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	R21-11

Document Change History

Date	Release	Changed by	Description
2021-11-25	R21-11	AUTOSAR Release Management	Release Life Cycle Status: R21-11 is in Evolution, R21-11 supersedes R20-11

Disclaimer

Table of Contents

References

1 Introduction

1.1 Scope of this document

This document provides an overview of the AUTOSAR standard Adaptive Platform release R21-11.

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 has started a discussion with all the working groups to replace these terms. AUTOSAR is committed to provide all specification documents without these terminology in the coming and future releases. Nevertheless, it may take several releases before the terms are completely replaced, as AUTOSAR has to continue its operations and thousands of pages of existing specifications have to be reviewed and updated in parallel.

1.2.2 Usage of W3C XML schema

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

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

License

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

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

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

Disclaimers

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.

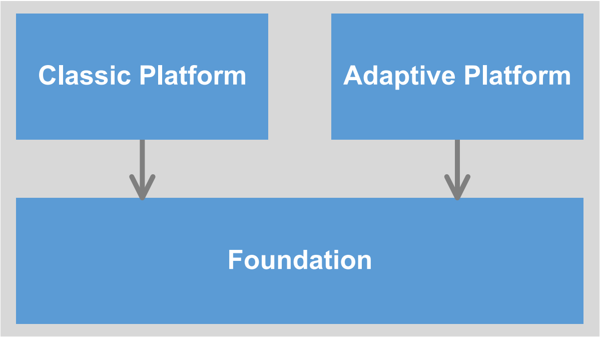


Figure 1.1: Dependencies of AUTOSAR Standards

1.3.5 Dependencies to other Standards

This release of the Adaptive Platform depends on the standard Foundation in release R21-11, 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.

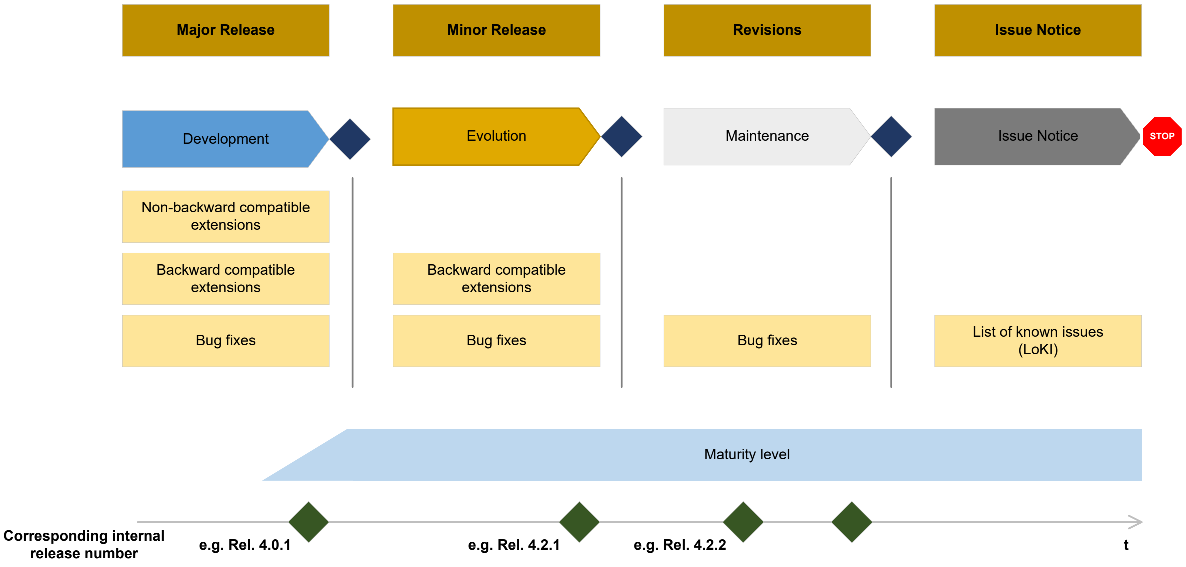


Figure 1.2: Life cycle model of AUTOSAR standards

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
AUTOSAR_00050	R21-11	R4.7.0

According to the release life cycle of AUTOSAR the release R21-11 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
AUTOSAR_00050	R21-11

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. The Demonstrator release 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 production) 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.

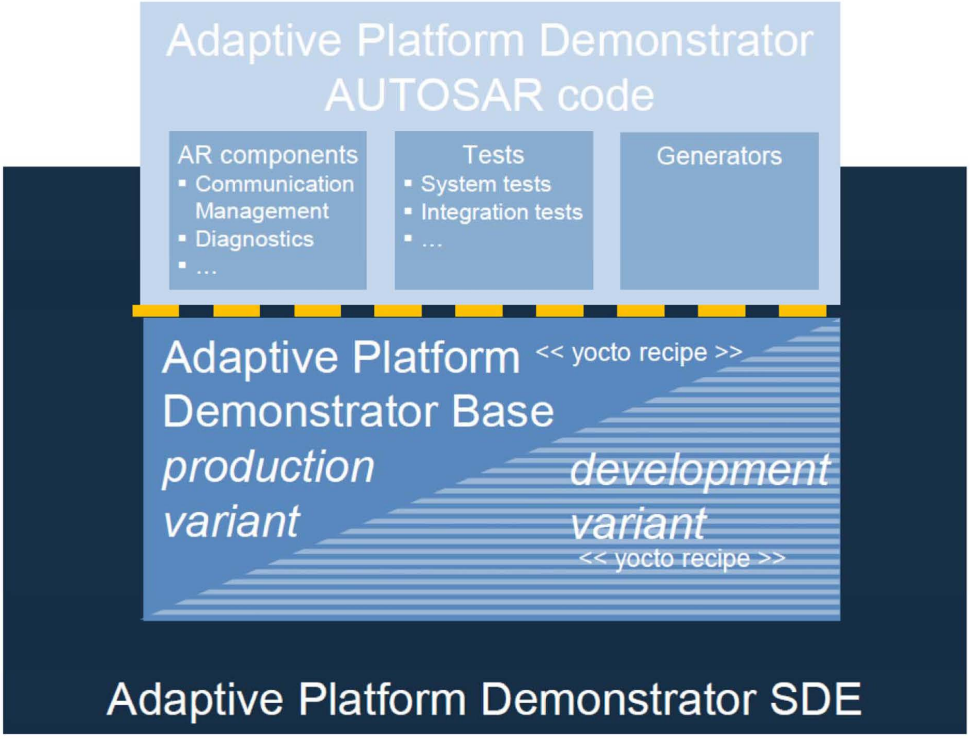


Figure 1.3: Overview of the AUTOSAR Adaptive Platform Demonstrator

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 R21-11. This chapter is structured according to the clusters of AUTOSAR release R21-11.
- 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 R20-11.

2.1 Release R21-11

Several concepts affecting the Adaptive Platform have been introduced with release R21-11 thereby adding functionality to the platform.
Those concepts are e.g. related to the communication protocol DDS (DDS Security and DDS Enhanced Discovery) or safety (E2E for Fields).

Additionally one concept targets the Classic and Adaptive Platform, strengthening the interaction between the two platforms.

2.1.1 Concepts

2.1.1.1 Introduced Concepts

The following concepts in 2.1.1.1.1 - 2.1.1.1.4 have been introduced.

2.1.1.1.1 Mode Dependent Configuration

This concept provides the facility to configure Mode Dependent Supervision (Alive Supervision, Deadline Supervision and Logical Supervision). The modes are derived from Function Group States.
Note: It is also possible to disable Supervision of a Supervised Entity Instance in a particular mode.

Additionally, this concept provides "Failed Reference Cycles Tolerance" per "Alive Supervision". Earlier, "Failed Reference Cycles Tolerance" was available per "Local Supervision".

The concept is applicable for "Adaptive Platform" only. "Classic Platform" already has "Mode Dependent Supervision".

2.1.1.1.2 DDS Security in Communications Management DDS Network Binding

The concept achieves distributed security capabilities in the context of and Identity and Access Management, through specific DDS Security standard mappings in the DDS Network Binding of Communications Management Functional Cluster.

2.1.1.1.3 DDS Enhanced Discovery

This concept improves the DDS Network Binding functional specification and related meta-model packages to make service instance discovery more flexible, efficient and scalable.

2.1.1.1.4 E2E For Fields

This concept extends the E2E protection to fields, the publisher subscriber pattern of AUTOSAR applications. The E2E protection is applied to Getter/Setter functions of fields and the notification of subscriber applications.

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 artifacts for the actually impacted specifications.

Concept Name	Specification Long Name	Standard	Concept Lifecycle
Mode Dependent Configuration	Specification of Health Monitoring	Foundation	draft
	Requirements on Health Monitoring		
	Specification of Platform Health Management	Adaptive Platform	
	Specification of Manifest		
	Requirements on Platform Health Management		
DDS Security in Communications Management DDS Network Binding	**Integration of DDS Security**	Adaptive Platform	draft
	Specification of Manifest		
	Specification of Communication Management		
DDS Enhanced Discovery	Specification of Manifest	Adaptive Platform	draft
	Specification of Communication Management		
E2E For Fields	Explanation of Diagram Source	Foundation	draft

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
	Specification of Module E2E Transformer	Classic Platform	
	Specification of Communication Management	Adaptive Platform	

Table 2.1: Impact of Concepts

2.1.1.3 Validated Concepts

The following concept has been validated:

- 10BASE-T1S (Part 1)

2.1.2 Specifications

2.1.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Integration of DDS Security (UID 1027, TR)

In addition to the above listed new specifications, the following documents have been added with R21-11:

- Specification of Language Binding for modeled AP data types (UID 994, SWS)

2.1.2.2 Migrated Specifications

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

- none

2.1.2.3 Obsolete Specifications

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

- Specification of RESTful Communication (UID 876, SWS)

2.1.2.4 Removed Specifications

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

- none

2.1.2.5 Reworked Specifications

The following specifications have been changed fundamentally in R21-11

- none

2.1.2.6 Moved Specification parts

The following specification parts have been moved to other documents in R21-11.

- none

2.1.2.7 Renamed Specifications

The following specification has been renamed in this release:

- (UID 888) AUTOSAR_SWS_UpdateAndConfigManagement to AUTOSAR_SWS_UpdateAndConfigurationManagement
- (UID 887) AUTOSAR_RS_UpdateAndConfigManagement to AUTOSAR_RS_UpdateAndConfigurationManagement

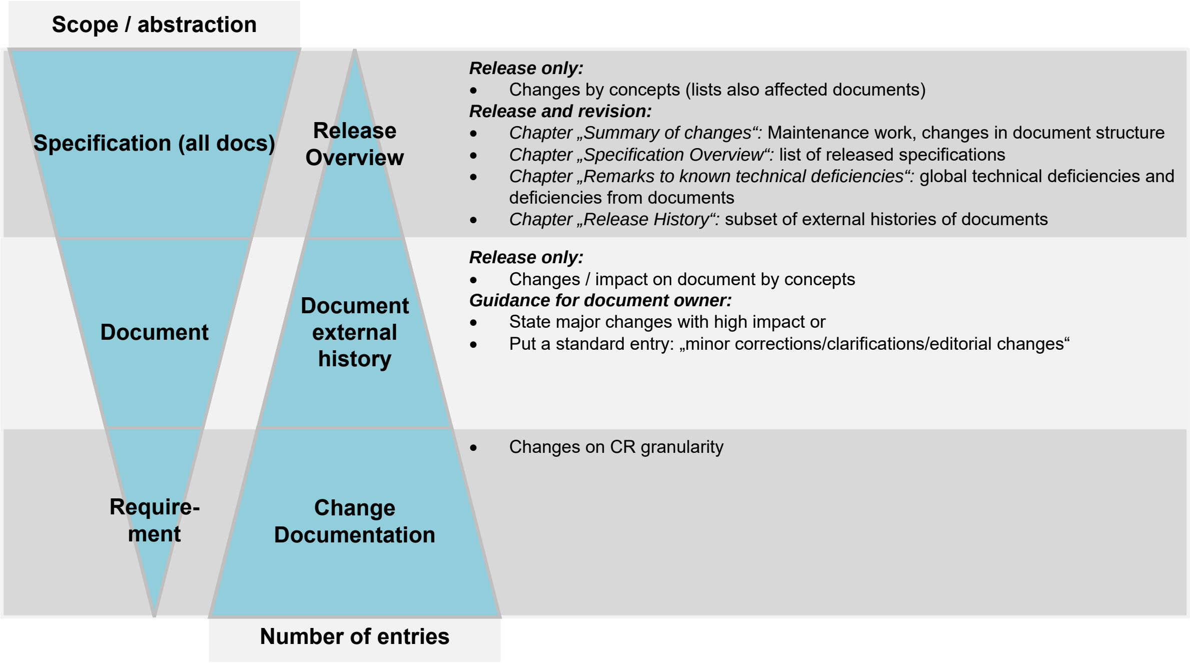
2.1.3 Release Documentation

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 is also available for Adaptive Platform since release R20-11.



Release only:

- Changes by concepts (lists also affected documents)

Release and revision:

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

Release only:

- Changes / impact on document by concepts

Guidance for document owner:

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

- Changes on CR granularity

Figure 2.1: History information in AUTOSAR

3 Specification overview

The published specifications are divided into the clusters

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

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	
Adaptive Foundation		
Explanation of ara::com API	AUTOSAR_EXP_ARAComAPI	
Explanation of IPsec Implementation Guidelines	AUTOSAR_EXP_ IPsecImplementationGuidelines	
Integration of DDS Security	AUTOSAR_TR_ DDSSecurityIntegration	Initial release
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	AUTOSAR_RS_ PlatformHealthManagement	
Requirements on Security Management for Adaptive Platform	AUTOSAR_RS_SecurityManagement	
Specification of Adaptive Platform Core	AUTOSAR_SWS_ AdaptivePlatformCore	
Specification of Communication Management	AUTOSAR_SWS_ CommunicationManagement	
Specification of Cryptography	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	

▽

△

Long Name	File Name	Life cycle changes
Specification of Intrusion Detection System Manager for Adaptive Platform	AUTOSAR_SWS_AdaptiveIntrusionDetectionSystemManager	
Specification of Language Binding for modeled AP data types	AUTOSAR_SWS_LanguageBindingForModeledAPdatatypes	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	AUTOSAR_SWS_PlatformHealthManagement	
Specification of RESTful Communication	AUTOSAR_SWS_RESTfulCommunication	obsolete
Specification of Time Synchronization	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	
Requirements on Update and Configuration Management	AUTOSAR_RS_UpdateAndConfigurationManagement	
Specification of Network Management	AUTOSAR_SWS_NetworkManagement	
Specification of Sensor Interfaces	AUTOSAR_SWS_SensorInterfaces	
Specification of State Management	AUTOSAR_SWS_StateManagement	
Specification of Update and Configuration Management	AUTOSAR_SWS_UpdateAndConfigurationManagement	
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	
Explanation of Adaptive Platform Software Architecture	AUTOSAR_EXP_SWArchitecture	
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	

▽

△

Long Name	File Name	Life cycle changes
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	
Specification of Timing Extension for Adaptive Platform	AUTOSAR_TPS_AdaptivePlatformTimingExtensions	

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
846	Explanation of ara::com API	EXP	4.3
851	Specification of Platform Health Management	SWS	4.4
858	Specification of Persistency	SWS	4.5

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 [4]. Additional, platform specific limitations regarding E2E protection are described in chapter 7.3.1 and 7.2.1.

The following limitations regarding optionality introduced with the Tag-Length-Value serialization principle described in [5] and [6] apply:

- Optional method arguments

[SWS_CM_CONSTR_00001]DRAFT Communication Management does currently not support the existence of optional method arguments.c

In addition the following features are not supported in the current version of this document:

- E2E protection of ServiceInterface.triggers

4.2 Specification of Execution Management (UID 721, SWS)

- The functionality "Resource Limitation" and "Fault Tolerance" is mentioned within this document but is not fully specified in this 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 communication APIs 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 a later release.
- The following topics need to be addressed in the upcoming releases:
  - Further enhancements of chapter 7.6.3.1 Control of Cyclic Executions need to give more details on the state machine
  - Explanations to be given how GetRandom() shall determine if a process is executed redundantly within runnable objects being processed by a worker pool
  - Error handling in Deterministic Client to be described

4.3 Explanation of ara::com API (UID 846, EXP)

The document is currently under major rework and will be replaced with a new version in the next release

4.4 Specification of Platform Health Management (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.
- Interface with the Diagnostic Manager is not specified in this release.
- Health Channels (HealthChannelExternalStatus) are set to obsolete. They are expected to be introduced at State Management in the next release.
- The configuration attribute for the alive notification cycle time (with respect to PHM sending AliveNotification to watchdog interface) is not specified for this release.

- A change in the value of Supervision (Alive/Deadline/Logical) configuration parameters between two Function Group states wherein the process being supervised continues to execute on switching between these states is not considered. The Supervision continues as per configuration in the Supervision Mode corresponding to old Function Group state.
- Similar to above limitation, dynamic change between Supervision exclusion (disable) and Supervision inclusion (enable) on Function Group state change wherein the process under consideration continues to execute on change in Function Group state is not supported. Supervision exclusion or inclusion can be applied starting with the Function Group state in which execution of the process begins and the same is applied until termination of the process.
- Currently specified mechanism of Notifying State Management on Global Supervision Status reaching state kStopped is insufficient in case of multiple failures. It could happen that the Global Supervision Status remains in state kStopped without further notification to State Management about successive failures. Thereby the recovery might be hindered.
- "PowerMode" dependent Supervision configuration is not supported in this release. See [9] for information on "PowerMode".
- Exact point in time at which Alive Supervision is to be started and stopped is not yet specified.

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`.

- 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 R21-11

Name	Specification history entry
Adaptive Platform Release Overview	• Release Life Cycle Status: R21-11 is in Evolution, R21-11 supersedes R20-11
Design guidelines for using parallel processing technologies on Adaptive Platform	• No content changes (only converted to LaTex)
Explanation of Adaptive Platform Design	• Removal of the REST chapter • Introduction of a IDSM (Intrusion Detection System Management) chapter • Introduction of SHM (System Health Management) in the PHM chapter • Refreshed contents in the Persistency chapter • Refreshed UCM and SM contents with regard to their interactions • Minor updates in the Exection Management, Diagnostics, Time Sychronization
Explanation of Adaptive Platform Software Architectural Decisions	• Added architectural decisions made for release R21-11 • Updated the list of affected functional clusters
Explanation of Adaptive Platform Software Architecture	• Applied a more fine-grained description schema for functional clusters and interfaces in the Building Block View. • Removed functional cluster RESTful Communication • Added functional cluster Adaptive Intrusion Detection System Manager • Added section for clarification of diagnostic deployment options
Explanation of ara::com API	• No changes. Fully reworked version of this document is going to be released in R22-11.
Explanation of IPsec Implementation Guidelines	• No content changes
Explanation of Safety Overview	• updates in chapters 4 and 5, tables 4.1, 4.2, 5.2, 5.1 • chapter 4.2: remove malfunction, keep failure only • rename abbreviation AP-HA to TLF (Top Level Failure) • update table 5.2: Feature -> Use-Case • update table 5.2: Malfunction -> Failure • remove hazard from table title • remove references to ara:res
Explanation of Sensor Interfaces	• Add the reference to ISO 23150:2021
Functional Cluster Shortnames	• Functional Cluster shortname rest for RESTful communication removed • New Functional Cluster: Intrusion Detection System Manager • Editorial changes in cluster titles

▽

△

Name	Specification history entry
General Requirements specific to Adaptive Platform	- Guidance on error handling added - More design guidelines added - the sub-namespace ::internal is reserved for vendor-specific use
Guidelines for the use of the C++14 language in critical and safety-related systems	- Added the obsolete statement
Guidelines for using Adaptive Platform interfaces	- A new chapter "Common Regulations" added - Minor updates in the Persistency chapter -
Integration of DDS Security	- Initial release
Methodology for Adaptive Platform	- New description of the Use Cases for the Adaptive Platform, starting with the big picture of the work flow to give an overall view of the methodology flow. - Introduced top-down and bottom-up usage scenarios - spec.items kept their ID and semantics but have been heavily reworked as per updated methodology description, task and artifact names etc.
Requirements of State Management	- Requirements from RS-Safety considered
Requirements on Automated Driving Interfaces	- Add LC-State Draft for all requirements; - Add the reference to ISO 23150:2021
Requirements on Communication Management	- Editorial changes - Removed RESTful communication
Requirements on Cryptography	Updated (upward traceability): - RS_CRYPTO_02001 - RS_CRYPTO_02003 - RS_CRYPTO_02003 - RS_CRYPTO_02004 - RS_CRYPTO_02008 - RS_CRYPTO_02009 - RS_CRYPTO_02106 - RS_CRYPTO_02113 Updated (req. text): - RS_CRYPTO_02209
Requirements on Execution Management	- Added: RS_EM_00015
Requirements on Identity and Access Management	- No content changes
Requirements on Manifest Specification	- Removed requirement for REST
Requirements on Operating System Interface	- Added RS_OSI_00209
Requirements on Persistency	- Re-introduced support of finalization of persistent data - Removed support of removal of persistent data - Harmonized names between SWS and RS - Improved formal aspects of requirements

▽

△

Name	Specification history entry
Requirements on Platform Health Management	• Added RS_PHM_09255, RS_PHM_09257, RS_PHM_09240, RS_PHM_09241 (moved from FO) • Removed RS_PHM_00110 • Cleanup of requirement trace
Requirements on Security Management for Adaptive Platform	• Document Conventions added
Requirements on Update and Configuration Management	• Renamed to RS_UpdateAndConfigurationManagement
Specification of Adaptive Platform Core	• Add spec items for error handling definitions • Add specifications for ScaleLinearAndTexttable, taken over from SWS_CommunicationManagement • Refine scope of ara::core::Initialize • Adapt some APIs to C++14's enhanced capabilities • Align Span with std::span from the C++20 standard • Reduce requirements imposed on handling Violations • Rename document into 'Adaptive Platform Core'
Specification of Communication Management	• Specified use cases and endpoint configuration for RawDataStreams • Added E2E communication protection for Fields • Added E2E profile P44m and P08m • Added new ServiceInterface element Trigger • Extend DDS Serialization of Payload chapter • Extend DDS Network binding chapter • Added Signal-Based Static Network binding • * Added Freshness Value Management (FVM) • * Minor vocabulary improvements and bugfixes
Specification of Cryptography	• Reworked SWS items in chapter 7 to improve testability • Improved traceability between chapters 7 and 8
Specification of Diagnostics	• Document quality improvement and fixing bugs • Incorporated Quality Scope Review Findings • Introduced UDS service 29 Introduced Event Combination in chapter 7
Specification of Execution Management	• Clarified handling of unexpected Process termination • ara::exec::StateClient API updated (constructor token removed) • Invalid state transitions identified and handling defined • ara::exec::-DeterministicClient API and behaviour clarified
Specification of Identity and Access Management	• No content changes
Specification of Intrusion Detection System Manager for Adaptive Platform	• No content changes
Specification of Language Binding for modeled AP data types	• Initial release

▽

△

Name	Specification history entry
Specification of Log and Trace	• Removed useless exceptionalsafety elements • Provided a logmode to logchannel mapping configuration possibility • Header files, c++ syntax errors clean-up • Validation sets, typos and up-traces updates
Specification of Manifest	• Removal of REST support • Rework of signal-to-service conversion • Rework of raw data stream configuration • Rework of diagnostic mapping • minor corrections / clarifications / editorial changes
Specification of Network Management	• Several quality improvements • Removed chapter 10
Specification of Operating System Interface	• Uptrace update
Specification of Persistency	• Clarified and extended specification of Persistency behavior • Improved configuration of storage location and versioning • kNotInitialized was removed • Deleted move constructors/operators
Specification of Platform Health Management	• Health Channels are set to obsolete • Removed retry after failed notification to State Management • Removed GetLocalSupervisionStatus() and GetGlobalSupervisionStatus() APIs from SupervisedEntity class • Added Determination of Supervision Status from Foundation SWS_HealthMonitoring • Added Mode Dependent Configuration Concept • Alignment of Enumeration Literal Indices of SupervisionStatus with Classic Platform WdgM types • Introduction of PhmErrorDomain • Introduction of WatchdogInterface
Specification of Platform Types for Adaptive Platform	• Updated the xml representation of bool and float such that typeEmitter is set to FUNDAMENTAL_TYPE.
Specification of RESTful Communication	• R21-11 APD Release • Set document status to obsolete.
Specification of Sensor Interfaces	• Update the service interface according to ISO 23150:2021; • Add the reference to ISO 23150:2021

▽

△

Name	Specification history entry
Specification of State Management	• Updated method name in Interface towards Update And Configuration Management • Added new error codes in Interface towards Update And Configuration Management • Fixed error handling in Interface towards Update And Configuration Management • Removed timeout supervision for update session • Removed items regarding LastResetCause in Interface towards Diagnostic Management • Added references from chapter 7 to chapter 9
Specification of Time Synchronization	• Document clean-up, Title changed • Chapter "9 Sequence diagrams" temporarily deleted for clean-up purposes • Uptraces updated • Review findings (terminology, typos, etc.) resolved
Specification of Timing Extension for Adaptive Platform	• Corrected specification item numbers and constraint identifier numbers to make the number unique and indicated correct status by setting it to DRAFT • Corrected spelling errors
Specification of Update and Configuration Management	• Renamed to SWS_UpdateAndConfigurationManagement • UCM errors ordering • Vehicle State Manager API detailing
System Tests of Adaptive Platform	• Added test cases for LT, E2E and CRYPTO • Added new sections for Health Management and State Management • Removed REST test cases