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	R23-11

Document Change History

Date	Release	Changed by	Description
2023-11-23	R23-11	AUTOSAR Release Management	• Release Life Cycle Status: R23-11 is in Evolution, R23-11 supersedes R22-11

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Requirements on Manifest Specification
AUTOSAR_AP_RS_ManifestSpecification

1 Introduction

1.1 Scope of This Document

This document provides an overview of the AUTOSAR standard Adaptive Platform release R23-11.

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`.

Disclaimers

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

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

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

1.3 AUTOSAR Standards

1.3.1 Introduction

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

Packaging its deliverables into different "standards"

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

1.3.2 Definition

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

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

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

1.3.3 Overview of AUTOSAR's Standards

AUTOSAR delivers the following standards:

Standard	Abbreviation
Adaptive Platform	AP
Classic Platform	CP
Foundation	FO

1.3.3.1 Adaptive Platform

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

1.3.3.2 Classic Platform

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

1.3.3.3 Foundation

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

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

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

1.3.4 Naming Scheme for Files and Specification Items

AUTOSAR has extended the naming scheme for files and specification items. The objective is to consistently include the AUTOSAR Standard to which the file or specification item belongs in the name. This addition also provides namespaces for the three AUTOSAR Standards and avoids conflicting names for specifications on the same topic in different AUTOSAR Standards. According to the new naming scheme, the abbreviation of the AUTOSAR Standard (AP, CP or FO) is added as first part of specification item IDs and as second part of file names. For details, please refer to [1].

From R23-11 onwards, the names of all files that are part of the release follow the new naming scheme. The IDs of existing specification items are not changed to avoid issues and migrations for AUTOSAR Partners that use these IDs internally.

1.3.5 Dependencies Between Standards

Each release of Classic and Adaptive Platform relies on a dedicated version of Foundation. The specific dependency is documented in chapter 1.4.5.

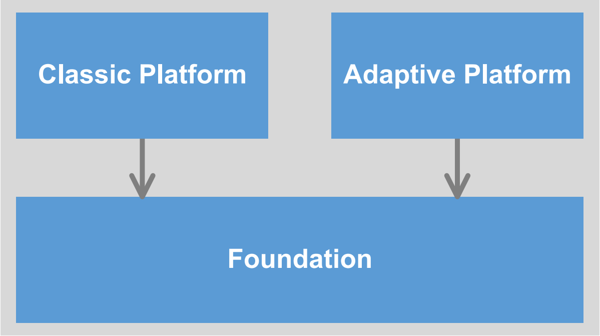

Figure 1.1: Dependencies of AUTOSAR standards

1.3.6 Dependencies to Other Standards

This release of the Adaptive Platform depends on the standard Foundation in release R23-11, 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 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.

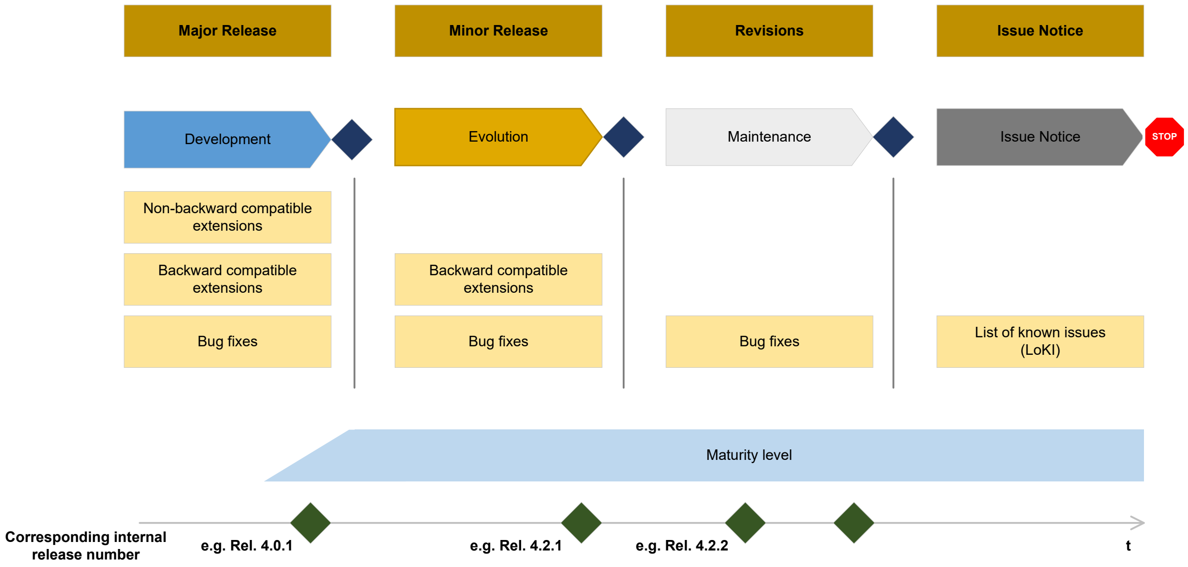

Figure 1.2: Life cycle model of AUTOSAR standards

1.4.2 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.

1.4.3 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.

1.4.4 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 numbers 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.

1.4.5 Overview of AUTOSAR Releases and Corresponding AUTOSAR Schema Versions

Until the Releases CP R4.4.0 and AP R19-03, AUTOSAR released the platforms separately where a Foundation release went along with each platform release. Since compatibility between the platforms is essential to be able to have AP and CP ECUs within one vehicle project, an XML schema needs to be available that works with the different releases. The following table gives an overview about the different schema versions and the corresponding platform releases they can be used for.

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

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

▽

△

Schema Version	Classic Platform release	Adaptive Platform release	Foundation release
AUTOSAR_00046	R4.4.0	R18-10	R1.5.0
AUTOSAR_00047	R4.4.0	R19-03	R1.5.1

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

Schema Version	Platform release	Internal release number
AUTOSAR_00048	R19-11	R4.5.0
AUTOSAR_00049	R20-11	R4.6.0
AUTOSAR_00050	R21-11	R4.7.0
AUTOSAR_00051	R22-11	R4.8.0
AUTOSAR_00052	**R23-11**	**R4.9.0**

According to the release life cycle of AUTOSAR the release R23-11 is a minor release.

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 1.4.1). 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 1.4.2.

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.

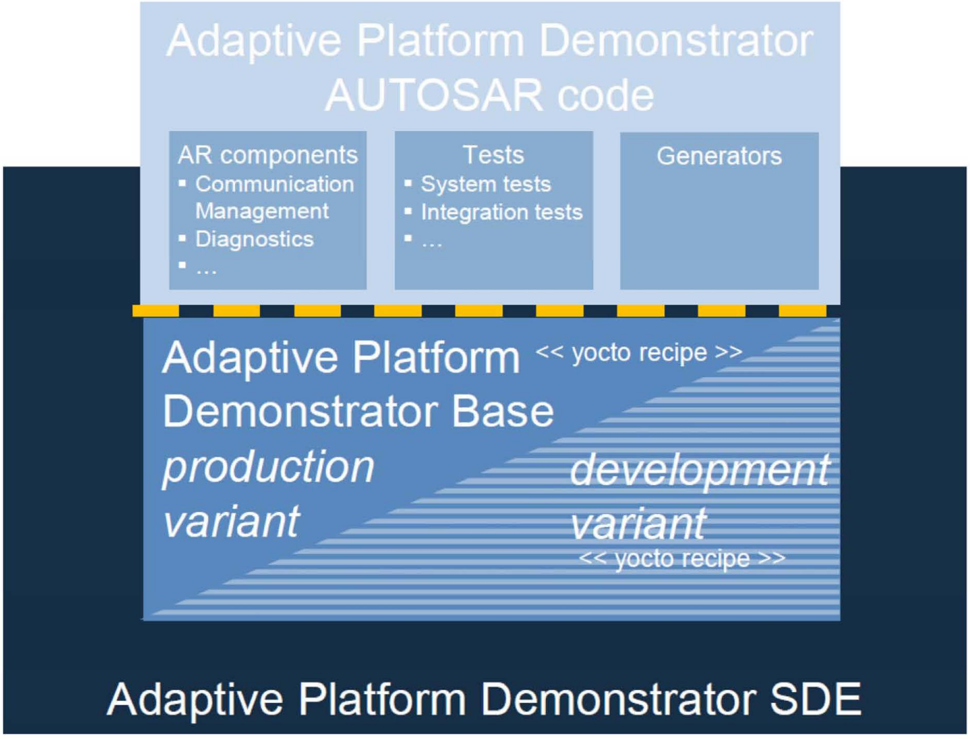

Figure 1.3: Overview of the AUTOSAR Adaptive Platform Demonstrator

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 R23-11. This chapter is structured according to the clusters of AUTOSAR release R23-11.
- Chapter 4 contains remarks about known technical deficiencies.
- Chapter 5 contains the detailed release history of all released specifications.

2 Summary of Changes in Release R23-11

This chapter contains a summary of changes which have been implemented since the previous release R22-11.

Several concepts affecting the Adaptive Platform have been introduced with release R23-11 thereby adding functionality to the platform.
Additionally one concept targets the Classic and Adaptive Platform, strengthening the interaction between the two platforms.

2.1 Concepts

2.1.1 Introduced Concepts

The following concepts in 2.1.1.1 - 2.1.1.4 have been introduced.

2.1.1.1 MACsec

The concept part 2 includes a new explanatory document to clarify the interaction between the AUTOSAR Adaptive Platform with MKA and MACsec capable operative systems. It aims to help implementers and integrators to configure the MKA and MACsec instances on Adaptive Platform devices.

2.1.1.2 Service Oriented Vehicle Diagnostics

The concept extends the Diagnostic Manager to support an SOVD interface according to the ASAM SOVD standard beside the already existing UDS interface.

This allows the implementation of diagnostics based on the HTTPS protocol resp. the RESTful approach.

2.1.1.3 Safe API for Hardware Accelerators

The generic API for Safe Hardware Accelerators utilization has been introduced in the AUTOSAR Adaptive Platform. It allows to utilize the hardware for heavy algorithms in the most effective way.
An Explanatory document for AUTOSAR AP was provided with the following items:

- The API guidelines
- Detailed API description

- API Architecture
- Safety approach
- Explanation and Examples for AP application developers

2.1.1.4 Tracing for Adaptive Platform

The concept introduces a tracing approach to capture timing information of software. It provides a mechanism to specify trace points in functional clusters or applications. The trace data can be collected with tracing tools with minimal runtime overhead. The trace points are configurable at compile time and tracing can be generally enabled or disabled at runtime using the manifest. It also provides a mechanism for low level tracing of operating system events at Kernel-Level with minimal overhead.

2.1.2 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
Tracing for Adaptive Platform	Specification of Manifest	Adaptive Platform	draft
	Technical Report on Operating System Tracing Interface		
	Requirements on Operating System Interface		
	Specification of Operating System Interface		
	Requirements on Execution Management		
	Specification of Execution Management		
	Specification of Log and Trace		
	Requirements on Debugging, Tracing and Profiling support of AUTOSAR Components	Foundation	

▽

△

Concept Name	Specification Long Name	Standard	Concept Lifecycle
	Requirements on Log and Trace		
	Standardized M1 Models used for the Definition of AUTOSAR		
Service Oriented Vehicle Diagnostics	Explanation of Service-Oriented Vehicle Diagnostics	Foundation	draft
	Requirements on Diagnostics		
	Specification of Manifest	Adaptive Platform	
	Specification of Diagnostics		
Safe API for hardware accelerators	List of Adaptive Platform Functional Clusters	Adaptive Platform	draft
	Explanation of Safe API for hardware accelerators		
MACsec	**Explanation of MACsec and MKA Protocols implementation and configuration guidelines**	Adaptive Platform	draft

Table 2.1: Impact of Concepts

2.1.3 Validated Concepts

The following concepts have been validated:

- Firewall for Adaptive Platform

2.2 Specifications

2.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Explanation of Safe API for hardware accelerators (UID 1086, EXP)
- Technical Report on Operating System Tracing Interface (UID 1083, TR)
- Explanation of MACsec and MKA Protocols implementation and configuration guidelines (UID 1067)

In addition to the above listed new specifications, the following documents have been added with R23-11:

- Specification of Raw Data Stream (UID 1098, SWS)
- Specification of Vehicle Update and Configuration Management
  (UID 1090, SWS):

  - The Specification of Update and Configuration Management (UID 888) has been split. Specification items have been taken over and the prefix has been changed from SWS_UCM_ to SWS_VUCM_

- Requirements of Vehicle Update and Configuration Management (UID 1097, RS)
- Explanation of Identity and Access Management (UID 1071, EXP)
- Explanation of ARA Applications in Rust (UID 1079, EXP)

2.2.2 Migrated Specifications

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

- none

2.2.3 Obsolete Specifications

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

- Specification of Timing Extensions for Adaptive Platform (UID 968) is de-facto frozen and shall be marked as obsolete in R24-11. The content shall be merged together with the content from document Specification of Timing Extensions for Classic Platform (UID 411) to provide a unified CP/AP Timing Extensions specification in R24-11.

2.2.4 Removed Specifications

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

- Specification of Identity and Access Management" (UID 900, SWS)
- Guidelines for the use of the C++14 language in critical and safety-related systems (UID 839, RS)

2.2.5 Reworked Specifications

The following specifications have been changed fundamentally in R23-11

- none

2.2.6 Moved Specification Parts

The following specification parts have been moved to other documents in R23-11.

- none

2.2.7 Renamed Specifications

The following specification has been renamed in this release:

- List of Adaptive Platform Functional Clusters (TUID 862, TR)

2.2.8 Life Cycle State of Metamodel

The life cycle of several Adaptive Platform elements in the Metamodel were kept in "draft" until release R21-11. Due to increasing market demands to keep AUTOSAR's XML schema stable, these parts of the model have been set to "valid" in release R22-11. This has an impact on the TPS_Manifest specification [2].

Please be aware that according to [1], only the life cycle states "valid" and "candidate" are used within the Metamodel.

2.3 Release Documentation

There are no major changes in the Release Documentation.

3 Specification Overview

The published specifications are divided into the clusters:

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

Long Name	File Name	Life cycle changes
Release Documentation		
Adaptive Platform Release Overview	AUTOSAR_AP_TR_ReleaseOverview	
AUTOSAR Adaptive Platform Specification Hashes	AUTOSAR_AP_TR_Specification Hashes	
Adaptive Foundation		
Explanation of ara::com API	AUTOSAR_AP_EXP_ARAComAPI	
Explanation of Identity and Access Management	AUTOSAR_AP_EXP_ IdentityAndAccessManagement	Initial release
Explanation of IPsec Implementation Guidelines	AUTOSAR_AP_EXP_ IPsecImplementationGuidelines	
Explanation of MACsec and MKA Protocols implementation and configuration guidelines	AUTOSAR_AP_EXP_MACsec	Initial release
Explanation of Service-Oriented Vehicle Diagnostics	AUTOSAR_AP_EXP_SOVD	
Integration of DDS Security	AUTOSAR_AP_TR_ DDSSecurityIntegration	
Requirements on Communication Management	AUTOSAR_AP_RS_ CommunicationManagement	
Requirements on Cryptography	AUTOSAR_AP_RS_Cryptography	
Requirements on Execution Management	AUTOSAR_AP_RS_ ExecutionManagement	
Requirements on Identity and Access Management	AUTOSAR_AP_RS_ IdentityAndAccessManagement	
Requirements on Operating System Interface	AUTOSAR_AP_RS_ OperatingSystemInterface	
Requirements on Persistency	AUTOSAR_AP_RS_Persistency	
Requirements on Platform Health Management	AUTOSAR_AP_RS_ PlatformHealthManagement	
Specification of Adaptive Platform Core	AUTOSAR_AP_SWS_Core	
Specification of Communication Management	AUTOSAR_AP_SWS_ CommunicationManagement	
Specification of Cryptography	AUTOSAR_AP_SWS_Cryptography	
Specification of Diagnostics	AUTOSAR_AP_SWS_Diagnostics	
Specification of Execution Management	AUTOSAR_AP_SWS_ ExecutionManagement	

▽

△

Long Name	File Name	Life cycle changes
Specification of Firewall for Adaptive Platform	AUTOSAR_AP_SWS_Firewall	
Specification of Intrusion Detection System Manager for Adaptive Platform	AUTOSAR_AP_SWS_IntrusionDetectionSystemManager	
Specification of Language Binding for modeled AP data types	AUTOSAR_AP_SWS_LanguageBindingForModeledAPdatatypes	
Specification of Log and Trace	AUTOSAR_AP_SWS_LogAndTrace	
Specification of Network Management	AUTOSAR_AP_SWS_NetworkManagement	
Specification of Operating System Interface	AUTOSAR_AP_SWS_OperatingSystemInterface	
Specification of Persistency	AUTOSAR_AP_SWS_Persistency	
Specification of Platform Health Management	AUTOSAR_AP_SWS_PlatformHealthManagement	
Specification of Raw Data Stream	AUTOSAR_AP_SWS_RawDataStream	Initial release
Specification of Time Synchronization	AUTOSAR_AP_SWS_TimeSynchronization	
Adaptive Services		
Explanation of Safe API for hardware accelerators	AUTOSAR_AP_EXP_SafeAPIHardwareAccelerators	Initial release
Explanation of Sensor Interfaces	AUTOSAR_AP_EXP_SensorInterfaces	
Requirements of State Management	AUTOSAR_AP_RS_StateManagement	
Requirements on Automated Driving Interfaces	AUTOSAR_AP_RS_AutomatedDrivingInterfaces	
Requirements on Update and Configuration Management	AUTOSAR_AP_RS_UpdateAndConfigurationManagement	
Requirements on Vehicle Update and Configuration Management	AUTOSAR_AP_RS_VehicleUpdateAndConfigurationManagement	Initial release
Specification of Sensor Interfaces	AUTOSAR_AP_SWS_SensorInterfaces	
Specification of State Management	AUTOSAR_AP_SWS_StateManagement	
Specification of Update and Configuration Management	AUTOSAR_AP_SWS_UpdateAndConfigurationManagement	
Specification of Vehicle Update and Configuration Management	AUTOSAR_AP_SWS_VehicleUpdateAndConfigurationManagement	Initial release
Technical Report on Operating System Tracing Interface	AUTOSAR_AP_TR_OperatingSystemTracingInterface	Initial release
General		
Design guidelines for using parallel processing technologies on Adaptive Platform	AUTOSAR_AP_EXP_ParallelProcessingGuidelines	
Explanation of Adaptive Platform Design	AUTOSAR_AP_EXP_PlatformDesign	
Explanation of Adaptive Platform Software Architecture	AUTOSAR_AP_EXP_SWArchitecture	
Explanation of ARA Applications in Rust	AUTOSAR_AP_EXP_ARARustApplications	Initial release
General Requirements specific to Adaptive Platform	AUTOSAR_AP_RS_General	
Guidelines for using Adaptive Platform interfaces	AUTOSAR_AP_EXP_InterfacesGuidelines	

▽

△

Long Name	File Name	Life cycle changes
List of Adaptive Platform Functional Clusters	AUTOSAR_AP_TR_FunctionalClusterList	
System Tests for Adaptive Platform Demonstrator	AUTOSAR_AP_TR_SystemTests	
Methodology and Manifests		
Collection of blueprints for AUTOSAR Adaptive Platform M1 models	AUTOSAR_AP_MOD_GeneralBlueprints	
Methodology for Adaptive Platform	AUTOSAR_AP_TR_Methodology	
Requirements on Manifest Specification	AUTOSAR_AP_RS_ManifestSpecification	
Specification of Manifest	AUTOSAR_AP_TPS_ManifestSpecification	
Specification of Platform Types for Adaptive Platform	AUTOSAR_AP_SWS_PlatformTypes	
Specification of Timing Extensions for Adaptive Platform	AUTOSAR_AP_TPS_TimingExtensions	

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
912	Specification of Sensor Interfaces	SWS	4.1
858	Specification of Persistency	SWS	4.2
851	Specification of Platform Health Management	SWS	4.3

4.1 Specification of Sensor Interfaces (UID 912 SWS)

- 4.1.1 ISO Optionals
  All the ISO optionals are supported in the service definition, but the presence of an optional signal is decided during the design time. The optional signal presence is indicated by a capability vector. There is no change of optional presence during the run time.
- 4.1.2 Sensor Control Interfaces
  Sensor control interfaces are not supported, e.g. actions like reset, initialization, and calibration.
- 4.1.3 Sensor Capabilities

- The configuration of sensor capabilities is not supported, e.g. sensor opening angle and detection range.

4.2 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.

4.3 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.
- Interface with the Diagnostic Manager is not specified in this release.
- Health Channels (HealthChannelExternalStatus) is set to obsolete.

Note: It is not intended to remove this feature from AUTOSAR Adaptive Platform overall. Rather, it is an architectural question to which Functional Cluster this feature belongs to, that is expected to be resolved for 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".

- Supervision is not supported for non-reporting processes (for information regarding what is a non-reporting process, please refer [10]). Rationale: Supervision depends on process states. Non-reporting process is not expected to report its Execution State to Execution Management. Hence, Platform Health Management cannot be informed about the necessary process states by Execution Management.
- Handling of multiple hardware watchdog instances is up to implementation and not standardized in the specification.
- State machine of Elementary Supervision Status is not specified for inter process supervisions (inter process Deadline Supervision and Logical Supervision) in this release.

5 Release History

5.1 Release R23-11

The following deliverables had major changes.

Name	Specification history entry
Adaptive Platform Release Overview	• Release Life Cycle Status: R23-11 is in Evolution, R23-11 supersedes R22-11
Design guidelines for using parallel processing technologies on Adaptive Platform	• Removal of Deterministic Execution
Explanation of Adaptive Platform Design	• An update of the logical architectural view and introduction of protocol, safety, • and security features illustrations in the logical view. • An overall update according to the latest specification change in the State • Management chapter • Removal of obsolete 'Recovery Action' and 'Deterministic Client' description in • the Execution Management chapter • An update of the architecture diagram in the Log and Trace chapter • An update of the architecture diagram in the Log and Trace chapter • Minor updates in the Architecture, Platform Health Management, Update and Configuration Management, Persistency, Identity and Access Management, Operating System Interface and Core Types chapters • Introduction of the Raw Data Stream chapter, and its removal from the Communication Management chapter.
Explanation of Adaptive Platform Software Architecture	• Extended the Use Case View and Runtime View with use cases and scenarios for the Functional Clusters Time Synchronization and Platform Health Management. • Removed Functional Cluster Identity and Access Management. • Added Functional Clusters Vehicle Update and Configuration Management and Raw Data Stream.
Explanation of ARA Applications in Rust	• Initial release
Explanation of ara::com API	• CommunicationGroups is now OBSOLETE • Removed Raw Data Stream chapters (moved to AP EXP InterfaceGuidelines)
Explanation of Identity and Access Management	• Initial release
Explanation of IPsec Implementation Guidelines	• No content changes
Explanation of MACsec and MKA Protocols implementation and configuration guidelines	• Initial release
Explanation of Safe API for hardware accelerators	• Initial release
Explanation of Sensor Interfaces	• No content changes
Explanation of Service-Oriented Vehicle Diagnostics	• Update for R23-11 with alignment to supported SOVD use case
General Requirements specific to Adaptive Platform	• Clarifications

▽

△

Name	Specification history entry
Guidelines for using Adaptive Platform interfaces	• Introduction of the Common Regulations, Diagnostics Management, and the Raw Data Streams chapter • Overall improvement of the Execution Management, Update and Configuration Management, and Persistency chapters • Overall update of the State Management chapter • Removal of the Deterministic Execution Client in the Execution Management chapter • Minor updates of Adaptive Core and Platform Health Management chapters
Integration of DDS Security	• ID format updated, model references fixed ([TR_DDSS_00204], [TR_DDSS_00005], [TR_DDSS_00104], [TR_DDSS_00202], [TR_DDSS_00205])
List of Adaptive Platform Functional Clusters	• Functional Cluster iam for Identity and Access Management removed • Functional Cluster ucm split into ucm and vucm • Functional Cluster shwa reserved for future releases
Methodology for Adaptive Platform	• clean up in chapters 3.9, 3.10, 3.11
Requirements of State Management	• No content changes
Requirements on Automated Driving Interfaces	• No content changes
Requirements on Communication Management	• Communication Groups marked as OBSOLETE
Requirements on Cryptography	• No content changes
Requirements on Execution Management	• Requirements for deterministic execution are set to obsolete • The right to create child processes can be configured by integrator • Added support for standardized trace points
Requirements on Identity and Access Management	• No content changes
Requirements on Manifest Specification	• Removed requirement for • Deterministic Client
Requirements on Operating System Interface	• Changed [RS_OSI_00201], [RS_OSI_00202], [RS_OSI_00203], [RS_OSI_00206] to point to appropriate RS_Safety[1] requirements. • Added [RS_OSI_00210], [RS_OSI_00211] about tracing support.
Requirements on Persistency	• Changed document name to include "AP" • Rewrote the document according to new requirements style • Added handling of authentication
Requirements on Platform Health Management	• Added requirements RS_PHM_00118 and RS_PHM_00119 • Removed RS_PHM_00103 • RS_PHM_00105: Supervised Entities replaced by Supervisions
Requirements on Update and Configuration Management	• Split document with RS Vehicle Update And Configuration Management
Requirements on Vehicle Update and Configuration Management	• Migration of document from standard RS Update And Configuration Management

▽

△

Name	Specification history entry
Specification of Adaptive Platform Core	• Add specification of ara::core::MemoryResource • Remove specification of ara::core::ScaleLinearAndTexttable • Refine specification about platform initialization • Refine specification of Future, and Promise with regards to error handling • Extend Array specification with accessor functions performing checked access • Make undefined behavior explicit by mandating Violations across various C++ data types • Rework of chapter 5 with dependencies to other modules
Specification of Communication Management	• Communication Groups are now OBSOLETE • Removed Raw Data Stream functionality from ara::com • Added new API for checking Subscription state on skeleton side • Harmonization of ara::com API Error Codes • Added clarifications for SOME/IP-SD protocol usage • Reworked usage of Access • Management grants in the ara::com API • Specified lifetime requirements for event sample data • Structural changes for better overview • Editorial changes and bugfixes
Specification of Cryptography	• Improved Crypto API Functional specification in Chapter 7 • Improved Chapter 8. API specification • Added initial version from Chapter 5. Dependencies to other functional clusters
Specification of Diagnostics	• Document quality improvement and fixing bugs • Incorporated Quality Scope Review Findings • SOVD Concept Part 2 implemented • Service 0x29 refinements
Specification of Execution Management	• Requirements for deterministic execution are set to obsolete • The right to create child processes can be configured by integrator • Added support for standardized trace points • API Refinement (ExecutionClient termination handler, remove FunctionGroup, C++ Core Guidelines compliance) • Clarification of Unrecoverable State
Specification of Firewall for Adaptive Platform	• Minor bugfixes
Specification of Intrusion Detection System Manager for Adaptive Platform	• Introduce ContextDataProvider. • Introduce TimestampProvider. • Clarifications regarding event ordering, interaction with DM, and the relationship • between PortPrototype and SecurityEventType.

▽

△

Name	Specification history entry
Specification of Language Binding for modeled AP data types	• API Table generation completed • Editorial changes • Rewording of "Orthogonal" to "Outside" for better clarity
Specification of Log and Trace	• AP Tracing with modeled Messages incorporated • Dependencies to other Functional Clusters • Interface for sending log messages added
Specification of Manifest	• Improve diagnostic configuration • Improve firewall configuration • minor corrections / clarifications / editorial changes
Specification of Network Management	• Replaced Network Management Service Interface with C++ API • Several quality improvements
Specification of Operating System Interface	• Added ARTI tracing interface and related Log messages.
Specification of Persistency	• Improved the clean up and the version handling during update • Removed local handling of minimumSustainedSize • Removed constraints regarding storage sync • Formal description of dependencies to other Functional Clusters
Specification of Platform Health Management	• Addition of thread safety information to PHM APIs • Renaming of PHM security event • Added "k" prefix to enum TypeOfSupervision • Addition of explanations and examples
Specification of Platform Types for Adaptive Platform	• No content changes
Specification of Raw Data Stream	• Initial release
Specification of Sensor Interfaces	• Adaption of ASAM and ADI enumerations; • Editorials clean up;
Specification of State Management	• Add Update and Configuration Management support to StateMachine approach • Add Network Management support to StateMachine approach • Add Controller/Agent StateMachine approach • Add UpdateAllowed service interface • Extend StartStartMachine feature of StateMachine approach • Replace Network Management service Interface by C++ API
Specification of Time Synchronization	• Start of transmission of sync message updated • List of Adaptive Platform Functional Clusters added • Application errors added
Specification of Timing Extensions for Adaptive Platform	• Â Editorial changes • See Disclaimer note in "AP TR AdaptivePlatformReleaseOverview"
Specification of Update and Configuration Management	• Split UCM Master into SWS Vehicle Update Configuration Management

▽

△

Name	Specification history entry
Specification of Vehicle Update and Configuration Management	• Initial release from split of SWS UCM document • Vehicle dependency handling • VSM interface refactoring
System Tests for Adaptive Platform Demonstrator	• Removal of DeterministicClient API
Technical Report on Operating System Tracing Interface	• Initial release

Table 5.1: Overview of specification release histories