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

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	19-03
Release Life Cycle Status	R19-03 is in Development, R19-03 supersedes R18-10

Document Change History

Date	Release	Changed by	Change Description
2019-03-29	19-03	AUTOSAR Release Management	Updated according to R19-03
2018-10-31	18-10	AUTOSAR Release Management	Initial release

Table of Contents

1 Introduction

1.1 Scope of this document

This document provides an overview on the AUTOSAR standard Adaptive Platform Release 19-03.

1.2 Dependencies to other standards

This release of the Adaptive Platform depends on the standard Foundation in Release 1.5.1, which

- defines protocols implemented by Adaptive Platform and
- contains the project objectives and the common requirements from which the features of the Adaptive Platform are derived.

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

1.3 Content of chapters

This document is structured as follows:

- Chapter 2 provides an introduction to the Adaptive Platform and its standardization approach.
- Chapter 3 provides a list of documentation references.
- Chapter 4 contains the overview of specifications comprising the Release 19-03. This chapter is structured according to the clusters of AUTOSAR Release 19-03.
- Chapter 5 provides a summary of changes since the previous release of the Adaptive Platform.
- Chapter 6 contains remarks about known technical deficiencies.
- Chapter 7 contains the detailed release history of all released specifications.
- Chapter 8 provides a set of definitions aimed to increase the understanding of the content of this document and the Release 19-03.

2 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:

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

2.1 Release strategy

The Adaptive Platform will be in lifecycle state “Development” according to AUTOSAR’s lifecycle model for its standards (see figures 1 and 2) until it will reach a certain maturity level. Until November 2019, AUTOSAR will release a new version of the Adaptive Platform in a 6-month cycle. The lifecycle state “Development” implies that users of the Adaptive Platform have no guarantee on backward compatibility. Consequently, all requirements have the lifecycle status *draft*.

The release R19-03 is considered to be a stabilization release.
Thus the focus of the release is on stabilization of the existing features from R18-10 and not on additional features.
According to the release scheme of AUTOSAR the release R19-03 is a Revision.

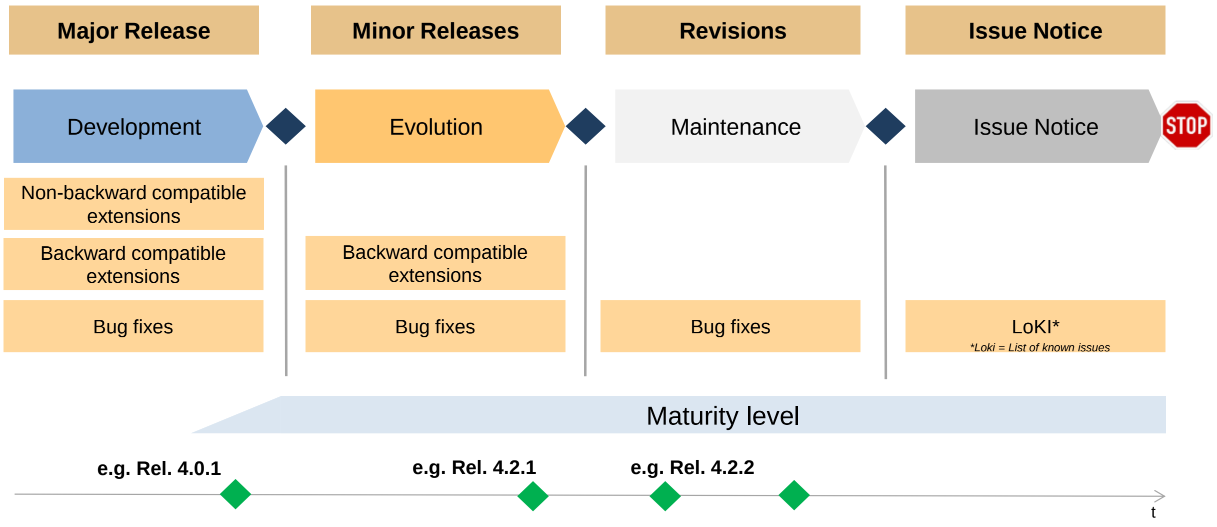

Figure 1: Lifecycle model of AUTOSAR standards and its application to Classic Platform

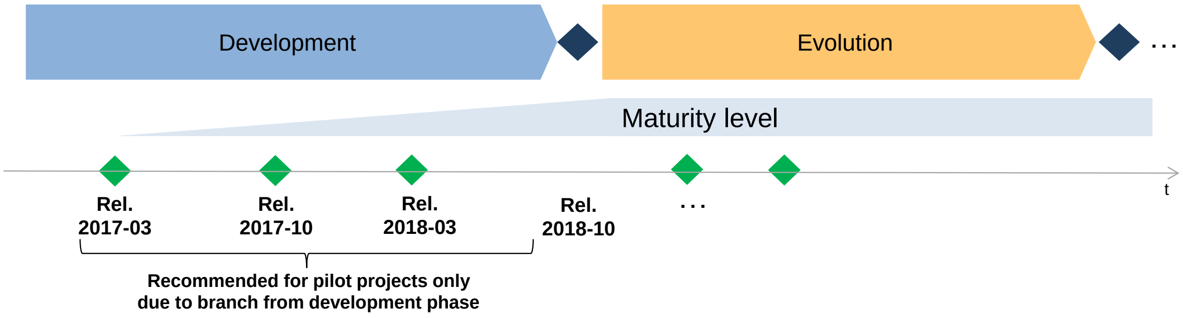

Figure 2: Application of AUTOSAR lifecycle to Adaptive Platform (Release numbers only exemplary)

Apart from the regular specifications that have been elaborated in intensive expert discussion, according to current planning, the releases may comprise draft specifications to indicate the intended scope and direction of discussion to the AUTOSAR development community.

The following must be considered for the draft specifications:

1. Minimal or less quality measures are being applied during development
2. No indication of quality / stability due to a lack of discussions between the AUTOSAR partners

2.2 Parallel validation of specification via implementation

The Adaptive Platform is partially validated through an AUTOSAR-internal implementation: the Adaptive Platform Demonstrator. This Demonstrator is available to all the partners and can provide further details to understand the underlying concepts of the Adaptive Platform. The Demonstrator provides an implementation example based on the specification rather than a reference implementation. All further development based on the Demonstrator 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 maintainability and extensibility of the Demonstrator software implementation.
The Demonstrator comes with traceability up to the specifications and explanatory documents or the so-called Functional Cluster Design specifications.
Additionally AUTOSAR develops System Tests to test the demonstrator implementation against the AUTOSAR requirements. These tests are also part of the release.

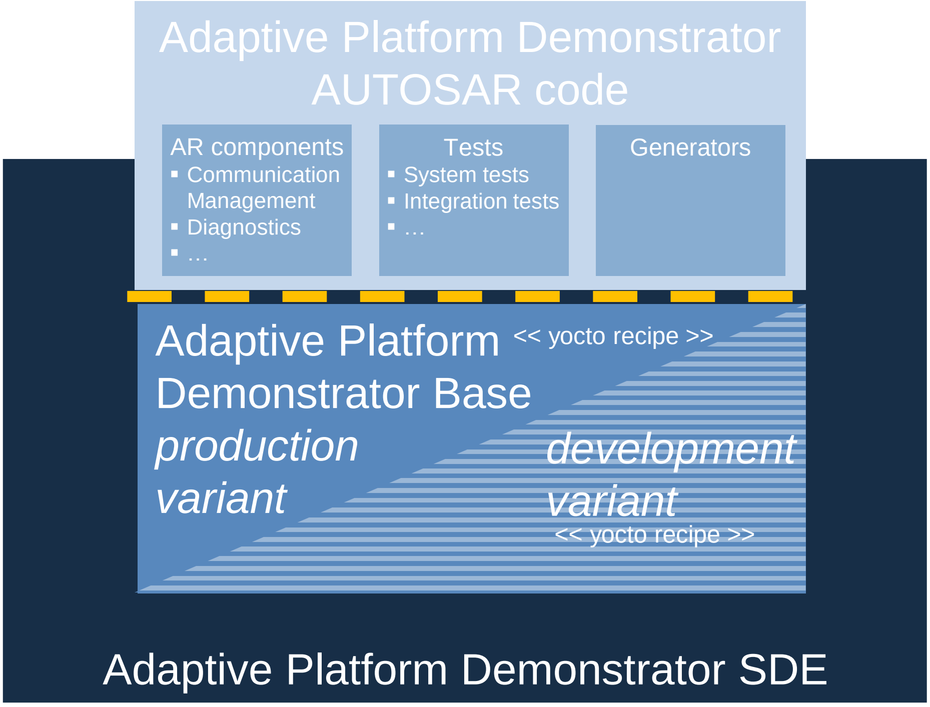

Figure 3: Overview of the AUTOSAR Adaptive Platform Demonstrator

2.3 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.

3 Related documentation

[1] AUTOSAR Specifications in general

[2] Glossary

4 Specification overview

4.1 Release 18-10

The published specifications are divided into the following clusters:

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

All specifications contain requirements which are identified by special braces:
⌈ = Beginning of the requirement content
⌋ = End of the requirement content
In addition, all XML files and schemas are considered as requirements.

The assignment of specifications to clusters is shown below.

Long Name	File Name	Life cycle changes	Draft Specification
Release Documentation			
Adaptive Platform Release Overview	AUTOSAR_TR_AdaptivePlatformReleaseOverview		
AUTOSAR Adaptive Platform Specification Hashes	AUTOSAR_TR_AdaptivePlatformSpecificationHashes		
Adaptive Foundation			
Explanation of ara::com API	AUTOSAR_EXP_ARAComAPI		
Explanation of IPsec Implementation Guidelines	AUTOSAR_EXP_IPsecImplementationGuidelines	Initial release	
Requirement on Time Synchronization for Adaptive Platform	AUTOSAR_RS_TimeSynchronization		
Requirements on Communication Management	AUTOSAR_RS_CommunicationManagement		
Requirements on Cryptography	AUTOSAR_RS_Cryptography		
Requirements on Execution Management	AUTOSAR_RS_ExecutionManagement		
Requirements on Identity and Access Management	AUTOSAR_RS_IdentityAndAccessManagement		
Requirements on Operating System Interface	AUTOSAR_RS_OperatingSystemInterface		
Requirements on Persistency	AUTOSAR_RS_Persistency		
Requirements on Platform Health Management for	AUTOSAR_RS_PlatformHealthManagement		

Long Name	File Name	Life cycle changes	Draft Specification
Adaptive Platform			
Requirements on Security Management for Adaptive Platform	AUTOSAR_RS_SecurityManagement		
Specification of Communication Management	AUTOSAR_SWS_CommunicationManagement		
Specification of Cryptography for Adaptive Platform	AUTOSAR_SWS_Cryptography		X
Specification of Execution Management	AUTOSAR_SWS_ExecutionManagement		
Specification of Identity and Access Management	AUTOSAR_SWS_IdentityAndAccessManagement		
Specification of Log and Trace	AUTOSAR_SWS_LogAndTrace		
Specification of Operating System Interface	AUTOSAR_SWS_OperatingSystemInterface		
Specification of Persistency	AUTOSAR_SWS_Persistency		
Specification of Platform Health Management for Adaptive Platform	AUTOSAR_SWS_PlatformHealthManagement		
Specification of RESTful communication	AUTOSAR_SWS_REST		
Specification of Time Synchronization for Adaptive Platform	AUTOSAR_SWS_TimeSynchronization		
Adaptive Services			
Explanation of Sensor Interfaces	AUTOSAR_EXP_SensorInterfaces	Initial release	
Requirements of State Management	AUTOSAR_RS_StateManagement	Initial release	
Requirements on Update and Configuration Management	AUTOSAR_RS_UpdateAndConfigManagement		
Specification for Network Management	AUTOSAR_SWS_NetworkManagement		
Specification of Diagnostics	AUTOSAR_SWS_Diagnostics		
Specification of State Management	AUTOSAR_SWS_StateManagement	Initial release	
Specification of Update and Configuration Management	AUTOSAR_SWS_UpdateAndConfigManagement		
Methodology and Manifests			
Meta Model	AUTOSAR_MMOD_MetaM		

Long Name	File Name	Life cycle changes	Draft Specification
	odel		
Meta Model-generated XML Schema	AUTOSAR_MMOD_XMLSchema		
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		
Supplementary material of the AUTOSAR XML Schema	AUTOSAR_TR_XMLSchemaSupplement		
Collection of blueprints for AUTOSAR Adaptive Platform models	AUTOSAR_MOD_AdaptivePlatformGeneralBlueprints	Initial release	
General			
Design guidelines for using parallel processing technologies on Adaptive Platform	AUTOSAR_EXP_ParallelProcessingGuidelines		
Explanation of Adaptive Platform Design	AUTOSAR_EXP_PlatformDesign		
Explanation of Safety Overview	AUTOSAR_EXP_SafetyOverview		
Functional Cluster Shortnames	AUTOSAR_TR_FunctionalClusterShortnames		
General Requirements specific to Adaptive Platform	AUTOSAR_RS_General		
General Specification of Adaptive Platform	AUTOSAR_SWS_General		
Guidelines for the use of the C++14 language in critical and safety-related systems	AUTOSAR_RS_CPP14Guidelines		
Guidelines for using Adaptive Platform interfaces	AUTOSAR_EXP_AdaptivePlatformInterfacesGuidelines	Initial release	
Specification of Core Types for Adaptive Platform	AUTOSAR_SWS_CoreTypes		
System Tests of Adaptive Platform	AUTOSAR_TR_AdaptivePlatformSystemTests		

4.2 Release 19-03

The published specifications are divided into the following clusters:

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

The assignment of specifications to clusters is shown below.

Long Name	File Name	Life cycle changes	Draft Specification
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_IPsec ImplementationGuidelines		
Requirement on Time Synchronization for Adaptive Platform	AUTOSAR_RS_Time Synchronization		
Requirements on Communication Management	AUTOSAR_RS_Communication Management		
Requirements on Cryptography	AUTOSAR_RS_Cryptography		
Requirements on Execution Management	AUTOSAR_RS_Execution Management		
Requirements on Identity and Access Management	AUTOSAR_RS_IdentityAnd AccessManagement		
Requirements on Operating System Interface	AUTOSAR_RS_OperatingSystem Interface		
Requirements on Persistency	AUTOSAR_RS_Persistency		
Requirements on Platform Health Management for Adaptive Platform	AUTOSAR_RS_PlatformHealth Management		
Requirements on Security Management for Adaptive Platform	AUTOSAR_RS_SecurityManagement		
Specification of Communication Management	AUTOSAR_SWS_Communication Management		
Specification of Cryptography for Adaptive Platform	AUTOSAR_SWS_Cryptography		X

Long Name	File Name	Life cycle changes	Draft Specification
Specification of Execution Management	AUTOSAR_SWS_Execution Management		
Specification of Identity and Access Management	AUTOSAR_SWS_IdentityAnd AccessManagement		
Specification of Log and Trace	AUTOSAR_SWS_LogAndTrace		
Specification of Operating System Interface	AUTOSAR_SWS_OperatingSystem Interface		
Specification of Persistency	AUTOSAR_SWS_Persistency		
Specification of Platform Health Management for Adaptive Platform	AUTOSAR_SWS_Platform Health Management		
Specification of RESTful communication	AUTOSAR_SWS_REST		
Specification of Time Synchronization for Adaptive Platform	AUTOSAR_SWS_Time Synchronization		
Adaptive Services			
Explanation of Sensor Interfaces	AUTOSAR_EXP_SensorInterfaces		
Requirements of State Management	AUTOSAR_RS_StateManagement		
Requirements on Update and Configuration Management	AUTOSAR_RS_UpdateAndConfig Management		
Specification for Network Management	AUTOSAR_SWS_Network Management		
Specification of Diagnostics	AUTOSAR_SWS_Diagnostics		
Specification of State Management	AUTOSAR_SWS_State Management		
Specification of Update and Configuration Management	AUTOSAR_SWS_UpdateAnd ConfigManagement		
Methodology and Templates			
Meta Model	AUTOSAR_MMOD_MetaModel		
Meta Model-generated XML Schema	AUTOSAR_MMOD_XMLSchema		
Supplementary material of the AUTOSAR XML Schema	AUTOSAR_TR_XMLSchema Supplement		
Methodology and Manifests			
Collection of blueprints for AUTOSAR Adaptive Platform models	AUTOSAR_MOD_AdaptivePlatform GeneralBlueprints		
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_AdaptivePlatform Types		

Long Name	File Name	Life cycle changes	Draft Specification
General			
Design guidelines for using parallel processing technologies on Adaptive Platform	AUTOSAR_EXP_ParallelProcessingGuidelines		
Explanation of Adaptive Platform Design	AUTOSAR_EXP_PlatformDesign		
Explanation of Safety Overview	AUTOSAR_EXP_SafetyOverview		
Functional Cluster Shortnames	AUTOSAR_TR_FunctionalClusterShortnames		
General Requirements specific to Adaptive Platform	AUTOSAR_RS_General		
General Specification of Adaptive Platform	AUTOSAR_SWS_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		
Specification of Core Types for Adaptive Platform	AUTOSAR_SWS_CoreTypes		
System Tests of Adaptive Platform	AUTOSAR_TR_AdaptivePlatformSystemTests		

All specifications contain requirements which are identified by special braces:
⌈ = Beginning of the requirement content
⌋ = End of the requirement content
In addition, all XML files and schemas are considered as requirements.

5 Summary of changes

This chapter contains a summary of changes which were implemented since the previous release.

- Regular maintenance of document.
- New Documents have been created and are released for the first time.
- Documents went through a major rework.

5.1 Release 18-10

In AUTOSAR R18-10 the Adaptive Platform has been elaborated further to harmonize with the Classic Platform.

Additionally the System Tests have been reworked to test the Adaptive Platform Demonstrator against the Requirements Specifications of the AUTOSAR Adaptive Platform.

Further improvements have been applied to all the specifications that are part of R18-10.

5.1.1 Concepts

The following concepts in 5.1.1.1 – 5.1.1.2 have been introduced.

5.1.1.1 Formal Model Query and Blueprint Derivation Mechanisms

The concept "Formal Model Query and Blueprint Derivation Mechanisms" is released as draft and will be validated in 2019.

The concept completes the extension of AUTOSAR Classic (CP) and Adaptive platforms (AP) with the AUTOSAR Model Query Language (ARMQL). This new language enables a highly efficient collaboration of AUTOSAR user due to resolving variation points in CP and AP by the same mechanism. It is published in textual form, not bound to a specific tool and significant better understandable as the existing Formula Language.

5.1.1.2 Extended Serialization for Data Structures in SOME/IP with tag/length/value encoding (TLV)

The concept TLV is released as draft and will be validated in 2019.

The concept adds support for improved forward and backward compatibility during evolution of interfaces on SOME/IP protocol-level. Moreover, the concept integrates support for optional struct members on protocol-level and application-level (RTE and ara::com).

5.1.2 Specifications

5.1.2.1 New Specifications

The following documents and templates were added to the R18-10:

- Specification of State Management (UID 908, SWS)
- Requirements of State Management (UID 909, RS)
- Explanation of Sensor Interfaces (UID 913, EXP)
- Guidelines for using Adaptive Platform interfaces (UID 929, EXP)
- Explanation of IPsec Implementation Guidelines (UID 930, EXP)
- Collection of blueprints for AUTOSAR Adaptive Platform models (UID 931, MOD)

5.1.2.2 Migrated Specifications

With this release, the following specifications were moved from Adaptive Platform to the Foundation standard:

- Requirements on Adaptive Network Management (UID 898, RS), merged with the new document Requirements on Network Management (UID 927, RS)

5.1.2.3 Obsolete Specifications

The following specification is set to status “obsolete” in this release:

- No specifications were set to “obsolete”.

5.1.2.4 Cancelled Specifications

The following specification is canceled in this release:

- No specifications were canceled.

5.1.3 Release Documentation

There were no major changes regarding the Release Documentation.

5.2 Release 19-03

Additionally the System Tests have been reworked to test the Adaptive Platform Demonstrator against the Requirements Specifications of the AUTOSAR Adaptive Platform.

For the “Specification of Cryptography for Adaptive Platform” the prefix for specification items has changed from “SWS_CRYPTO_XXXXX” to “SWS_CRYPT_XXXXX” due to an overlap with the Classic Platform “Specification of Crypto Driver”.

Further improvements have been applied to all the specifications that are part of R19-03.

5.2.1 Concepts

No Concepts have been introduced with AP R19-03.

5.2.2 Specifications

5.2.2.1 New Specifications

No new specifications have been introduced with AP R19-03.

5.2.2.2 Migrated Specifications

No specifications have been migrated with AP R19-03.

5.2.2.3 Obsolete Specifications

The following specification is set to status “obsolete” in this release:

- Guidelines for the use of the C++14 language in critical and safety-related systems (UID 839, RS). The work has been handed over to MISRA and the document will no longer be maintained by AUTOSAR.

5.2.2.4 Cancelled Specifications

No specifications were canceled.

5.2.3 Release Documentation

There were no major changes regarding the Release Documentation.

6 Remarks to known technical deficiencies

The technical deficiencies per specification are – if applicable – mentioned inside the respective specification in a chapter called “Known Limitations”.

There are the following technical deficiencies which are not related to a particular specification: None

6.1 Release 18-10

6.1.1 Known technical deficiencies per document

Document Long Name	Known Limitations
Methodology for Adaptive Platform	The following sections are still under discussion: • Section 2.4.5 (Set up an initial Machine), • Section 2.4.6 (Create Software Packages) • Section 2.4.7 (Management and provision of Software Packages)
Specification of Communication Management	• 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 Signal to Service mapping in this specification does not contain behavior specification. • The E2E communication protection works only for events which are polled and which are transmitted at least once per fault tolerant time interval. This means, it requires: o Periodic invocation of the method Update in a polling mode o Periodic or mixed-periodic invocation of the method Send o In case Update or Send are not invoked periodically, then some communication failure modes are not detected (loss, delay and possibly also repetition). In this case, if E2E is used, then additional measures need to be taken at application level to address those non-detected failure modes. • The values of some E2E parameters are defined by the standard and shall not be changed. • Optional method arguments: The Specification does not support the existence of optional method arguments • Some limitations apply for optional arguments introduced with the TLV serialization

Document Long Name	Known Limitations
Specification of Core Types for Adaptive Platform	• 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 7 (“Functional Specification”) describes some behavior informally that should rather be given as specification items.
Specification of Cryptography for Adaptive Platform	• The entire Crypto API is provisional and likely to be fully revised in the upcoming releases. • The content was not updated for the current release.
Specification of Diagnostics	• It is probable that shared_ptr will change in an upcoming release.
Specification of Execution Management	• Support for Resource Limitation is not complete • Support for Fault Tolerance (reacting to, and coping with errors in EM itself) is not complete • Support for establishment of a Trusted Platform ([RS_EM_00014]) is not specified
Specification of Identity and Access Management	• The topic of providing identity information of Adaptive Applications to PEPs is still under discussion. • Requirements and specification details regarding Application ID / Application Instance ID and providing application identity in general may be affected by this discussion and may change accordingly. • There is no API specification available yet
Specification of Manifest	• The AUTOSAR SWS REST defines a low-level API for REST-based communication. • The content of section 11, on the other hand, applies for the configuration of a not-yet standardized API on top of the ara::rest API.
Specification of Network Management	• Only supports UdpNM at the moment • Does not allow node detection - Repeat Message State requests but handles incoming requests • Cannot be configured as master network coordinator • Does not support coordinated shutdown using information from CBV • Does not support passive mode or passive startup • Mapping of logical networks to BitVector positions (in the message) not available in the manifest • New: ○ User data cannot be accessed from applications in a standardized way – the service interface was moved as NM now interacts with SM instead of applications. User data has to be clarified for next release.

Document Long Name	Known Limitations
Specification of Operating System Interface	• There is currently no (sufficient) API providing periodic time-based processing to fulfill [RS_OSI_00102]. • Authorized access to APIs is not supported [RS_OSI_00205, RS_OSI_00208].
Specification of Persistency	• The interpretation of deployment related information in the AUTOSAR model is not yet covered in detail in this specification. In addition, the concept of a roll-back after an update is not yet supported. • The configuration of encryption for Persistency is not defined in Specification of Manifest.
Specification of Platform Health Management for Adaptive Platform	• Daisy chaining (i.e. forwarding Supervision Status, Checkpoint or Health channel information to an entity external to PHM or another PHM instance) is currently not supported in this document release. • Platform Health Management configuration related to Supervision Modes is not fully supported in this document release. • An API to inform Supervised Entities about the Supervision states is available only in polling mode. No API using notification mode is available in this release. • Interface with the Diagnostic Manager is not specified in this release.
Specification of RESTful communication	• The interfaces are only specified to the point to make semantics clear • The error handling for RESTful communication is currently limited due to the fact that errors are not reported in the context of a request transmission
Specification of State Management	• Section 7.3 on Component States are partially discussed and not finished yet. • The RequestRestart and Communication Control for Diagnostic reasons is a proposal only and subject to change.
Specification of Time Synchronization for Adaptive Platform	• The Time Synchronization module is bound to Adaptive Platform Systems. • For the TS, it is necessary that at least there is one TBR in the system, otherwise no functionality can be provided to the Adaptive Applications • API design is not fully compliant to Adaptive Platform Design Rules which request the usage of UpperCamelCase.

Document Long Name	Known Limitations
System Tests of Adaptive Platform	• Test cases may not cover whole RS as specified against test cases • Test setup figure may not exactly reflect the test configuration • Test cases may not be fully covered by corresponding system test implementations • System test cases are just examples, since there could be many ways to define and implement use case scenarios • DIAG does not have any RS traceability, as it is intended to reuse WP-T results • LT does not have any RS traceability. Traceability will be added in next release

6.2 Release 19-03

6.2.1 Known technical deficiencies per document

Document Long Name	Known Limitations
Specification of Time Synchronization for Adaptive Platform	• The Time Synchronization (TS) module is bound to Adaptive Platform Systems • For the TS, it is necessary that there is at least one Time Base Resource (TBR) in the system, otherwise no functionality can be provided to the Adaptive Applications (i.e. the Adaptive Applications would not get any handle for TBR). • API design is not fully compliant to Adaptive Platform Design Rules which requests the usage of UpperCamelCase. • The current concept on TimeSync is not in line with the port prototype approach. • The topic on InstanceSpecifier is not yet finalized. • Further changes to be expected in R19-11.
Specification of Identity and Access Management	• A detailed API will be added in a future release. • For other Functional Clusters, implementation on Policy Enforcement Points are envisaged for the next release (R19-11). • Currently limited to ara::com

Document Long Name	Known Limitations
Specification of Diagnostics	• Chapter 7 Functional specification is not reworked for the experimentally introduced diagnostic interfaces of chapter 8.5 C++ Diagnostic API Interfaces • OBD ISO 15031 and WWH OBD ISO 27145 is not supported by the DM. • Software Cluster/Diagnostic Server instances are supported by DM interfaces but are not specified in detail. • DoIP edge node is not supported by the DM. • The following UDS services are not implemented by the DM: o 0x23 ReadMemoryByAddress o 0x24 ReadScalingDataByIdentifier o 0x2A ReadDataByPeriodicIdentifier o 0x2C DynamicallyDefineDataIdentifier o 0x2F InputOutputControlByIdentifier o 0x38 RequestFileTransfer o 0x3D WriteMemoryByAddress o 0x83 AccessTimingParameter o 0x84 SecuredDataTransmission o 0x87 LinkControl • Sub-functions of UDS services are implemented according to ISO 14229-1 unless explicitly stated. • The UDS mirror event memory is not supported by the DM. As a result of this, the DM does not support the UDS service. o 0x19 with subfunction 0x0F (reportMirrorMemoryDTCByStatusMask) o 0x19 with subfunction 0x10 (reportMirrorMemoryDTCExtDataRecordBy-DTCNumber) o 0x19 with subfunction 0x11 (reportNumberOfMirrorMemoryDTCByStatus-Mask) • The OBD/WWH OBD is not supported by the DM. As a result of this, the DM does not support the UDS service. o 0x19 with subfunction 0x05 (reportDTCStoredDataByRecordNumber) o 0x19 with subfunction 0x12 (reportNumberOfEmissionsOBDDTCByStatus-Mask) o 0x19 with subfunction 0x13 (reportEmissionsOBDDTCByStatusMask) o 0x19 with subfunction 0x42 (reportWWHOBDDTCByMaskRecord) o 0x19 with subfunction 0x55 (reportWWHOBDDTCWithPermanentStatus) • Event Memory: Variant handling at runtime for events/DTCs is not supported. • Event Memory: Details for combined events are not specified. • Event Memory: Event displacement is not supported. The DM stores for each DTC related data. • Event Memory: Interface to read the number of event memory entries is not supported.

Document Long Name	Known Limitations
	- Event Memory: Internal configuration parameters and DM values as extended data are not supported. - Persistent Storage of failed attempts to change security level : After each increment of the attempt counter, it shall be persisted to survive accidental or intended resets. Here the option to select the persistent storage is mandatory in Adaptive Autosar.
Specification of Cryptography for Adaptive Platform	The current version of this document is missing some functionality that is available in the AUTOSAR Classic Platform: - Secure Counter: There is currently no API available to access secure counter primitives that an implementation may provide. The following functional domains and descriptions are still missing in the current version of Crypto API specification: - Asynchronous interfaces: Currently there is only a synchronous API specification and asynchronous behavior (if required) should be implemented on the consumer application level. It can be done via utilization of dedicated execution threads for long-time operations. - X.509 certificates support: Crypto API doesn’t provide complete specification of the X.509 certificates management on the client (ECU) side yet. Current version of Crypto API specifies only minimal subset of interfaces responsible for basic X.509 functionality and related on utilization of cryptographic algorithms. Current API supports extraction and parsing of only basic attributes of X.509 certificates and certification requests. An extension of the API specification by additional interfaces dedicated for complete support of X.509 extensions is planned for the next release of this specification. Note: Generally current specification of the X.509 Provider API is preliminary and subject for extensions and changes. - Memory management: In the current version of the specification Crypto Provider supports the safety-aligned memory management concept suitable for real-time applications. Up to the next release this concept will be extended for X.509 Provider too. Application of any memory management mechanisms specific for support of asynchronous calls (like std::future) is in scope a developer responsibility.

Document Long Name	Known Limitations
	• Formats of cryptographic objects: Current version of Crypto API has minimal support of well-known cryptographic formats encoding/decoding: support of only DER and PEM encoding for X.509 certificates and certificate signing requests is required from any implementation of Crypto API. For other cryptographic objects an implementation can support only "raw" formats. Following extension of the Crypto API by unified interfaces for encoding/decoding of complex objects to standard formats is planned for the next release of this specification. • Key slots modeling: Now Crypto API defines some structures that should be produced as a result of the key slots modeling process. But the whole concept of the key slots modeling is not finished yet. Therefore Key Storage API can be updated slightly for next release in order to extend support of the ara::core::InstanceSpecifier type as one of mechanisms for the Logical Key Slot identification. • Functional specification: Detailed functional specification (chapter 7) is not available yet and will be elaborated for next Autosar AP release. • Depth of inheritance: The performance of the inheritance tree design applied for the Crypto Provider interfaces is still subject to further investigation. Therefore a redesign of APIs defined in namespace ara::crypto::cryp may be executed for next Autosar release, in order to achieve a very limited inheritance depth (or completely "flattened" API design).
Requirements on Execution Management	• The following requirements are described within this document but not otherwise considered in this release: [RS_EM_00050] – System-wide coordination [RS_EM_00051] – External trigger conditions [RS_EM_00111] – Identification of Processes [RS_EM_00014] – Trusted Platform. • The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.
Requirements on Operating System Interface	• The following requirements are described within this document but not otherwise considered in this release: [RS_OSI_00204] [RS_OSI_00208] • The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.

Document Long Name	Known Limitations
Specification of Communication Management	• 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: o 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 Signal to Service mapping in this specification does not contain behavior specification. • The E2E communication protection works only for events which are polled and which are transmitted at least once per fault tolerant time interval. This means, it requires: o Periodic invocation of the method GetNewSamples (see [SWS_CM_00701]) in a polling mode o Periodic or mixed-periodic invocation of the method Send (see [SWS_CM_00162] and [SWS_CM_90437]) • In case GetNewSamples or Send are not invoked periodically, then some communication failure modes are not detected (loss, delay and possibly also repetition). In this case, if E2E is used, then additional measures need to be taken at application level to address those non-detected failure modes. • The values of the following E2E parameters are defined by the standard and shall not be changed. See E2E Protocol Specification. o dataIdMode o counterOffset o crcOffset o dataIdNibbleOffset o offset • EndToEndTransformationComSpecProps are not supported. • The following limitations regarding optionality introduced with the Tag-Length-Value • serialization principle described in SOME/IP Protocol Specification and Specification of Manifest apply: o Optional method arguments: The Specification does not support the existence of optional method arguments.

Document Long Name	Known Limitations
Specification of Execution Management	• The following functionality is mentioned within this document but is not fully specified in this release: o Section 7.7 Resource Limitation and Section 7.8 Fault Tolerance – these sections have been expanded in this release but are not complete. In particular the contents will be expanded with more properties and formal requirements in the next release. • The following functionality is not specified in this release: o Support of a Trusted Platform ([RS_EM_00014]). • Section 6.1 details requirements from Execution Management Requirement Specification [1] that are not elaborated within this specification. The presence of these requirements in this document ensures that the requirement tracing is complete and also provides an indication of how Execution Management will evolve in future releases of the AUTOSAR Adaptive Platform. • The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.
Specification of Log and Trace	• The provided Logging framework API is designed to be independent from the underlying Logging back-end implementation and as such doesn't impose limitations.
Specification of Operating System Interface	• The following functionality is mentioned within this document but is not fully specified in this release: o The currently known limitations are the requirements in Requirements on Operating System Interface which are listed within Appendix A. o There is currently no sufficient API providing periodic time-based processing to fulfill [RS_OSI_00102]. This will be defined in a future release.
Specification of Persistency	• The configuration of encryption for Persistency is not defined in Specification of Manifest.
Specification of Platform Health Management for Adaptive Platform	• [SWS_PHM_00110] 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. (RS_PHM_00108, RS_PHM_00109) • [SWS_PHM_00111] Platform Health Management configuration related to Supervision Modes is not fully supported in this document release. (RS_PHM_00104, RS_HM_09253) • [SWS_PHM_00112] An API to inform Supervised Entities about the Supervision states is available only in polling mode. No API using notification mode is available in this release. (RS_HM_09237) • Interface with the Diagnostic Manager is not specified in this release.

Document Long Name	Known Limitations
Specification of RESTful communication	• The interfaces are only specified to the point to make semantics clear. To be precise this document does not yet fully specify the qualification C++ functions noexcept, overloading of functions to provide move semantics for optimization purposes nor does it claim to be const-correct. Move semantics in particular are specified where required for semantic correctness only. Also only HTTP network binding aspects of the AUTOSAR meta model are currently supported by the SWS_REST. No modeling of the Rest-ServiceInterface internal structure is possible with the current SWS_REST. The error handling for RESTful communication is currently limited due to the fact that errors are not reported in the context of a request transmission.
Specification of Network Management	• The Adaptive Network Management currently only supports UdpNM. • The Adaptive Network Management does not allow node detection (Repeat Message State) but only handles incoming requests. • The Adaptive Network Management cannot be configured as the master network coordinator. • The Adaptive Network Management does not support coordinated shutdown using the information in CBV. • The Adaptive Network Management does not support passive mode and passive startup. • Passive start-up would mean that a node has started (i.e. goes to Normal mode), but the network has been woken up by another node. • Modeling part for mapping the logical networks to the BitVector positions as defined in chapter 7.3 is not available in the manifest. • Update and access of User Data was removed as the service interface to Applications has been removed. State Management will control the network request/release and it must be clarified if user data changes/indications shall be done via State Management or directly by applications.
Specification of State Management	• The following functionality is mentioned within this document but is not (fully) specified in this release: ○ Section 7.2 This document will show the basic principles of the intended functionality of State Management. To enable State Management to be portable, in future versions of this document standardized fields and values shall be introduced. ○ Section 7.3 Communication Control for Diagnostic reasons this is not yet discussed with Adaptive Diagnostics. ○ Section 7.3 RequestRestart for Diagnostic reasons this is discussed with Adaptive Diagnostics, but some interface details are not yet finalized.

Document Long Name	Known Limitations
Specification of Update and Configuration Management	• UCM is not responsible to initiate the update process. UCM realizes a service interface to achieve this operation. The user of this service interface is responsible to verify that the vehicle is in a safe state before executing a software update procedure on demand. It is also in the responsibility of the user to communicate with other AUTOSAR Adaptive Platforms or AUTOSAR Classic Platforms within the vehicle. Therefore management of software dependencies between different physical or virtual ECU software platforms is currently out of UCM's scope but will be managed by the UCM Master which will be introduced in the next release. • The UCM receives a locally available software package for processing. The software package is usually downloaded from the OEM backend. The download of the software packages has to be done by another application, i.e. UCM does not manage the connection to the OEM backend. Prior to triggering their processing, the software packages have to be transferred to UCM by using the provided ara::com interface. • The UCM update process is designed to cover updates on use case with single AUTOSAR Adaptive Platform. UCM can update Adaptive Applications, the AUTOSAR Adaptive Platform itself, including all functional clusters and the underlying OS. Distinction between different types of updates, such as safety critical updates vs infotainment updates, isn't addressed in this release. Currently such distinction shall be included into vendor specific meta-data. • The UCM is not responsible for enforcing authentication and access control to the provided interfaces. The document currently does not provide any mechanism for the confidentiality protection as well as measures against denial of service attacks. The assumption is that the platform preserves the integrity of parameters exchanged between UCM and its user.
Specification of Core Types for Adaptive Platform	• The specification of Promise misses the specializations for <void> and <T&> that std::promise defines. • The specification lacks comparison operators for class ErrorCode. • The specification of class Result uses an insufficient conditional noexcept classifier that does not consider the properties of the <E> type. • 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 7 ("Functional Specification") describes some behavior informally that should rather be given as specification items.

Document Long Name	Known Limitations
System Tests of Adaptive Platform	• Test cases may not cover whole RS specifications as specified against test cases • Test setup figure may not exactly reflect the test configuration • Test cases may not be fully covered by corresponding system test implementations • System test cases are just examples, since there could be many ways to define and implement use case scenarios • DIAG does not have any RS traceability, as it is intended to reuse WP-T results • LT does not have any RS traceability. Traceability will be added in next release • In the E2E test case, the common parts of the E2E profiles are checked
Specification of Manifest	• The AUTOSAR SWS REST defines a low-level API for REST-based communication. The content of section 11, on the other hand, applies for the configuration of a not-yet standardized API on top of the ara::rest API.
Methodology for Adaptive Platform	• The sections related to the deployment of Software Packages, i.e., Section 2.4.5 (Set up an initial Machine), Section 2.4.6 (Create Software Packages) and Section 2.4.7 (Management and provision of Software Packages), are still under discussion.

7 Release history

7.1 Release 18-10

Release R18-10 was originally released on the 31st of October 2018.

Name	Specification history entry
Design guidelines for using parallel processing technologies on Adaptive Platform	• Minor changes
Explanation of Adaptive Platform Design	• Changes to reflect the latest SWS contents
Explanation of ara::com API	• Added InstanceIdentifier and InstanceSpecifier explanation • Restructured chapter structure • Adapt FindService signatures • Added sample code for event usage • Restructured chapter structure • Proxy and skeleton instances are not copyable • Changed certain data types to ara::core namespace. • Adapted to new error handling based on ara::core::ErrorCode
Explanation of Automated Driving Interfaces	• Initial release
Explanation of Ipsec Implementation Guidelines	• Initial release
Explanation of Safety Overview	• Restructuring of document inspired by ISO 26262 • Rework chapters 1-5 • Add functional safety requirements table
Functional Cluster Shortnames	• Renaming of Identity and Access Management, Operating System Interface, Update and Configuration Management
General Requirements specific to Adaptive Platform	• More details to clause 1 Scope of document given • Former chapter 4.3 on Design requirements putted below chapter 4.2 Non-functional requirements • Following requirements have been revised: [RS_AP_00111], [RS_AP_00113], [RS_AP_00114], [RS_AP_00115], [RS_AP_00122], [RS_AP_00120], [RS_AP_00121], [RS_AP_00124], [RS_AP_00125] • Following requirements have been deleted: [RS_AP_00117], [RS_AP_00118] • Following requirements have been added: [RS_AP_00127], [RS_AP_00128], [RS_AP_00129], [RS_AP_00130], [RS_AP_00131], [RS_AP_00132], [RS_AP_00134]
General Specification of Adaptive Platform	• SWS_AP_00003 removed since there is no demand in RS_AP_00003 which requires it anymore
Guidelines for the use of the C++14 language in critical and safety-related systems	• Added traceability for ISO 26262 (B.6) • New rules resulting from continued analysis of the C++ Core Guideline • Finished addressing MISRA review comments of the 2017-03 release • Improvements of already existing rules, more details in the Changelog (D.3) • Marked the specification as obsolete

Name	Specification history entry
Guidelines for using Adaptive Platform interfaces	• Initial release
Methodology for Adaptive Platform	• Renamed Application Manifest to Execution Manifest • Moved references from spec.item body to foot notes • Editorial changes
Requirement on Time Synchronization for Adaptive Platform	• Minor changes and bugfixes • Editorial changes
Requirements of State Management	• Initial release
Requirements on Communication Management	• Minor changes and bugfixes
Requirements on Cryptography	• Removed: [RS_CRYPTO_02303] and [RS_CRYPTO_02402] • Updated: [RS_CRYPTO_02006]
Requirements on Execution Management	• Removed: RS_EM_00003, RS_EM_00004, RS_EM_00110 and RS_EM_00111. • Added: [RS_EM_00014]
Requirements on Identity and Access Management	• Functional Description of Capabilities • Functional Description of Access • Control for Inter-Platform Communication • Requirement for Superset Manifests
Requirements on Manifest Specification	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation in AUTOSAR_RS_ManifestSpecification
Requirements on Operating System Interface	• Removed: RS_OSI_00102 and RS_OSI_00105 • Added: [RS_OSI_00207], [RS_OSI_00208].
Requirements on Persistency	• Restructured document • Added resource handling • Added UCM related requirements
Requirements on Platform Health Management for Adaptive Platform	• minor corrections / clarifications / editorial changes
Requirements on Security Management for Adaptive Platform	• Chapter 2.3 ’Protected Runtime Environment’ revised
Requirements on Update and Configuration Management	• Requirements on Operating System updates • Requirement on Security • Requirement on History
Specification of Communication Management	• Introduced Adaptive Core types • Introduced exception-less API • Refined DDS network binding • Minor changes and bugfixes

Name	Specification history entry
Specification of Core Types for Adaptive Platform	• Add chapter 2 with acronyms • Add chapter 4 with limitations of the current specifications • Add chapter 5 with dependencies to other modules • Add chapter 7 • Add classes representing the approach to error handling to chapter 8 • Adapt classes Future and Promise to the error handling approach • Add global functions for initialization and shutdown of the framework • Add class InstanceSpecifier to chapter 8 • Add more types and functions from the C++ standard
Specification of Cryptography for Adaptive Platform	• No changes
Specification of Diagnostics	• Diagnostic Protocol replaced by Diagnostic Conversations • ResponseOnEvent, CommunicationControl, EcuReset added • Chapter 7 overall rework and updates • Chapter 8 split into chapter 8 (C++ API) and chapter 9 (Service Interfaces)
Specification of Execution Management	• Refinement of Deterministic Execution • Updated Process lifecycle to clarify • Process and Execution States • Updated Application Recovery Actions
Specification of Identity and Access Management	• Reworked functional specification • Removed API specification for general rework
Specification of Log and Trace	• Changed initialization APIs • Improved references • Log file definition
Specification of Manifest	• Finish introduction of CppImplementationDataType • Support for optional elements in structures • Rework configuration of adaptive platform modules
Specification of Network Management	• Updated interaction with State Management • Removed APIs and Services (interaction is done via SM) • Temporary removed user data access to applications
Specification of Operating System Interface	• Add Resource Control • Added Shared object support
Specification of Persistency	• Introduction of ara::core types and switch to exceptionless API • Rework of redundancy approach • Support for resource limitation • Improvements and harmonization of KeyValueStorage and FileProxy API
Specification of Platform Health Management for Adaptive Platform	• Described the interfaces with functional clusters execution management and state management
Specification of Platform Types for Adaptive Platform	• Rework to CppImplementationDataTypes
Specification of RESTful communication	• Updated APIs to use ara::core types • Minor editorial fixes
Specification of State Management	• Initial release

Name	Specification history entry
Specification of Time Synchronization for Adaptive Platform	• Minor changes and bugfixes • Editorial changes
Specification of Update and Configuration Management	• Updated interaction other functional clusters like PER and EMO/SM • Introduction of vehicle package distribution
System Tests of Adaptive Platform	• Added RS traceability for test cases • Added ISO 9646 framework and mapping on system test architecture • Added more test cases for CM, REST, EMO, and UCM

7.2 Release 19-03

Release R19-03 was originally released on the 29th of March 2019.

Name	Specification history entry
Adaptive Platform Release Overview	• Updated according to R19-03
Design guidelines for using parallel processing technologies on Adaptive Platform	• Minor changes
Explanation of Adaptive Platform Design	• Changes to reflect the latest SWS contents. Chapter 17.4 C++ coding guidelines deleted.
Explanation of ara::com API	• Changed explanation of Event reception due to new ara::com API
Explanation of IPsec Implementation Guidelines	• No changes
Explanation of Safety Overview	• No content changes • minor layout changes
Explanation of Sensor Interfaces	• No content changes
Functional Cluster Shortnames	• No content changes
General Requirements specific to Adaptive Platform	• No content changes
General Specification of Adaptive Platform	• No content changes
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	• Clause 4 revised to reflect the updated design on State Management
Methodology for Adaptive Platform	• No content changes

Name	Specification history entry
Requirement on Time Synchronization for Adaptive Platform	• No content changes
Requirements of State Management	• Updated requirements due to reworked intended design
Requirements on Communication Management	• No content changes
Requirements on Cryptography	• Editorial changes and rephrasing • Improved requirements description and rationale (Updated : [RS_CRYPTO_02001] [RS_CRYPTO_02002] [RS_CRYPTO_02003] [RS_CRYPTO_02004] [RS_CRYPTO_02005] [RS_CRYPTO_02007] [RS_CRYPTO_02009] [RS_CRYPTO_02109] [RS_CRYPTO_02116] [RS_CRYPTO_02202] [RS_CRYPTO_02206])
Requirements on Execution Management	• Updated: RS_EM_00008 and RS_EM_00010
Requirements on Identity and Access Management	• Introduction of Grant concept
Requirements on Manifest Specification	• editorial changes
Requirements on Operating System Interface	• Added: use case for [RS_OSI_00201] and [RS_OSI_00202]
Requirements on Persistency	• Updated introduction • Information on wear leveling • Finalization was removed
Requirements on Platform Health Management for Adaptive Platform	• removed references to RS_Main_00330
Requirements on Security Management for Adaptive Platform	• Unnecessary requirement [RS SEC 05006] removed
Requirements on Update and Configuration Management	• Spelling fixes • Minor explanation improvements
Specification for Network Management	• Introduced Service Interface for interaction via SM • Introduced possibility to group PNCs/Channels/VLANs

Name	Specification history entry
Specification of Communication Management	• Predictable Resource Allocation for Samples • Usage of Future::Get/Wait with an unreliable transport • Removed exceptions on reception of malformed messages • Changes to Identity and Access Management to incorporate Grant design • Minor changes and bugfixes
Specification of Core Types for Adaptive Platform	• Add specification of the template specialization Result<void, E>
Specification of Cryptography for Adaptive Platform	• Direct" prefix of Crypto API is removed, because now it is single • All bugs found after R18-03 are fixed • Crypto API is converted for usage of basic ara::core types • Crypto API is converted for support of the "Exception-less" approach
Specification of Diagnostics	• Document quality improvement and fixing bugs • Introduced ara::diag interfaces in draft state
Specification of Execution Management	• Refinement of State Management semantics • Document structure modified to reflect current template
Specification of Identity and Access Management	• Reworked chapter 7 to incorporate new concept of Grants • Incorporation of several bug tickets
Specification of Log and Trace	• Changed APIs (Logstream, Logmanager, Logging) • Refactoring and editorial changes
Specification of Manifest	• Introduction of Diagnostic Port Interfaces • Overhaul of Software Cluster and introduction of Software Package • Support for Identity and Access Management • Network Management Configuration
Specification of Operating System Interface	• Clarified that PSE51 following POSIX-1003.1-2003 is the currently-targeted version. • Minor changes in tracing, clean up
Specification of Persistency	• Improved naming of classes/methods/functions • Reworked installation/update • Support for parallel execution in multiple threads • Cleaned up usage of ara::core concepts
Specification of Platform Health Management for Adaptive Platform	• Modified the API for Supervised Entity and Health Channel • Modified the interface with the Execution Manager
Specification of Platform Types for Adaptive Platform	• minor corrections / clarifications / editorial changes
Specification of RESTful communication	• No content changes
Specification of State Management	• Removed components • RequestState and ReleaseRequest are now deprecated • State Managements internal states can now be influenced by "Trigger" and are distributed by "Notifier" fields
Specification of Time Synchronization for Adaptive Platform	• Functional description detached from actual API • Improved resource discovery

Name	Specification history entry
Specification of Update and Configuration Management	• Updating Package Management state machine • New requirements for robustness against reset • Improving specification item atomicity • Fixing errors in chapter Service Interfaces
Supplementary material of the AUTOSAR XML Schema	• Updated according to R19-03
System Tests of Adaptive Platform	• Changed format for RS traceability items • Added new section and test cases for Time Synchronization • Added more test cases for CM, EMO, and DIAG

8 Appendix

8.1 Definitions

8.1.1 Release number

AUTOSAR applies a four-digit numbering scheme Ryy-mm to identify releases.

- yy = year
- mm = month

8.1.2 Specification item and requirement life cycle states

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.
- **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. This is the default for Adaptive Platform.
- **Obsolete:** This indicates that the related entity is obsolete and will be removed in the next release.

If there is no life cycle state information stated then the state is Draft.

Requirements: The requirement attribute “type” indicates the life cycle state of the requirement. The states are the same as the specification item states.