The messages and answers are can be defined as template, from where all the necessary methods and events can be generated.
StateMangement defines the PowerModes based on the CommunicationGroup pattern to achieve a synchonized behavior between StateMangement and all running Autosar Adaptive Applications for a more flexible shutdown behavior.

2.1.1.2 Impact of Concepts

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

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

- Specifications in **bold** font are completely new specifications originating from the particular concept.

- Specifications in *italic* font are affected indirectly as they provide artefacts for the actually impacted specifications.

Concept Name	Specification Long Name	Standard	Concept Lifecycle
RS Safety	**Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform**	Foundation	draft
	Requirements on Log and Trace		
	Requirements on IPsec Protocol		
	Requirements on Health Monitoring		
	Requirements on Update and Configuration Management	Adaptive Platform	
	Requirements on Persistency		
	Requirements on Operating System Interface		
	Requirements on Execution Management		
	Requirements on Communication Management		
10BASE-T1S	Specification of Manifest	Adaptive Platform	draft
	Specification of Time Synchronization over Ethernet	Classic Platform	
	Specification of TCP/IP Stack		
	Specification of Network Management Interface		
	Specification of Ethernet Transceiver Driver		
	Specification of Ethernet Switch Driver		
	Specification of Ethernet Interface		
	Specification of Ethernet Driver		
	System Template		
	Requirements on Ethernet Support in AUTOSAR		
	Specification of Socket Adaptor		
Deterministic Synchronization	Specification of Execution Management	Adaptive Platform	draft
Integration of IAM	Specification of Platform Health Management for Adaptive Platform	Adaptive Platform	draft
Crypto API	Specification of Manifest	Adaptive Platform	draft

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Crypto API	Specification of Cryptography for Adaptive Platform	Adaptive Platform	draft
	Requirements on Cryptography		
Vehicle Network State Manager	Specification of Network Management	Adaptive Platform	draft
	Specification of UDP Network Management	Classic Platform	
	Specification of Network Management Interface		
	Specification of FlexRay Network Management		
	Specification of Communication Manager		
	Specification of CAN Network Management		
	System Template		
	Requirements on Network Management		
	Requirements on Mode Management		
	Specification of the AUTOSAR Network Management Protocol	Foundation	
	Requirements on AUTOSAR Network Management		
AD Sensor Interfaces	**Specification of Sensor Interfaces**	Adaptive Platform	draft
	Requirements on Automated Driving Interfaces		
Static Configuration of Remote ECU Identity and Access Management	Specification of Manifest	Adaptive Platform	draft
	Specification of Communication Management		
ara Communications Groups	Requirements on Communication Management	Adaptive Platform	draft
	Specification of Communication Management		
	Specification of Manifest		
Intrusion Detection System Manager	**Specification of Intrusion Detection System Protocol**	Foundation	partially validated
	Requirements on Intrusion Detection System		
	Requirements on Security Extract Template		

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Intrusion Detection System Manager	**Security Extract Template**	Foundation	partially validated
	Glossary		
	Requirements on Diagnostic Extract Template	Classic Platform	
	Requirements on AUTOSAR Features		
	Specification of Socket Adaptor		
	Layered Software Architecture		
	Specification of Key Manager		
	Specification of Secure Onboard Communication		
	General Requirements on Basic Software Modules		
	General Specification of Basic Software Modules		
	Specification of CAN Driver		
	Specification of CAN Interface		
	Specification of Diagnostic Communication Manager		
	Specification of Diagnostic Event Manager		
	Specification of Ethernet Interface		
	Specification of Intrusion Detection System Manager		
	Specification of NVRAM Manager		
	Specification of TCP/IP Stack		
	Diagnostic Extract Template		
	Software Component Template		
	List of Basic Software Modules		
	Requirements on Manifest Specification	Adaptive Platform	
	Specification of Intrusion Detection System Manager for Adaptive Platform		
	Specification of Communication Management		
	Specification of Cryptography for Adaptive Platform		

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
IdsM (Intrusion Detection System Manager)	Specification of Manifest	Adaptive Platform	partially validated

Table 2.1: Impact of Concepts

2.1.1.3 Validated Concepts

The following concept has been validated:

- Socket-Network Binding for ARA::com

2.1.2 Specifications

2.1.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Specification of Intrusion Detection System Manager for Adaptive Platform (UID 978, SWS)
- Requirements on Automated Driving Interfaces (UID 911, RS)
- Specification of Sensor Interfaces (UID 912)

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

- Explanation of Adaptive Platform Software Architecture (UID 982, EXP)
- Explanation of Adaptive Platform Software Architectural Decisions (UID 983, EXP)
- Specification of Timing Extension for Adaptive Platform (UID 968, TPS)

2.1.2.2 Migrated Specifications

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

- Meta Model (UID 59, MMOD)
- Meta Model-generated XML Schema (UID 230, MMOD)
- Supplementary material of the AUTOSAR XML Schema (UID 649, TR)
- Specification of Abstract Platform (UID 947, TPS)

2.1.2.3 Obsolete Specifications

The following specification has been set to status "obsolete" in this release:

- Guidelines for the use of the C++14 language in critical and safety-related systems (UID 839, RS)

  This specification will be released by MISRA in their future release but will still be used by AUTOSAR.

2.1.2.4 Removed Specifications

The following specification has been set to status "removed" in this release:

- General Specification of Adaptive Platform (UID 715, SWS)

2.1.2.5 Reworked Specifications

The following specifications have been changed fundamentally in R19-11

- none

2.1.2.6 Moved Specification parts

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

- The “Specification of Core Types” (UID 903, SWS) has been renamed to “Specification of the Adaptive Core” (UID 903, SWS).
- The content of “General Specification of Adaptive Platform (UID 715, SWS)” has been integrated into the “Specification of the Adaptive Core”(UID 903, SWS).

2.1.3 Release Documentation

There are no major changes in the Release Documentation.

2.2 History information in AUTOSAR

The following diagram shows the location of documentation of changes.

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

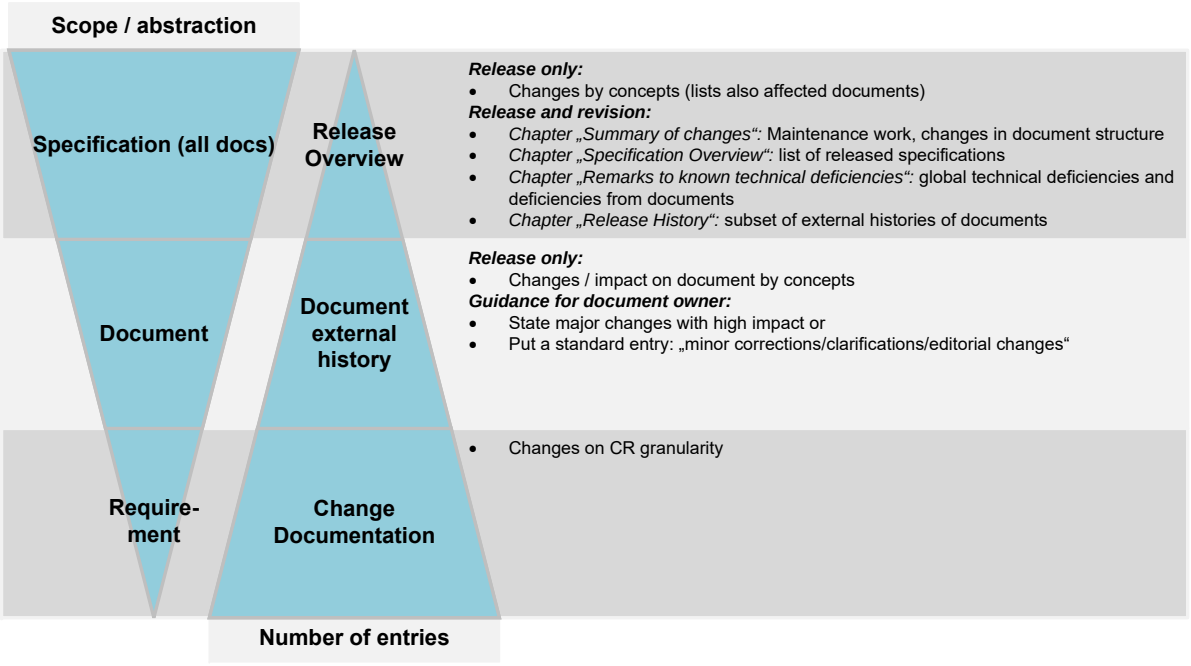

Figure 2.1: History information in AUTOSAR

3 Specification overview

The published specifications are divided into the clusters

- Release Documentation
- General
- Methodology and Manifests
- Adaptive Foundation
- Adaptive Services

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

Long Name	File Name	Life cycle changes
Release Documentation		
Adaptive Platform Release Overview	AUTOSAR_TR_AdaptivePlatform ReleaseOverview	
AUTOSAR Adaptive Platform Specification Hashes	AUTOSAR_TR_AdaptivePlatform SpecificationHashes	
General		
Design guidelines for using parallel processing technologies on Adaptive Platform	AUTOSAR_EXP_ ParallelProcessingGuidelines	
Explanation of Adaptive Platform Design	AUTOSAR_EXP_PlatformDesign	
Explanation of Adaptive Platform Software Architectural Decisions	AUTOSAR_EXP_ SWArchitecturalDecisions	Initial release
Explanation of Adaptive Platform Software Architecture	AUTOSAR_EXP_SWArchitecture	Initial release
Explanation of Safety Overview	AUTOSAR_EXP_SafetyOverview	
Functional Cluster Shortnames	AUTOSAR_TR_ FunctionalClusterShortnames	
General Requirements specific to Adaptive Platform	AUTOSAR_RS_General	
Guidelines for the use of the C++14 language in critical and safety-related systems	AUTOSAR_RS_CPP14Guidelines	obsolete
Guidelines for using Adaptive Platform interfaces	AUTOSAR_EXP_ AdaptivePlatformInterfacesGuidelines	
System Tests of Adaptive Platform	AUTOSAR_TR_ AdaptivePlatformSystemTests	
Methodology and Manifests		
Collection of blueprints for AUTOSAR Adaptive Platform M1 models	AUTOSAR_MOD_ AdaptivePlatformGeneralBlueprints	
Methodology for Adaptive Platform	AUTOSAR_TR_AdaptiveMethodology	
Requirements on Manifest Specification	AUTOSAR_RS_ManifestSpecification	
Specification of Manifest	AUTOSAR_TPS_ManifestSpecification	
Specification of Platform Types for Adaptive Platform	AUTOSAR_SWS_ AdaptivePlatformTypes	

▽

△

Long Name	File Name	Life cycle changes
Specification of Timing Extension for Adaptive Platform	AUTOSAR_TPS_AdaptivePlatformTimingExtensions	Initial release
Adaptive Foundation		
Explanation of ara::com API	AUTOSAR_EXP_ARAComAPI	
Explanation of IPsec Implementation Guidelines	AUTOSAR_EXP_IPsecImplementationGuidelines	
Requirements on Communication Management	AUTOSAR_RS_CommunicationManagement	
Requirements on Cryptography	AUTOSAR_RS_Cryptography	
Requirements on Execution Management	AUTOSAR_RS_ExecutionManagement	
Requirements on Identity and Access Management	AUTOSAR_RS_IdentityAndAccessManagement	
Requirements on Operating System Interface	AUTOSAR_RS_OperatingSystemInterface	
Requirements on Persistency	AUTOSAR_RS_Persistency	
Requirements on Platform Health Management for Adaptive Platform	AUTOSAR_RS_PlatformHealthManagement	
Requirements on Security Management for Adaptive Platform	AUTOSAR_RS_SecurityManagement	
Specification of Communication Management	AUTOSAR_SWS_CommunicationManagement	
Specification of Cryptography for Adaptive Platform	AUTOSAR_SWS_Cryptography	
Specification of Diagnostics	AUTOSAR_SWS_Diagnostics	
Specification of Execution Management	AUTOSAR_SWS_ExecutionManagement	
Specification of Identity and Access Management	AUTOSAR_SWS_IdentityAndAccessManagement	
Specification of Intrusion Detection System Manager for Adaptive Platform	AUTOSAR_SWS_AdaptiveIntrusion-DetectionSystemManager	Initial release
Specification of Log and Trace	AUTOSAR_SWS_LogAndTrace	
Specification of Operating System Interface	AUTOSAR_SWS_OperatingSystemInterface	
Specification of Persistency	AUTOSAR_SWS_Persistency	
Specification of Platform Health Management for Adaptive Platform	AUTOSAR_SWS_PlatformHealthManagement	
Specification of RESTful communication	AUTOSAR_SWS_REST	
Specification of the Adaptive Core	AUTOSAR_SWS_AdaptiveCore	
Specification of Time Synchronization for Adaptive Platform	AUTOSAR_SWS_TimeSynchronization	
Adaptive Services		
Explanation of Sensor Interfaces	AUTOSAR_EXP_SensorInterfaces	
Requirements of State Management	AUTOSAR_RS_StateManagement	
Requirements on Automated Driving Interfaces	AUTOSAR_RS_AutomatedDrivingInterfaces	Initial release
Requirements on Update and Configuration Management	AUTOSAR_RS_UpdateAndConfigManagement	

▽

△

Long Name	File Name	Life cycle changes
Specification of Network Management	AUTOSAR_SWS_ NetworkManagement	
Specification of Sensor Interfaces	AUTOSAR_SWS_SensorInterfaces	Initial release
Specification of State Management	AUTOSAR_SWS_StateManagement	
Specification of Update and Configuration Management	AUTOSAR_SWS_ UpdateAndConfigManagement	

Table 3.1: Specification Overview

4 Remarks to known technical deficiencies

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

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

Document UID	Long Name	Document Type	Section Reference
717	Specification of Communication Management	SWS	4.1
721	Specification of Execution Management	SWS	4.2
723	Specification of Diagnostics	SWS	4.3
851	Specification of Platform Health Management for Adaptive Platform	SWS	4.4
858	Specification of Persistency	SWS	4.5
883	Specification of Cryptography for Adaptive Platform	SWS	4.6
888	Specification of Update and Configuration Management	SWS	4.7
898	Specification of Network Management	SWS	4.8
900	Specification of Identity and Access Management	SWS	4.9
903	Specification of the Adaptive Core	SWS	4.10

Table 4.1: Overview of known technical deficiencies

4.1 Specification of Communication Management (UID 717, SWS)

The current version of this document is missing some functionality which is not standardized and specified within the *SWS Communication Management* document but described in *Explanation of ara::com API* [1] and implemented in the demonstrator code:

- **Local Buffer Overruns**
  Currently it is not specified what happens if local buffers are full because the application accesses data slower than they are received over the network.

The general limitations regarding E2E protection and the detectable failure modes are described in [2]. Additional, platform specific limitations regarding E2E protection are described in chapter E2EForMethodsLimitations.

The following limitations regarding optionality introduced with the Tag-Length-Value serialization principle described in [3] and [4] apply:

- **Optional method arguments**
  The Specification does not support the existence of optional method arguments.

4.2 Specification of Execution Management (UID 721, SWS)

The following functionality is mentioned within this document but is not fully specified in this release:

Section 7.7 Resource Limitation and Section 7.8 Fault Tolerance of [5] – these sections have been expanded in this release but are not complete. In particular the contents will be expanded with more properties and formal requirements in the next release.

Section 7.6.4 describes synchronization requirements for redundant deterministic execution that were required but not elaborated in 7.6.2. The interface of using a communication API other than `ara::com` is not in the scope of the specification. We focus on the single domain synchronization for the current release, i.e. the redundant deterministic execution is in the same OS or ECU. The models and configuration for deterministic synchronization and the details of interaction with Software Lockstep will be specified in later release.

Section 6.1 details requirements from `Execution Management` Requirement Specification [6] that are not elaborated within this specification. The presence of these requirements in this document ensures that the requirement tracing is complete and also provides an indication of how `Execution Management` will evolve in future releases of the `AUTOSAR Adaptive Platform`.

The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.

4.3 Specification of Diagnostics (UID 723, SWS)

- Only scheduler type 1 from [7] is supported for service 0x2A
- Subfunction 'defineByMemoryAddress' for service 0x2C is not supported
- OBD ISO 15031 and WWH OBD ISO 27145 is not supported by the `DM`.
- *Software Cluster*/*Diagnostic Server instances* are supported by `DM` interfaces but are not specified in detail.
- *DoIP edge node* is not supported by the `DM`.
- The following UDS services are not implemented by the `DM`:
  - 0x23 ReadMemoryByAddress
  - 0x24 ReadScalingDataByIdentifier
  - 0x2F InputOutputControlByIdentifier
  - 0x38 RequestFileTransfer
  - 0x3D WriteMemoryByAddress

  - 0x83 AccessTimingParameter
  - 0x84 SecuredDataTransmission
  - 0x87 LinkControl
- Sub-functions of UDS services are implemented according to ISO 14229-1[8] unless explicitly stated.
- The UDS mirror event memory is not supported by the `DM`. As a result of this, the `DM` does not support the `UDS service`
  - 0x19 with subfunction 0x0F (reportMirrorMemoryDTCByStatusMask)
  - 0x19 with subfunction 0x10 (reportMirrorMemoryDTCExtDataRecordByDTCNumber)
  - 0x19 with subfunction 0x11 (reportNumberOfMirrorMemoryDTCByStatusMask)
- The OBD/WWH OBD is not supported by the `DM`. As a result of this, the `DM` does not support the `UDS service`
  - 0x19 with subfunction 0x05 (reportDTCStoredDataByRecordNumber)
  - 0x19 with subfunction 0x12 (reportNumberOfEmissionsOBDDTCByStatusMask)
  - 0x19 with subfunction 0x13 (reportEmissionsOBDDTCByStatusMask)
  - 0x19 with subfunction 0x42 (reportWWHOBDDTCByMaskRecord)
  - 0x19 with subfunction 0x55 (reportWWHOBDDTCWithPermanentStatus)
- The following general `UDS services` of `ReadDTCInformation` are not supported:
  - 0x19 with subfunction 0x03 (reportDTCSnapshotIdentification)
  - 0x19 with subfunction 0x08 (reportDTCBySeverityMaskRecord)
  - 0x19 with subfunction 0x09 (reportSeverityInformationOfDTC)
  - 0x19 with subfunction 0x0B (reportFirstTestFailedDTC)
  - 0x19 with subfunction 0x0C (reportFirstConfirmedDTC)
  - 0x19 with subfunction 0x0D (reportMostRecentTestFailedDTC)
  - 0x19 with subfunction 0x0E (reportMostRecentConfirmedDTC)
  - 0x19 with subfunction 0x15 (reportDTCWithPermanentStatus)
  - 0x19 with subfunction 0x16 (reportDTCExtDataRecordByRecordNumber)
- Event Memory: Variant handling at runtime for events/DTCs is not supported.

- Event Memory: Details for combined events are not specified.
- Persistent Storage of failed attempts to change security level: After each increment of the attempt counter, it shall be persisted to survive accidental or intended resets. Here the option to select the persistent storage is mandatory in Adaptive Autosar.

4.4 Specification of Platform Health Management for Adaptive Platform (UID 851, SWS)

- `Daisy chaining` (i.e. forwarding Supervision Status, `Checkpoint` or `Health Channel` information to an entity external to PHM or another PHM instance) is currently not supported in this document release.
- `Platform Health Management` configuration related to Supervision Modes is not fully supported in this document release.
- An API to inform Supervised Entities about the Supervision states is available only in polling mode. No API using notification mode is available in this release.
- Interface with the Diagnostic Manager is not specified in this release.

4.5 Specification of Persistency (UID 858, SWS)

Although a `Key-Value Storage` and `File Storage` can be configured as write-only, the current API always allows read access. Read access is even possible when a file has been opened with `ara::per::FileStorage::OpenFileWriteOnly`.

4.6 Specification of Cryptography for Adaptive Platform (UID 883, SWS

The following functional domains and descriptions are still missing in the current version of Crypto API specification:

- **Asynchronous interfaces**
  Currently there is only a synchronous API specification and asynchronous behavior (if required) should be implemented on the consumer application level. It can be done via utilization of dedicated execution threads for long-time operations.
- **Full X.509 certificate support incl. OCSP and OCSP stabling**
  `CryptoAPI` doesn't provide complete specification of the X.509 certificates management on the client (ECU) side yet. Current version of Crypto API specifies only minimal subset of interfaces responsible for basic X.509 functionality and related on utilization of cryptographic algorithms. Current API supports extraction and

parsing of only basic attributes of X.509 certificates and certification requests. An extension of the API specification by additional interfaces dedicated for complete support of X.509 extensions is planned for the next release of this specification.
Note: Generally current specification of the X.509 Provider API is preliminary and subject for extensions and changes.

- **Formats of certificate objects**
  Current version of `CryptoAPI` has minimal support of well-known cryptographic formats encoding/decoding: support of only DER and PEM encoding for X.509 certificates and certificate signing requests is required from any implementation of `CryptoAPI`. For other cryptographic objects an implementation can support only "raw" formats. Following extension of the `CryptoAPI` by unified interfaces for encoding/decoding of complex objects to standard formats is planned for the next release of this specification.

4.7 Specification of Update and Configuration Management (UID 888, SWS)

`UCM` is not responsible to initiate the update process. `UCM` realizes a service interface to achieve this operation. The user of this service interface is responsible to verify that the vehicle is in a updatable state before executing a software update procedure on demand. It is also in the responsibility of the user to communicate with other `AUTOSAR Adaptive Platforms` or `AUTOSAR Classic Platforms` within the vehicle.

The `UCM` receives a locally available software package for processing. The software package is usually downloaded from the OEM backend. The download of the software packages has to be done by another application, i.e. `UCM` does not manage the connection to the OEM backend. Prior to triggering their processing, the software packages have to be transferred to `UCM` by using the provided `ara::com` interface.

The `UCM` update process is designed to cover updates on use case with single `AUTOSAR Adaptive Platform`. `UCM` can update `Adaptive Applications`, the `AUTOSAR Adaptive Platform` itself, including all functional clusters and the underlying OS.

The `UCM` is not responsible for enforcing authentication and access control to the provided interfaces. The document currently does not provide any mechanism for the confidentiality protection as well as measures against denial of service attacks. The assumption is that the platform preserves the integrity of parameters exchanged between `UCM` and its user.

The `UCM` do not support update of `ECU`s not supporting ARA::COM or UDS with aligned diagnostic flash sequence support.

4.8 Specification of Network Management (UID 898, SWS)

The Adaptive Network Management currently only supports UdpNM.

The Adaptive Network Management does not allow node detection (Repeat Message State) but only handles incoming requests.

The Adaptive Network Management cannot be configured as the master network coordinator.

The Adaptive Network Management does not support coordinated shutdown using the information in CBV.

The Adaptive Network Management does not support passive mode and passive start-up. Passive start-up would mean that a node has started (i.e. goes to Normal mode), but the network has been woken up by another node.

Modeling part for mapping the logical networks to the BitVector positions as defined in chapter 7.3 is not available in the manifest.

Update and access of User Data has been removed as the service interface to Applications has been removed. State Management will control the network request/release and it must be clarified if user data changes/indications shall be done via State Management or directly by applications.

4.9 Specification of Identity and Access Management (UID 900, SWS)

- A detailed API will be added in a future release
- Currently limited to ara::com
- For other Functional Clusters, implementation on Policy Enforcement Points are envisaged for a future release.

4.10 Specification of the Adaptive Core (UID 903, SWS)

- The specification of some data types (Array, Map, Optional, String, StringView, Variant) mentions “supporting constructs”, but lacks a precise scope definition of this term.
- The specification of some data types (Map, Vector, String) is lacking a comprehensive definition of memory allocation behavior; it currently only describes it as “implementation-defined”.
- Chapter FunctionalSpecification describes some behavior informally that should rather be given as specification items.

5 Release history

5.1 Release R20-11

Name	Specification history entry
Explanation of Adaptive Platform Design	• Moderate amount of changes in the State Management, Update and Configuration Management, Cryptography, and Safety • Minor changes in Execution Management, Diagnostics, Persistency, Identity and Access Management • Minor changes in the architecture logical view
Methodology for Adaptive Platform	• editorial changes (consolidated usage of terms ECU and Machine) • removed obsolete signal-based Service Interface • removed obsolete appendix "Used classes in Manifest files"
Requirements on Manifest Specification	• Added requirements for Raw Data Stream Deployment
Specification of Manifest	• Remodeling of Phm contribution • Reporting of Security Events • Support for cryptographic Operations • Remodeling of Diagnostic Mapping • minor corrections / clarifications / editorial changes
General Requirements specific to Adaptive Platform	• More design guidelines for special member functions added • Support of C++ 14 added
Requirements on Communication Management	• E2E protection – Events and methods – Traceability to Safety Requirements • Service uniqueness: Offered service with Fully Qualified Service ID • Raw Data Streaming requirements set to valid • Communication Groups (draft)
Specification of Communication Management	• Added SecOC Behaviour, API and Freshness Value Management to specification • Standardized API Error Codes for ara::com API • Added unique ErrorDomain identifiers • Added Named Constructor Approach • Updated E2E Support for methods and events • Updated Raw Data Streaming chapters • Introduced optional execution context parameter to APIs with an asynchronous callback • Changed kCapabilityEnforcmentError to kGrantEnforcementError • Moved magic numbers for "entry type" field to PRS_SOMEIPServiceDiscovery • Editorial Changes

△

Name	Specification history entry
Requirements on Operating System Interface	• Uptrace to RS_Safety[1] document • Clarified Execution Management description • Moved time-triggered execution to RS_ExecutionManagement[2]
Specification of Operating System Interface	• Uptrace update • Clarified Execution Management description • Removed undefined mention of Unrecoverable State
Requirements on Execution Management	• Added: RS_EM_00150
Specification of Execution Management	• Further refinement of State Management API and semantics • Update process lifecycle (terminating report optional) • Added Deterministic Synchronization support • EM-PHM interaction
Specification of Diagnostics	• Document quality improvement and fixing bugs • Incorporated Quality Scope Review Findings • Validated requirements from concept DoIPExtension • Introduced UDS services 2A & 2C
Adaptive Platform Release Overview	• Release Life Cycle Status: R20-11 is in Evolution, R20-11 supersedes R19-11
Guidelines for the use of the C++14 language in critical and safety-related systems	• no changes since document is set to obsolete
Demonstrator Design of Functional Cluster Communication Management	• Add Instance Specifier to Instance Identifier translation. • Add Fire and Forget methods. • Add support for Fields in DDS. • Add Service Versioning. • Remove ara::per usage in ara::com. • Add support for invalid values. • Use new APIs for getting events with SOME/IP.
Demonstrator Design of Functional Cluster Execution Management	• Updated known limitations • Minor editorial changes
Demonstrator Design of Functional Cluster Diagnostics	• No content changes
Explanation of ara::com API	• Replaced term "(Un)Checked Exception" by proper formulations • Clarified the usage and transferation of "Instance Specifier" • Removed the reference to "AUTOSAR_RS_CPP14Guidelines"
Specification of Platform Health Management for Adaptive Platform	• Changed role of PHM to a monitor who notifies State Management, thus rework of logic and interfaces. • Integration of Identity and Access Management for PHM • Moving specification of Health Channel Supervision from Foundation to Adaptive Platform • Reintroduced Enum for Checkpoints and Health Status

▽

△

Name	Specification history entry
Requirements on Platform Health Management for Adaptive Platform	• Added RS_PHM_09255, RS_PHM_09257, RS_PHM_09240, RS_PHM_09241 (moved from FO) • Removed RS_PHM_00110 • Cleanup of requirement trace
Specification of Log and Trace	• Introduced Non-modeled messages and Modeled messages to Chapter 7.3 Log Messages • Introduced Logger::WithLevel() API, to log messages and pass the LogLevel as an API parameter • Refactoring and editorial changes
Requirements on Persistency	• Added support for file meta-data • Added up-traces to RS Safety • Adapted to common document structure • Added requirement history
Specification of Persistency	• Replaced POSIX based file access API and improved error handling and symmetry of other APIs • Full support for encryption and redundancy by hashes using Crypto API • Added information to application about safety related problems • Improved installation/update and redundancy
Demonstrator Design of Functional Cluster Persistency	• Introduced ara::per::ifc::Initialize and ara::per::ifc::Deinitialize • Introduced mechanisms for handling CppImplementationDataTypes • Introduced storage location identification through ara::core::InstanceSpecifier
Demonstrator Design of Functional Cluster Log and Trace	• Removed InitLogging() and implemented usage of ara::core::Initialize • Refactoring and editorial changes
Functional Cluster Shortnames	• Renaming of Core Types to Adaptive Core
Specification of Platform Types for Adaptive Platform	• editorial changes
Specification of RESTful communication	• No functional changes
Specification of Time Synchronization for Adaptive Platform	• TSYNC API redesign and requirments updates • Harmomized with CP and RS Documents • Document adapted to new template • Terminology clarification and cleanup
Requirements on Security Management for Adaptive Platform	• Reworded PEE requirement [RS_SEC_05019] • Added chapter "Conventions" and "Acronyms" • Minor editorial changes to comply with AUTOSAR RS template • Fix spelling in [RS_SEC_05012]
Specification of Cryptography for Adaptive Platform	• Rewrote the document to align with AUTOSAR standard • Update of Crypto API according to WG-SEC feedback

▽

△

Name	Specification history entry
Design guidelines for using parallel processing technologies on Adaptive Platform	• No content changes
Requirements on Update and Configuration Management	• Added traceability to RS SAFETY document
Specification of Update and Configuration Management	• Classic Plaftorm update specification for UCM Master • Refactored UCM Master API • Simplified UCM Master State Machine • Detailed campaign history information
Requirements on Cryptography	• Removed: [RS_CRYPTO_02406] • Updated: [RS_CRYPTO_02201]
System Tests of Adaptive Platform	• Added test cases for CM, REST, EMO, DIAG, PER, IAM, UCM and CRYPTO • Added cross reference links to corresponding Test Configuration in the test cases • Updated limitations for CRYPTO • Removed acronyms which are already part of AUTOSAR glossary
Demonstrator Design of Functional Cluster Update and Configuration Management	• Improved state machine implementation • Partial implementation of vehicle package manager • Migrated shared code of UCM and VPM into library • Reworked document structure
Explanation of Safety Overview	• moved functional and technical safety requirements to RS_Safety • editorial changes • correction of typos and layout • updated abbreviation table • update of identified safety artifacts in AUTOSAR Adaptive Platform
Specification of Network Management	• Several quality improvments • Changed NetworkState DataType from bool to NetworkStateType
Requirements on Identity and Access Management	• Capability renamed to Intent • Changes in architecture regarding process separation reflected in document
Specification of Identity and Access Management	• Introduced policy enforcement in Adaptive Applications • Added IAM for Platform Health Management
Demonstrator Design of Functional Identity and Acess Management	• Rework of the Access Manager and removal of capabilities • Introduction of separate grant for demonstrational purposes • Usage of the execution manager to identify processes

▽

△

Name	Specification history entry
Specification of the Adaptive Core	• Add specifications about "Explicit Operation Abortion" • Add specification about reserved symbol prefixes • Add specification of class SteadyClock • Add section about async signal safety of ARA APIs • Extend error domain scope with vendor-defined error domains • Add specifications about defining own error domains • Various extensions and fixes to the C++ data types • Incorporate contents of SWS_General • Rename document into "Adaptive Core"
Demonstrator Design of Functional Cluster RESTful	• No content changes
Demonstrator Design of Functional Cluster Time Synchronization	• Known Limitations chapter clean-up
Specification of State Management	• Interface towards Update And Configuration Management updated • Interface towards adaptive Diagnostics updated • Introduced Diagnostic Reset based on Communication Groups • Interface towards Platform Health Management updated • Error reactions for supervised entity failures moved to State Management • Introduced PowerModes based on Communication Groups • RequestState and ReleaseRequest interface removed
Requirements of State Management	• No content changes
Requirements on Automated Driving Interfaces	• Initial release
Specification of Sensor Interfaces	• Initial release
Explanation of Sensor Interfaces	• Updated optional elements supporting in section 5.3
Demonstrator Design of Functional Platform Health Management	• Implementation of PHM Daemon • Implementation of APIs for Recovery and Health Channel to SM has been introduced • Health Monitoring (i.e. Chapter 7) from ASWS_HM is moved to SWS_PHM
Guidelines for using Adaptive Platform interfaces	• The name of the chapter "Core Types" to "Adaptive Core" and some minor changes in the chapter • Moderate changes in the State Management chapter • Minor changes in the Persistency Chapter
Explanation of IPsec Implementation Guidelines	• No content changes • Changed Document Status from Final to published
Specification of Timing Extension for Adaptive Platform	• Initial release
Specification of Intrusion Detection System Manager for Adaptive Platform	• Initial release
Explanation of Adaptive Platform Software Architecture	• Initial release

▽

△

Name	Specification history entry
Explanation of Adaptive Platform Software Architectural Decisions	• Initial release

Table 5.1: Release History