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	R24-11

Document Change History

Date	Release	Changed by	Description
2024-11-27	R24-11	AUTOSAR Release Management	• Release Life Cycle Status: R24-11 is in Evolution, R24-11 supersedes R23-11

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] Requirements on Manifest Specification
AUTOSAR_AP_RS_ManifestSpecification

[2] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

1 Introduction

1.1 Scope of This Document

This document provides an overview of the AUTOSAR standard Adaptive Platform release R24-11.

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 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.

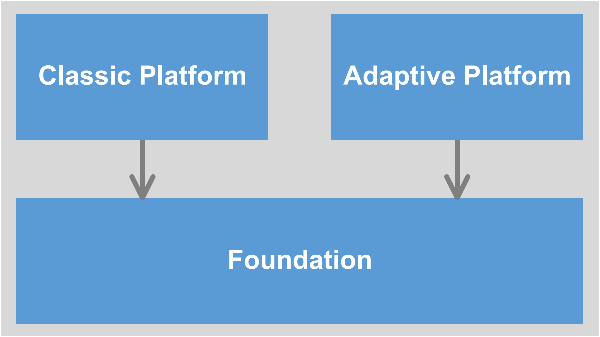

Figure 1.1: Dependencies of AUTOSAR standards

1.3.5 Dependencies to Other Standards

This release of the Adaptive Platform depends on the standard Foundation in release R24-11, 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.

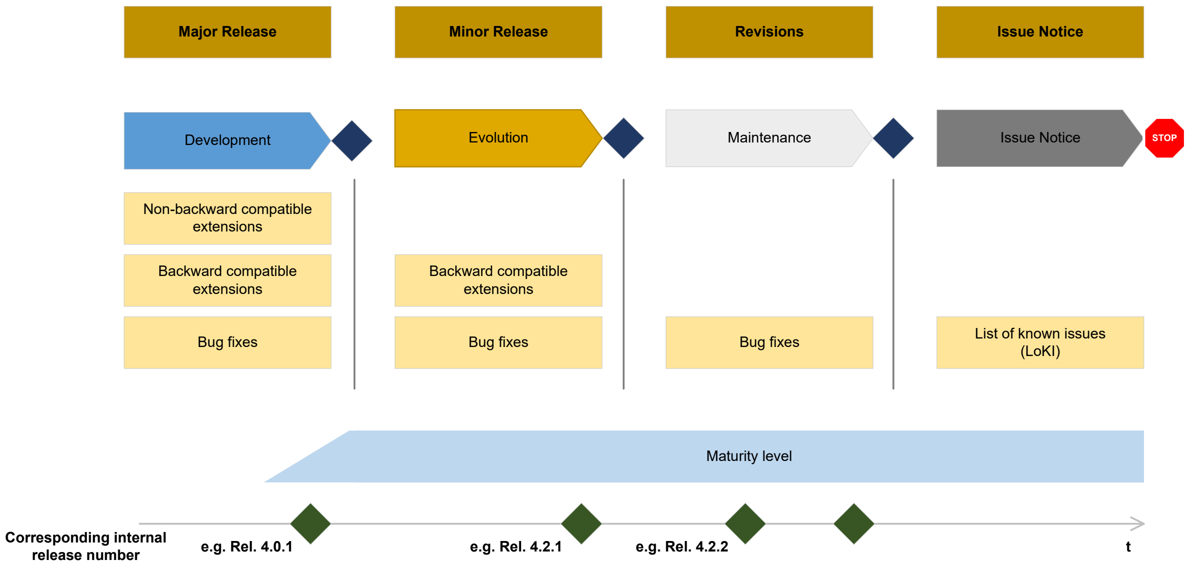

Figure 1.2: Life cycle model of AUTOSAR standards

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 "Status". 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
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
AUTOSAR_00053	**R24-11**	**R4.10.0**

According to the release life cycle of AUTOSAR the release R24-11 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 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.

1.5.2 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.

1.5.3 Adaptive Platform Stabilization

AUTOSAR sees the market need to have interfaces to the applications that are stable throughout the different releases of AUTOSAR. Therefore, AUTOSAR decided to declare the R24-11 a stabilization release. The main goals are

- Achieve a stable and consistent set of APIs for production-grade applications
- Provide a more predictable software environment
- Increase interoperability by improving the compatibility across releases and vendors
- Increase predictability for AUTOSAR partners

To assure these goals, a life-cycle support with efficient management of long-term support and maintenance will be introduced after R24-11.
In R24-11 AUTOSAR set the priority for the Adaptive Platform on bug fixes with necessary incompatible changes to avoid those in future. To name a few:

- Specificatiion of thread safety is mandatory for all APIs including callbacks
- Most APIs are (conditional) noexcept to assure exception safety
- Introduction of rollback semantics for each error code

In addition, several single bugs throughout all Functional Clusters have been solved.

Some issues remain for further stabilization in R25-11 and might lead to incompatible changes. To name a few:

- The stabilization of the Specification of Cryptography (883, SWS)
- Some leftovers of architectural decisions in single clusters (e.g. about thread safety)
- Violations and log messages stay draft in all specifications
- For some APIs and for all move, copy, and converting constructors thread safety is currently implementation defined and will be speficied in future
- Recoverability class is not specified for some APIs
- The minimum C++ version is going to be set to C++17

The affected specification items will keep the life-cycle status "draft". In addition you can find details in chapter 4.
Please be aware, that newly introduced concepts (see chapter 2.1.1), which have not been validated, are introduced with the life cycle state "draft" as well.

1.5.4 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.

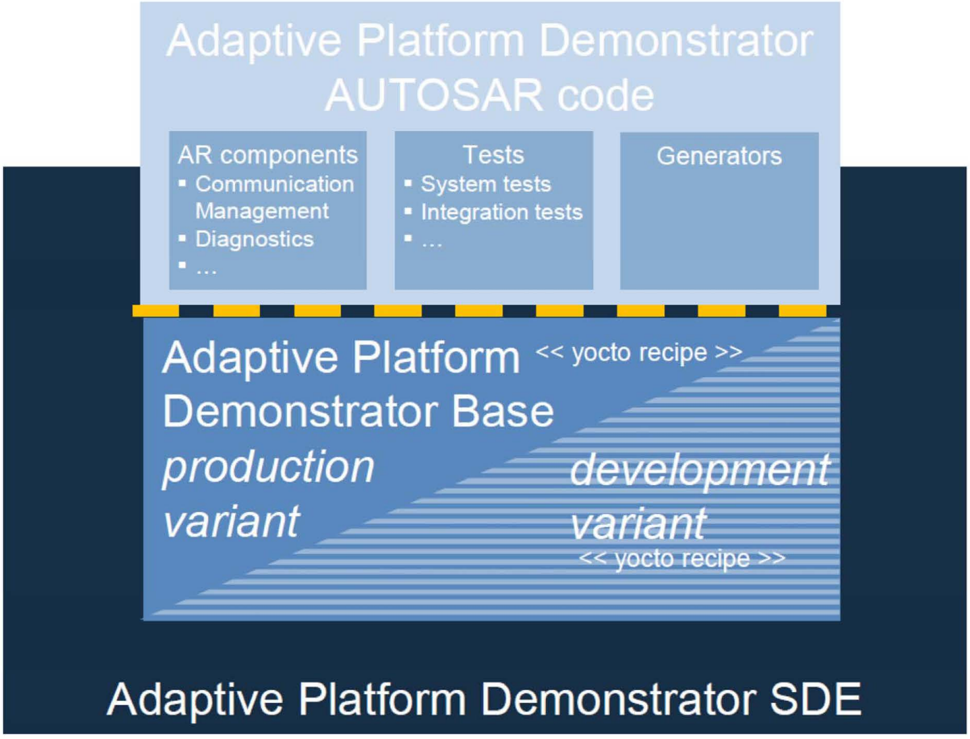

Figure 1.3: Overview of the AUTOSAR Adaptive Platform Demonstrator

1.6 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 R24-11. This chapter is structured according to the clusters of AUTOSAR release R24-11.
- 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 R24-11

This chapter contains a summary of changes which have been implemented since the previous release R23-11.

Several concepts affecting the Adaptive Platform have been introduced with release R24-11 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- 2.1.1.5 have been introduced.

2.1.1.1 Automotive API

Concept introduces new functional cluster Automotive API Gateway. It offers interface that allows standardized data-centric communication with the vehicle using VISS protocol. Such functional cluster requires ARXML representation of the VSS data model described in the released Technical Report.

2.1.1.2 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.

The new Requirements document for Safe Hardware Acceleration in AUTOSAR AP was provided in a ’Draft’ status. Explanatory document was renamed to be in consistency with all other documents related to the concept.

2.1.1.3 Adaptive Platform Machine Configuration

The configuration of a machine on the AUTOSAR adaptive platform by means of an M2-based modeling exposes several significant drawbacks that are addressed by the concept.

The stated objective of the concept is to provide an alternative modeling approach that overcomes existing drawbacks and restrictions and provides domain experts with a better and more flexible approach to describe the machine/target configuration on the AUTOSAR adaptive platform.

In comparison to an equivalent implementation of the configuration model for the AUTOSAR adaptive platform on the M2 level, the chosen approach provides various benefits that are explained in TPS APMC.

The results of the concept will change the canonical way for machine configuration on the AUTOSAR adaptive platform, but not immediately after the concept has been finalized.

It is planned to keep the existing M2-based configuration model for some time to give all stakeholders time to switch to the new configuration approach at a convenient point in time.

The big picture of the manifest configuration on the AUTOSAR adaptive platform looks very similar to the configuration model on the AUTOSAR classic platform, as described by the TPS Ecu Configuration.

The difference between the ApmC and EcuC is that the approach on the AUTOSAR adaptive platform is simpler because it is not constrained by some of the boundary conditions that apply for the AUTOSAR classic platform.

2.1.1.4 DDS Protocols

The goal of this concept part is to centralize and homogenize DDS Service-oriented communication protocols by means of:

1. Identifying Service-oriented usages of DDS in Classic and Adaptive Platforms, such as Service Instance Discovery and Provided-Required Service Instance Communication
2. Defining mappings of usages to DDS concepts (entities, types, topics, QoS policies) and mechanisms (standard API calls)
3. Refactoring the DDS Network Binding or Adaptive Platform Communications Management Functional Cluster to rely upon and reference mappings where necessary

2.1.1.5 Deterministic Communication with TSN

This concept part "Extension for AP" introduced support for a subset of the IEEE1722 specified transport protocols (AAF, 61883_IIDC, RVF, CRF, TSCF and NTSCF) on AP. An application can utilize the transport of data via Raw Data Streams, where the whole subset of IEEE1722 specified transport protocol is supported. An application can utilize TSCF and NTSCF transport protocol via signal to service translation, where signals transported as IEEE1722 encapsulated bus frames (CAN or LIN) are mapped to services.

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

Concept Name	Specification Long Name	Affected Standard	Concept Lifecycle
Automotive API	**Requirements on Automotive API Gateway**	Adaptive Platform, Foundation	draft
	Specification of Automotive API Gateway		
	Explanation of Automotive API		
	Technical Report on VSS Representation		
	Specification of Adaptive Platform Core		
	Explanation of Adaptive Platform Design		
	Explanation of Adaptive Platform Software Architecture		
Safe API for hardware accelerators	Design guidelines for using parallel processing technologies on Adaptive Platform	Adaptive Platform, Foundation	draft
	Requirements on Safe Hardware Acceleration		
	Explanation of Safe API for hardware accelerators		
Adaptive Platform Machine Configuration	**Adaptive Platform Machine Configuration**	Adaptive Platform, Foundation	draft
DDS Protocols	Specification of Communication Management	Adaptive Platform, Classic Platform, Foundation	draft
Deterministic Communication with TSN	Requirements on Communication Management	Adaptive Platform, Classic Platform	draft
	Specification of Communication Management		

▽

△

Concept Name	Specification Long Name	Affected Standard	Concept Lifecycle
	Explanation of Adaptive Platform Software Architecture		
	Specification of Manifest		
	Specification of Raw Data Stream		
	Explanation of Adaptive Platform Design		
	Guidelines for using Adaptive Platform interfaces		

Table 2.1: Impact of concepts

2.1.3 Validated Concepts

The following concepts have been validated:

- DDS Enhanced Discovery
- Tracing for Adaptive Platform

2.2 Specifications

2.2.1 New Specifications

The following new specifications have been introduced via concepts:

- Requirements on Safe Hardware Acceleration (UID 1087, RS)
- Adaptive Platform Machine Configuration (UID 1112, TPS)
- Requirements on Automotive API Gateway (UID 1115, RS)
- Specification of Automotive API Gateway (UID 1116, SWS)
- Technical Report on VSS Representation (UID 1117, TR)
- Explanation of Automotive API (UID 1121, EXP)

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

- Technical Report on Demands and Constraints on Base Software (UID 1120, TR)

2.2.2 Migrated Specifications

With this release, the following specifications have been moved from ... to ...:

- none

2.2.3 Obsolete Specifications

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

- none

2.2.4 Removed Specifications

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

- none

2.2.5 Reworked Specifications

The following specifications have been changed fundamentally in this release:

This release removes support for Offset Time Bases because AUTOSAR does no longer see relevant use cases. All traceable items that only related to Offset Time Bases are removed. From traceable items that related to both Synchronized and Offset Time Bases, the part related to Offset Time Bases is removed. Any explanatory text that related to Offset Time Bases is removed. This change affects the following documents of the AUTOSAR Adaptive Platform:

- Specification of Manifest (UID 713, TPS)
- Specification of Time Synchronization (UID 880, SWS)

For effects on the other AUTOSAR platforms, refer to their release overview documents.

2.2.6 Moved Specification Parts

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

- List of Adaptive Platform Functional Clusters (UID 862, TR) to Standardized M1 Models used for the Definition of AUTOSAR (UID 636, MOD)

2.2.7 Renamed Specifications

The following specifications have been renamed in this release:

- none

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 [1].

Please be aware that according to [2], 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:

- ReleaseDocumentation
- AdaptiveFoundation
- AdaptiveServices
- General
- MethodologyAndManifests

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

Long Name	File Name	Life cycle changes
ReleaseDocumentation		
Adaptive Platform Release Overview	AUTOSAR_AP_TR_ReleaseOverview	
AUTOSAR Adaptive Platform Specification Hashes	AUTOSAR_AP_TR_Specification Hashes	
AdaptiveFoundation		
Explanation of ara::com API	AUTOSAR_AP_EXP_ARAComAPI	
Explanation of IPsec Implementation Guidelines	AUTOSAR_AP_EXP_IPsec ImplementationGuidelines	
Explanation of Identity and Access Management	AUTOSAR_AP_EXP_IdentityAnd AccessManagement	
Explanation of MACsec and MKA Protocols implementation and configuration guidelines	AUTOSAR_AP_EXP_MACsec	
Explanation of Service-Oriented Vehicle Diagnostics	AUTOSAR_AP_EXP_SOVD	
Requirements on Communication Management	AUTOSAR_AP_RS_Communication Management	
Requirements on Cryptography	AUTOSAR_AP_RS_Cryptography	
Requirements on Execution Management	AUTOSAR_AP_RS_Execution Management	
Requirements on Identity and Access Management	AUTOSAR_AP_RS_IdentityAndAccess Management	
Requirements on Operating System Interface	AUTOSAR_AP_RS_OperatingSystem Interface	
Requirements on Persistency	AUTOSAR_AP_RS_Persistency	
Requirements on Platform Health Management	AUTOSAR_AP_RS_PlatformHealth Management	
Specification of Communication Management	AUTOSAR_AP_SWS_Communication Management	
Specification of Adaptive Platform Core	AUTOSAR_AP_SWS_Core	
Specification of Cryptography	AUTOSAR_AP_SWS_Cryptography	
Specification of Diagnostics	AUTOSAR_AP_SWS_Diagnostics	
Specification of Execution Management	AUTOSAR_AP_SWS_Execution Management	
Specification of Firewall for Adaptive Platform	AUTOSAR_AP_SWS_Firewall	

▽

△

Long Name	File Name	Life cycle changes
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	
Specification of Time Synchronization	AUTOSAR_AP_SWS_TimeSynchronization	
Integration of DDS Security	AUTOSAR_AP_TR_DDSSecurityIntegration	
AdaptiveServices		
Explanation of Automotive API	AUTOSAR_AP_EXP_AutomotiveAPI	initial release
Explanation of Safe API for hardware accelerators	AUTOSAR_AP_EXP_SafeHardwareAccelerationAPI	
Explanation of Sensor Interfaces	AUTOSAR_AP_EXP_SensorInterfaces	
Requirements on Automated Driving Interfaces	AUTOSAR_AP_RS_AutomatedDrivingInterfaces	
Requirements on Automotive API Gateway	AUTOSAR_AP_RS_AutomotiveAPIGateway	initial release
Requirements on Safe Hardware Acceleration	AUTOSAR_AP_RS_SafeHardwareAcceleration	initial release
Requirements of State Management	AUTOSAR_AP_RS_StateManagement	
Requirements on Update and Configuration Management	AUTOSAR_AP_RS_UpdateAndConfigurationManagement	
Requirements on Vehicle Update and Configuration Management	AUTOSAR_AP_RS_VehicleUpdateAndConfigurationManagement	
Specification of Automotive API Gateway	AUTOSAR_AP_SWS_AutomotiveAPIGateway	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	
Technical Report on Operating System Tracing Interface	AUTOSAR_AP_TR_OperatingSystemTracingInterface	
Technical Report on VSS Representation	AUTOSAR_AP_TR_VSSRepresentation	initial release
General		
Explanation of ARA Applications in Rust	AUTOSAR_AP_EXP_ARARustApplications	

▽

△

Long Name	File Name	Life cycle changes
Guidelines for using Adaptive Platform interfaces	AUTOSAR_AP_EXP_Interfaces Guidelines	
Design guidelines for using parallel processing technologies on Adaptive Platform	AUTOSAR_AP_EXP_Parallel ProcessingGuidelines	
Explanation of Adaptive Platform Design	AUTOSAR_AP_EXP_PlatformDesign	
Explanation of Adaptive Platform Software Architecture	AUTOSAR_AP_EXP_SWArchitecture	
General Requirements specific to Adaptive Platform	AUTOSAR_AP_RS_General	
Technical Report on Demands and Constraints on Base Software	AUTOSAR_AP_TR_Demands ConstraintsBaseSW	
System Tests for Adaptive Platform Demonstrator	AUTOSAR_AP_TR_SystemTests	
MethodologyAndManifests		
Collection of blueprints for AUTOSAR Adaptive Platform M1 models	AUTOSAR_AP_MOD_General Blueprints	
Requirements on Manifest Specification	AUTOSAR_AP_RS_Manifest Specification	
Specification of Platform Types for Adaptive Platform	AUTOSAR_AP_SWS_PlatformTypes	
Adaptive Platform Machine Configuration	AUTOSAR_AP_TPS_Machine Configuration	initial release
Specification of Manifest	AUTOSAR_AP_TPS_Manifest Specification	
Specification of Timing Extensions for Adaptive Platform	AUTOSAR_AP_TPS_Timing Extensions	
Methodology for Adaptive Platform	AUTOSAR_AP_TR_Methodology	

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
898	Specification of Network Management	SWS	4.1
858	Specification of Persistency	SWS	4.2
851	Specification of Platform Health Management	SWS	4.3
903	Specification of Adaptive Platform Core	SWS	4.4
880	Specification of Time Synchronization	SWS	4.5
723	Specification of Diagnostics	SWS	4.6
978	Specification of Intrusion Detection System Manager for Adaptive Platform	SWS	4.7

Table 4.1: Overview of known technical deficiencies

4.1 Specification of Network Management

The Adaptive Network Management

- is only supporting UdpNM
- does not allow node detection but handles incoming repeat message requests
- cannot be configured as the master network coordinator
- does not support coordinated shutdown using the information in CBV
- does not support passive mode and passive startup
- does not provide service interfaces to update and access user data

4.2 Specification of Persistency

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

- Daisy chaining (i.e. forwarding Supervision Status or Checkpoint 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.
- 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.
- Supervision is not supported for non-reporting processes. 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.

4.4 Specification of Adaptive Platform Core

- 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 Functional specification describes some behavior informally that should rather be given as specification items.
- Some parts of the chapter API specification (SynchronizedPoolResource, UnsynchronizedPoolResource, Executor, String, some constructors) have not been assigned a thread safety category as specified in the chapter Thread-safety.

4.5 Specification of Time Synchronization

- Thread safety is missing for all callbacks

4.6 Specification of Diagnostics

- Thread safety is not fully defined

4.7 Specification of Intrusion Detection System Manager for Adaptive Platform

- Open errors regarding thread safety and arerror are accepted.

5 Release History

5.1 Release R24-11

The following deliverables had major changes.

Name	Specification history entry
Adaptive Platform Release Overview	• Release Life Cycle Status: R24-11 is in Evolution, R24-11 supersedes R23-11
Adaptive Platform Machine Configuration	• Initial release of the document
Design guidelines for using parallel processing technologies on Adaptive Platform	• Minor narrative text updates
Explanation of ARA Applications in Rust	• No content changes
Explanation of Adaptive Platform Design	• Introduction of Automotive API Gateway chapter. • Updates of State Management, Update and Config Management, Persistency, Log and Trace, Raw Data Stream, Time Synchronization, Adaptive Core chapters according to the respective SWS changes. • Removal of Health Channels and System Health Management from Platform Health Management chapter. • Various clean-ups and minor updates.
Explanation of Adaptive Platform Software Architecture	• Added the Functional Cluster Automotive API Gateway. • Extended the Use Case View and Runtime View with use cases and scenarios. • Added information about supported bus types and protocols. • Described design principles and patterns used in the Adaptive Platform.
Explanation of Automotive API	• Initial release
Explanation of IPsec Implementation Guidelines	• No content changes
Explanation of Identity and Access Management	• No content changes
Explanation of MACsec and MKA Protocols implementation and configuration guidelines	• Editorial changes
Explanation of Safe API for hardware accelerators	• No content changes
Explanation of Sensor Interfaces	• No content changes
Explanation of Service-Oriented Vehicle Diagnostics	• No content changes
Explanation of ara::com API	• Updated tutorial examples to use complete ARXML
General Requirements specific to Adaptive Platform	• Clarifications in relation to AP stabilization • The use of noexcept • Thread Safety • Versioning of Service Interface • Definition of rollback semantics • MISRA C++ 2023 • Own namespace for Platform Extension interfaces
Guidelines for using Adaptive Platform interfaces	• Updates of Raw Data Stream, Platform Health Management, Update and Config Management, Adaptive Core chapters according to the respective SWS changes. • Various clean-ups.
Integration of DDS Security	• No content changes

▽

△

Name	Specification history entry
Methodology for Adaptive Platform	• Editorial changes
Requirements of State Management	• Remove support for DiagnosticReset
Requirements on Automated Driving Interfaces	• No content changes;
Requirements on Automotive API Gateway	• Initial release
Requirements on Communication Management	• Communication Groups Removed • IEEE1722 Streams • Editorial changes
Requirements on Cryptography	• No content changes
Requirements on Execution Management	• Requirements for deterministic execution are removed • Minor changes, document clean up
Requirements on Identity and Access Management	• No content changes
Requirements on Manifest Specification	• Changed status of requirements to valid; • Added requirement for Firewall configuration; • minor corrections, clarifications and editorial changes;
Requirements on Operating System Interface	• Minor changes, document clean up
Requirements on Persistency	• Replaced chapter Conventions by external artifact
Requirements on Platform Health Management	• Removed obsolete Health Channel requirements RS_PHM_00102, RS_PHM_09255 and RS_PHM_09257 • Removed obsolete Daisy Chaining requirements RS_PHM_00108 and RS_PHM_00109 • Removed requirements RS_PHM_00001, RS_PHM_00002 and RS_PHM_00003 • Clarified requirement RS_PHM_00107
Requirements on Safe Hardware Acceleration	• Initial release
Requirements on Update and Configuration Management	• Introduced suspend and resume requirement • Introduced standardized trace points requirement
Requirements on Vehicle Update and Configuration Management	• Introduced standardized trace points requirements
Specification of Adaptive Platform Core	• Extend specification of ara::core::MemoryResource and derived classes • Add full specification of ara::core::Optional, ara::core::Variant, and ara::core::StringView • Add T& specializations of ara::core::Optional and ara::core::Result • Specify exception safety and thread safety of ARA APIs • Mandate ara::core::ErrorCode as ErrorType of Result and Future/Promise • Various extensions and fixes to the C++ data types • Adapt the document to the new template with a generated Chapter 8 • Refine specification about platform initialization • Enable ara::core::Initialize() take command line arguments • Refine Violation specification of ARA APIs with standardized Violation messages
Specification of Automotive API Gateway	• Initial release

▽

△

Name	Specification history entry
Specification of Communication Management	• Added IEEE1722 Signal-Service-Translation network binding • Fixed interoperability issues between AP and CP regarding SOME/IP Error responses • Added handling of Transport Fault Conditions • Added support for MACsec cryptograhic protection • Reworked ara::com C++ API decriptions based on header files with generated API Tables • Harmonized error codes and violations for all ara::com APIs • Editorial changes and bugfixes
Specification of Cryptography	• Improve Crypto APIs by removing default values and using out parameter instead of returning ara::core::vector • Improve Keyslot and Certificate access control • Improve UpdatesObserver functional specification • Improve Signature handling • Add Log and Trace functionality to Crypto API
Specification of Diagnostics	• Document quality improvement, clarifications and fixing bugs • Document structure updated • Formalized generated interface classes for DID, RID and DataElements • Standardized Violations added • Term Reentrancy is changed to Concurrency • Support DoIP amendment 2023 protocol version 4 • Harmonization with CP • Explicit no-debouncing for ara::diag::monitor • SecurityEvents added
Specification of Execution Management	• Requirements for deterministic execution are removed • Added support for Function Group access control • API refinement (define lifetime of arguments, error codes, removed thread-safety information, return values) • Added clarification on Execution Dependencies
Specification of Firewall for Adaptive Platform	• Various API changes (specification of error domain, thread-safety, error recoverability, ...) • Migration to new SW template • Updated SEv context data specification table
Specification of Intrusion Detection System Manager for Adaptive Platform	• Introduce QualifiedEventsReceiver and ReportingMode Provider. • Support versioning of context data. • Update document structure.
Specification of Language Binding for modeled AP data types	• Added C++ language binding of ApApplicationErrors • Moved all header file specification to chapter 8
Specification of Log and Trace	• Application term and references deleted and replaced by Adaptive Application • Refactoring and editorial changes

▽

△

Name	Specification history entry
Specification of Manifest	• Improve configuration of diagnostic transport layers • Improve configuration of IEEE 1722-related transport • Offset Time Base Feature Removal • minor corrections / clarifications / editorial changes
Specification of Network Management	• Added NetworkManagementPortInterface and Port Prototype • Reworked ara::com C++ API decriptions based on header files with generated API Tables • Editorial changes
Specification of Operating System Interface	• Extended ARTI tracing interface and related Log messages. • Minor changes, document clean up
Specification of Persistency	• Improved update sequence, allowing for independent updates of configuration • Storage access type no longer dependent on port type • Renamed error kIsEof as ara::per::PerErrc.kEof, added production errors and log messages • Clarified synchronous behavior of SWS_PER_00049.ara::per::KeyValueStorage::SyncToStorage and SWS_PER_00122.ara::per::ReadWrite Accessor::SyncToFile • ara::per::ReadWriteAccessor changed to "final"
Specification of Platform Health Management	• Removed Health Channels from specification • Changed return type of RecoveryHandler API to ara::core::Future • Set SupervisedEntity class and RecoveryAction APIs to final • Update of threadsafety and exception safety information on APIs • Introduction of violation messages to ReportCheckpoint() and constructors
Specification of Platform Types for Adaptive Platform	• No content changes
Specification of Raw Data Stream	• Added API for IEEE 1722 Audio Video Transport Protocol streams • Editorial changes and bugfixes
Specification of Sensor Interfaces	• fixed missing service interface versions;
Specification of State Management	• Remove support for DiagnosticReset • Remove obsolete mentionings of CommunicationGroups • Remove non-implementable requirements for customer-specific StateManagement implementation • Adopt document to new SWS template
Specification of Time Synchronization	• Offset Time Base feature removed
Specification of Timing Extensions for Adaptive Platform	• No content changes
Specification of Update and Configuration Management	• New suspend and resume requirements introduction • Progress monitoring refactoring
Specification of Vehicle Update and Configuration Management	• Refactored progress monitoring • Adapted V-UCM to UCM new suspend and resume requirement

▽

△

Name	Specification history entry
System Tests for Adaptive Platform Demonstrator	• Removed test case for PHM related to health channel • Updated CM test scenarios
Technical Report on Demands and Constraints on Base Software	• Initial Release
Technical Report on Operating System Tracing Interface	• Removed 'draft' state from Specification Items • Correction of API syntax(ArtiVersionInfoType, Calling Context) • Minor editorial changes
Technical Report on VSS Representation	• Initial release

Table 5.1: Overview of specification release histories