Document Title	Requirements on Manifest Specification
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	712

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R21-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2021-11-25	R21-11	AUTOSAR Release Management	Removed requirement for • REST
2020-11-30	R20-11	AUTOSAR Release Management	Added requirements for • Raw Data Stream Deployment
2019-11-28	R19-11	AUTOSAR Release Management	Added requirements for • Service Versioning • Signal-To-Service Translation • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• editorial changes;
2018-10-31	18-10	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2018-03-29	18-03	AUTOSAR Release Management	Added requirements for • Configuration of logging and tracing • Time Synchronization • DDS • Configuration of function groups
2017-10-27	17-10	AUTOSAR Release Management	Added requirements for • Software Component System Design • Security and Safety • Signal-based communication • REST

2017-03-31	17-03	AUTOSAR Release Management	• Initial release

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.

Table of Contents

1 Scope of Document

This document specifies requirements on the Manifest of the Adaptive Platform.

The Manifest is a formal specification of configuration content that ships along a given piece of software and is used to deploy the software in the field.

The Manifest content can be divided into two areas:

- The Application-related Manifest can be taken to configure the deployment of software, but on the other hand, the same piece of configuration information can also be used in the design of the software.

- The machine-related Manifest content describes the configuration of a machine that runs an AUTOSAR adaptive platform, i.e. without any application running on the machine.

2 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template [1], chapter Support for Traceability.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template [1], chapter Support for Traceability.

3 Requirements Specification

This chapter describes all requirements driving the work to define the TPS_ManifestSpecification [2].

[RS_MANI_00015]{DRAFT} **Definition of the nature of a manifest** ⌈

Description:	The Manifest specification shall define the term "manifest".
Rationale:	A unique understanding of the term "manifest" is an immediate prerequisite for the usage of a manifest. Since the term is so prominent on the AUTOSAR adaptive platform it is important to provide a proper definition.
Dependencies:	–
Use Case:	Readers want to be able to fully grasp the meaning of the AUTOSAR specification.
Supporting Material:	–

⌋*(RS_Main_00002, RS_Main_00503)*

3.1 Execution Manifest Overview

The manifest contains all necessary information about an AUTOSAR application that is necessary for the configuration of the middleware.

[RS_MANI_00035]{DRAFT} **Definition of an uploadable software package** ⌈

Description:	The Manifest specification shall support the definition of an uploadable software package such that the content of a real software configuration uploaded to a target platform can be formalized.
Dependencies:	–
Use Case:	An integrator takes application software delivered by software developers and executes the integration to the intended target platform. This process creates a bunch of new model elements that are required to configure the target platform such that the uploaded software can work. The integrator therefore needs a means to identify all related model content needed for configuration on the target platform. This approach also needs to be formalized in the meta-model.
Supporting Material:	–

⌋*(RS_Main_00150)*

3.1.1 Execution Manifest Requirements

[RS_MANI_00001]{DRAFT} **Adaptive AUTOSAR Application** ⌈

Description:	The Manifest specification shall be able to describe an Adaptive AUTOSAR Application.
Rationale:	The description of the Adaptive AUTOSAR Application represents the prerequisite for the creation of a manifest, i.e. the information the Adaptive Platform needs to integrate the application into the middleware and execution model.
Dependencies:	–
Use Case:	Software that is taken as an input for an integration step to configure the software for an instance of the *AUTOSAR adaptive platform*.
Supporting Material:	see also definition of Adaptive Application in [3]

⌋*(RS_Main_00150)*

3.1.1.1 Application Design

[RS_MANI_00004]{DRAFT} **Support of application design** ⌈

Description:	The Manifest specification shall provide design elements that are necessary to develop and to describe software for the Adaptive AUTOSAR platform.
Rationale:	Standardized software component design description.
Dependencies:	–
Use Case:	Development of Application Software Components.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00300, RS_Main_00080, RS_Main_00310)*

[RS_MANI_00002]{DRAFT} **Declaration of provided and required services in an application** ⌈

Description:	The Manifest specification shall support the declaration of services that the application implements and uses. This shall be the only way to describe the interaction of an application with other applications and the platform.
Rationale:	A contract between service consumers and service providers needs to be defined.
Dependencies:	–
Use Case:	The declaration of services allows service consumers to discover services and to use them. Only the Service description is visible to the outside world.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00140, RS_Main_00080)*

[RS_MANI_00003]{DRAFT} **Specification of service interfaces** ⌈

Description:	The Manifest specification shall allow the specification of service interfaces that define the service functionality.
Rationale:	A service interface defines the way in which applications can interact and exchange information.
Dependencies:	–
Use Case:	Application Design, generation of C++ proxies and skeletons from the service interface description in order to implement the service interface of an application.
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00060, RS_Main_00190, RS_Main_00080)*

[RS_MANI_00017]{DRAFT} Specification of the mapping of Service Interfaces ⌈

Description:	The Manifest specification shall allow the specification of a mapping of service interfaces such that the granularity of service-oriented communication can be controlled by the model author.
Rationale:	A service interface defines the way in which an application can interact and exchange information. However, there are cases where the choice made by application developers is not met by the designers of external communication.
Dependencies:	–
Use Case:	Application designers and designers of external communication have different approaches onto granularity of service definition.
Supporting Material:	–

⌋*(RS_Main_00320)*

[RS_MANI_00005]{DRAFT} Configuration of diagnostic capabilities of an application ⌈

Description:	The Manifest specification shall support the configuration of diagnostic capabilities of an application.
Rationale:	Each application shall be able to describe how it interacts with the Diagnostic Management.
Dependencies:	–
Use Case:	Description of diagnostic services the application implements and provides to the Diagnostic Management.
Supporting Material:	–

⌋*(RS_Main_00260, RS_Main_00080)*

[RS_MANI_00016]{DRAFT} Usage of data types specifically on the AUTOSAR adaptive platform ⌈

Description:	The Manifest specification shall support the usage of data types as defined in classic AUTOSAR and shall also support additional or deviating data types natively available on the AUTOSAR adaptive platform.
Rationale:	An application on the AUTOSAR adaptive platform shall be able to describe the usage of data types.
Dependencies:	–
Use Case:	Usages of data types that are natively available on the AUTOSAR adaptive platform, e.g. `vector` or `string`.
Supporting Material:	–

⌋*(RS_Main_00002, RS_Main_00190)*

[RS_MANI_00039]{DRAFT} Usage of implementation specific data types ⌈

Description:	The Manifest specification shall support the usage of implementation specific data types in the model.
Rationale:	Implementation specific data types in the model correspond to data types of a supported programming language. The translation is described in the language binding.
Dependencies:	–
Use Case:	Generation of C++ proxies and skeletons from the service interface description in order to implement the service interface of an application.
Supporting Material:	–

⌋*(RS_Main_00513)*

[RS_MANI_00025]{DRAFT} Definition and configuration of serialization ⌈

Description:	The Manifest specification shall allow to define and configure the serialization in the application design.
Rationale:	Serialization code is generated out of the service description and is linked with the application component object file to an application binary.
Dependencies:	–
Use Case:	SOME/IP serialization properties like length fields that are put in front of variable data.
Supporting Material:	–

⌋*(RS_Main_00280)*

[RS_MANI_00027]{DRAFT} Support for access to persistent data ⌈

Description:	The Manifest specification shall allow application software on the *AUTOSAR adaptive Platform* to access persistently stored data.
Rationale:	In many cases it is necessary to let a subset of the data managed by a given application survive a restart of either the application itself or the underlying operating system.
Dependencies:	–
Use Case:	Application software needs access to mileage counter.
Supporting Material:	–

⌋*(RS_Main_00440)*

[RS_MANI_00040]{DRAFT} Support for access to synchronized time ⌈

Description:	The Manifest specification shall allow application software on the *AUTOSAR adaptive Platform* to access synchronized time.
Rationale:	Application software shall be able to interact with synchronized time bases.
Dependencies:	–
Use Case:	Application software needs access to operation hours counter. Application software needs access to GPS time.
Supporting Material:	–

⌋*(RS_Main_00010, RS_Main_00420, RS_Main_00514, RS_Main_00510, RS_Main_-00491, RS_Main_00280)*

[RS_MANI_00032]{DRAFT} Support for platform health management ⌈

Description:	The Manifest specification shall provide application software on the *AUTOSAR adaptive Platform* means to interact with the platform health management (PHM).
Rationale:	In order to allow supervision and health management of application software the application software has to provide information about the supervision, checkpoints and health channels.
Dependencies:	–
Use Case:	Application software needs to report its progress such that the progress can be monitored by the platform health management.
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_MANI_00030]{DRAFT} Definition of optional elements in composite data structures ⌈

Description:	The Manifest specification shall support the configuration of optional elements for composite data structures used for communication.
Dependencies:	–
Use Case:	The sender of a composite data structure is unable to fill the entire data structure with meaningful values. The sender renders specific elements of the composite data as unavailable and the receiver is able to deal with the fact that the composite data structure is only partially available. An example for such a complex data structure could be an environment model where specific characteristic parts of the environment, e.g. a traffic sign may or may not exist depending on the location where the information in the environment model is captured.
Supporting Material:	–

⌋*(RS_Main_00300)*

[RS_MANI_00031]{DRAFT} Interaction with Crypto Software ⌈

Description:	The Manifest specification shall support the interaction of application software with crypto software on the adaptive platform.
Dependencies:	–
Use Case:	The application software owns data with a certain level of confidentiality (e.g. personally identifiable information, payment data) that needs to be cryptographically secured. For this purpose, the application software needs the ability to provide an API for the interaction with platform-level crypto software to achieve the encryption of sensitive information.
Supporting Material:	–

⌋ *(RS_Main_00445, RS_Main_00514, RS_Main_00510)*

[RS_MANI_00034]{DRAFT} Specification of intents ⌈

Description:	The Manifest specification shall support the definition of intents where application software can declare the intended usage of service elements.
Dependencies:	–
Use Case:	The application software uses a specific ServiceInterface for its interaction with the outside world. However, the application software intends to restrict its usage of the ServiceInterface to certain elements of the ServiceInterface. This information is helpful for the definition of access control and also contributes to the creation of a security model for the overall functionality.
Supporting Material:	–

⌋ *(RS_Main_00514)*

[RS_MANI_00068]{DRAFT} **Support for security event reporting interface definitions** ⌈

Description:	The Manifest specification shall support the configuration of service interfaces to report security events to the Intrusion Detection System Manager. These interfaces shall be usable by both application and platform software.
Rationale:	Detection and reporting of possible security events is a security feature with high market need due to future legal enforcements.
Dependencies:	–
Use Case:	Detection and reporting of a malicious network intruder.
Supporting Material:	–

⌋*(RS_Main_00514)*

3.1.1.2 Execution

[RS_MANI_00006]{DRAFT} **Support of application deployment** ⌈

Description:	The Manifest specification shall support a connection between application design and application deployment.
Rationale:	Definition of an executable that at runtime makes a POSIX process.
Dependencies:	–
Use Case:	The Execution Manager uses the manifest content to start up and configure each process individually.
Supporting Material:	–

⌋*(RS_Main_00320, RS_Main_00049, RS_Main_00050)*

[RS_MANI_00007]{DRAFT} **Configuration of application startup behavior** ⌈

Description:	The Manifest specification shall support the configuration of an application startup that is dependent on the current machine state.
Rationale:	Different machine states require a different set of (executed) applications.
Dependencies:	–
Use Case:	The Execution Manager uses the manifest content to start up and shutdown each process depending on the available machine state.
Supporting Material:	–

⌋*(RS_Main_00320, RS_Main_00049, RS_Main_00050)*

[RS_MANI_00050]{DRAFT} **Support of Deterministic Client** ⌈

Description:	The Manifest specification shall support the configuration of the so-called Deterministic Client.
Rationale:	The Deterministic Client supports the implementation of ASIL-C/D applications for e.g. highly automated driving.
Dependencies:	–
Use Case:	The Execution Manager is able to set up the Deterministic Client according to the configuration.
Supporting Material:	–

⌋*(RS_Main_00320, RS_Main_00049, RS_Main_00050)*

[RS_MANI_00060]{DRAFT} **Support of Identity and Access Management** ⌈

Description:	The Manifest specification shall support the configuration of the so-called Identity and Access Manager (IAM).
Rationale:	The Identity and Access Manager supports the security concept of the AUTOSAR adaptive platform by formalizing the access to specific APIs in terms of intended usage and actual grants.
Dependencies:	–
Use Case:	The IAM blocks access to specific APIs according to the configuration.
Supporting Material:	–

⌋*(RS_Main_00320, RS_Main_00514)*

[RS_MANI_00061]{DRAFT} **Support of Diagnostic Interfaces** ⌈

Description:	The Manifest specification shall support the configuration of diagnostic interfaces.
Rationale:	The interaction with the Diagnostic Manager can be quite complex if implemented directly on top of ara::com. By introducing dedicated PortInterfaces for diagnostic purposes the complexity can be hidden behind the implementation to a large extent.
Dependencies:	–
Use Case:	Application software interacts with the Diagnostic Manager.
Supporting Material:	–

⌋*(RS_Main_00260)*

3.2 Deployment Manifest Overview

The Manifest contains all necessary information about the deployment of an AUTOSAR application onto the adaptive AUTOSAR platform that is necessary for the configuration of the middleware.

3.2.1 Deployment Manifest Requirements

[RS_MANI_00011]{DRAFT} **Instantiation of provided and required services in an application** ⌈

Description:	The Manifest specification shall support the instantiation of provided and required services in an application by defining service instances and assigning these to the respective services.
Rationale:	Each time an application is instantiated on a machine for each provided service and for each required service a service instance shall be defined. Each time a single service is used in different roles within a single application different instances shall be defined.
Dependencies:	–
Use Case:	Camera Service may be instantiated as FrontCamera and RearCamera instances.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_00320, RS_Main_01005)*

[RS_MANI_00009]{DRAFT} **Service instance configuration on the network-level** ⌈

Description:	The Manifest specification shall support the configuration of service instances on the network-level and the assignment of these service instances to a machine.
Rationale:	For each service that is provided or requested on a machine a service instance shall be defined.
Dependencies:	–
Use Case:	Facilitate the definition of service instances without requiring the definition of a component model.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_00320, RS_Main_01005)*

[RS_MANI_00008]{DRAFT} **Service interface deployment to a transport layer mechanism** ⌈

Description:	The Manifest specification shall support the deployment of a service interface to one or several transport layer mechanisms.
Rationale:	Different service interfaces support different transport mechanisms because of deployment decisions.
Dependencies:	–
Use Case:	Offering of a service interface via SOME/IP and/or user-defined transport layer mechanisms.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_00320, RS_Main_01005)*

[RS_MANI_00014]{DRAFT} **User defined transport layer mechanisms** ⌈

Description:	The Manifest specification shall support the usage of transport layer mechanisms that are not standardized by AUTOSAR.
Rationale:	Adaptive AUTOSAR applications shall be able to communicate with non-AUTOSAR applications located on the same or on remote ECUs using transport layers that are not standardized by AUTOSAR.
Dependencies:	–
Use Case:	Usage of shared memory based IPC transport mechanism not standardized by AUTOSAR.
Supporting Material:	–

⌋*(RS_Main_01001, RS_Main_01005)*

[RS_MANI_00024]{DRAFT} SOME/IP transport layer mechanisms ⌈

Description:	The Manifest specification shall support the usage of the SOME/IP transport layer mechanism.
Rationale:	Adaptive AUTOSAR applications shall be able to communicate with other adaptive (or classic) AUTOSAR applications located on remote ECUs using SOME/IP transport layer mechanisms.
Dependencies:	–
Use Case:	Standardized service-oriented communication.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_01005)*

[RS_MANI_00038]{DRAFT} DDS transport layer mechanisms ⌈

Description:	The Manifest specification shall support the usage of the DDS transport layer mechanism.
Rationale:	Adaptive AUTOSAR applications shall be able to communicate with other adaptive AUTOSAR applications located on remote ECUs using DDS transport layer mechanisms.
Dependencies:	–
Use Case:	Standardized DDS communication.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_01005)*

[RS_MANI_00019]{DRAFT} Service discovery message exchange configuration ⌈

Description:	The Manifest specification shall provide means to configure the service discovery message exchange for the different supported transport layer mechanisms.
Rationale:	Service discovery messages are exchanged in the System with multicast addressing to a specific configured IP multicast address at a specific configured port number.
Dependencies:	–
Use Case:	SOME/IP service discovery configuration.
Supporting Material:	–

⌋*(RS_Main_00280, RS_Main_01005)*

[RS_MANI_00028]{DRAFT} Configuration of Safety protection ⌈

Description:	The Manifest specification shall provide means to configure the Safety protection of data specified in a Service Interface.
Rationale:	Adaptive AUTOSAR applications shall be able to communicate with other adaptive (or classic) AUTOSAR applications over a communication path in a safe way.
Dependencies:	–
Use Case:	Sending and receiving of End-to-End protected Events over SOME/IP.
Supporting Material:	–

⌋*(RS_Main_00010)*

[RS_MANI_00036]{DRAFT} Configuration of security protection ⌈

Description:	The Manifest specification shall provide means to configure the security protection of data specified in a Service Interface.
Rationale:	Adaptive AUTOSAR applications shall be able to communicate with other adaptive (or classic) AUTOSAR applications over a secured communication channel.
Dependencies:	–
Use Case:	Sending and Receiving of Events over SOME/IP in an encrypted or authenticated way.
Supporting Material:	–

⌋*(RS_Main_00514, RS_Main_00510, RS_Main_01008)*

[RS_MANI_00037]{DRAFT} Configuration of logging and tracing ⌈

Description:	The Manifest specification shall support applications that provide logging information onto the communication bus, the console, or to the file system with necessary configuration settings.
Rationale:	Applications shall have the possibility to send log or trace messages.
Dependencies:	–
Use Case:	Application Identification from the Manifest is added as additional meta-information into the logging message, so that clients are able to relate, sort or filter the received logging messages.
Supporting Material:	–

⌋*(RS_Main_00491)*

[RS_MANI_00063]{DRAFT} The Manifest specification shall support the translation between signal-based and service-oriented communication ⌈

Description:	The Manifest specification shall support the translation between signal-based and service-oriented communication.
Rationale:	Adaptive Platform restricts communication to Service-oriented communication, the rest of the vehicle however still uses Signal-based communication means - therefore a translation of these two approaches has to be performed.
Dependencies:	As an alternative the translation may be implemented on a Classic platform gateway Ecu.
Use Case:	Data which is produced on a Can ECU is needed at an Adaptive machine in a safe and secure manner. The translation may be implemented on an Adaptive platform Machine.
Supporting Material:	–

⌋*(RS_Main_00652)*

[RS_MANI_00064]{DRAFT} Service contract version for a service interface ⌈

Description:	The Manifest specification shall support the definition of a service contract version number for a service interface to allow multiple versions of the same service to be operational at the same time.
Rationale:	Service Interface contract versioning indicates the service development over time (meaning changes of the Service interface and/or behavior over time).
Dependencies:	–
Use Case:	A Service Interface is extended with an additional method. The version change that indicates a backward compatible change eliminates the need of rebuilding all clients that are using this service.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_01002, RS_Main_00140)*

[RS_MANI_00065]{DRAFT} **Service contract versioning for all Transport Deployment Protocols** ⌈

Description:	The Manifest specification shall support the service contract versioning on the level of transport layer deployment protocols.
Rationale:	To delegate the service contract information from design phase to deployment phase the transport deployment protocols need to support versioning.
Dependencies:	–
Use Case:	Support of service contract versioning of SOME/IP
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_01002, RS_Main_00140)*

[RS_MANI_00066]{DRAFT} **Service Versioning Blacklist** ⌈

Description:	The Manifest specification shall provide mechanisms to influence the service search and to block connections to particular service providers.
Rationale:	This feature enables last minute changes to block provider versions.
Dependencies:	–
Use Case:	Last minute changes/patches on OEM site.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_01002, RS_Main_00140)*

[RS_MANI_00067]{DRAFT} **Raw data stream deployment** ⌈

Description:	The Manifest specification shall support the deployment of raw data streams defined on the application level to a transport technology.
Rationale:	Adaptive AUTOSAR Applications sends and receives byte streams over Ethernet to and from sensors in an ADAS System.
Dependencies:	–
Use Case:	Write raw data to a TCP/IP Socket.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_00280)*

[RS_MANI_00069]{DRAFT} **Support for deployment definition of the Intrusion Detection System Manager** ⌈

Description:	The Manifest specification shall support the definition of deployment of the Intrusion Detection System Manager together with its network configuration to enable detection and reporting of security events.

▽

$\triangle$

Rationale:	Detection and reporting of possible security events is a security feature with high market need due to future legal enforcements.
Dependencies:	–
Use Case:	Detection and reporting of a malicious network intruder.
Supporting Material:	–

⌋*(RS_Main_00514)*

3.3 Machine Manifest Overview

The Manifest contains all necessary information about the configuration of a machine.

3.3.1 Machine Manifest Requirements

[RS_MANI_00018]{DRAFT} **Network connections of the machine** ⌈

Description:	The Manifest specification shall provide means to configure the network connections of a machine.
Rationale:	Configure the network connections for in-vehicle usage.
Dependencies:	–
Use Case:	IPv4 and IPv6 configuration.
Supporting Material:	–

⌋*(RS_Main_00230)*

[RS_MANI_00020]{DRAFT} Hardware resources of the machine ⌈

Description:	The Manifest specification shall provide means to describe the hardware resources of a machine.
Rationale:	The knowledge of existing hardware resources is essential if new applications are allocated to the machine.
Dependencies:	–
Use Case:	Description of available RAM. Description of available CPU power.
Supporting Material:	–

⌋*(RS_Main_00503, RS_Main_00435)*

[RS_MANI_00021]{DRAFT} Description of machine states ⌈

Description:	The Manifest specification shall provide means to describe available machine states.
Rationale:	Provides mechanism to define machine states for various operational conditions.
Dependencies:	–
Use Case:	Application is running only in a given machine state.
Supporting Material:	–

⌋*(RS_Main_00002, RS_Main_00460)*

[RS_MANI_00022]{DRAFT} Adaptive Platform configuration ⌈

Description:	The Manifest specification shall provide means to configure the adaptive platform on a specific machine.
Rationale:	Decision which Adaptive Modules and their respective configuration are deployed to a specific machine.
Dependencies:	–
Use Case:	For one dedicated Platform Instance the following Adaptive Modules are deployed: ExecutionManagment, CommunicationManagement and HealthManagement, but not Diagnostics.
Supporting Material:	–

⌋*(RS_Main_00002)*

[RS_MANI_00023]{DRAFT} Adaptive Module configuration ⌈

Description:	The Manifest specification shall provide means to configure the instance of a module on a specific machine.
Rationale:	Configuration of Watchdog and OS.
Dependencies:	–
Use Case:	Configure the effective access role of OS.
Supporting Material:	–

⌋*(RS_Main_00002)*

[RS_MANI_00041]{DRAFT} Configuration of function groups ⌈

Description:	The Manifest specification shall provide means to configure function groups that are available on a machine.
Rationale:	Function group states individually control starting and terminating of functionally coherent applications.
Dependencies:	–
Use Case:	Usage of function group to start and stop all diagnostic relevant applications.
Supporting Material:	–

⌋*(RS_Main_00460)*

3.4 System Design Overview

The AUTOSAR model description supports the software component system design for the entire vehicle with the possibility to describe all Software Components of both Autosar Platforms that will be used in a System.

3.4.1 System Design Requirements

[RS_MANI_00026]{DRAFT} **Software Component System Design** ⌈

Description:	The Manifest specification shall support the system (vehicle) software component design with the possibility to describe software components of different AUTOSAR platforms (Classic Platform and Adaptive Platform) in a common model.
Rationale:	A typical vehicle will most likely be equipped with ECUs developed on the AUTOSAR classic platform and ECUs developed on the AUTOSAR adaptive platform. Therefore a software component system design modeling for the entire vehicle shall be supported by AUTOSAR.
Dependencies:	–
Use Case:	Modeling of a Classic Platform Software Component that communicates with an Adaptive Platform Software Component.
Supporting Material:	–

⌋*(RS_Main_00161, RS_Main_00310)*

[RS_MANI_00062]{DRAFT} **Support for Partial Networking** ⌈

Description:	The Manifest specification shall support the definition and configuration of partial networks.
Rationale:	Power saving during vehicle operation time with the partial networking mechanism.
Dependencies:	–
Use Case:	The power consumption is reduced by e.g shutting down of seat control functions.
Supporting Material:	–

⌋*(RS_Main_00420)*

[RS_MANI_00029]{DRAFT} **Mapping description between Signal-based communication and Service-Oriented communication** ⌈

Description:	The Manifest specification shall support the communication description between Classic Platform Software Components that are available on an ECU that only supports the Signal-based communication and Adaptive Applications that only support the Service-Oriented communication.
Rationale:	An application shall be able to access received Signals and map their content to service oriented communication that is used on the Adaptive Platform according to a provided mapping description.
Dependencies:	–
Use Case:	Integration of an application on Adaptive platform that is communicating with a Software Component on Classic Platform that is available on an ECU that is only connected to a CAN bus and sends and receives Signals.
Supporting Material:	–

⌋*(RS_Main_00161)*

4 Requirements Tracing

The following table references the requirements specified in [4] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_Main_00001]	AUTOSAR shall provide a software platform for embedded real-time systems	[RS_MANI_00032]
[RS_Main_00002]	AUTOSAR shall provide a software platform for high performance computing platforms	[RS_MANI_00015] [RS_MANI_00016] [RS_MANI_00021] [RS_MANI_00022] [RS_MANI_00023]
[RS_Main_00010]	AUTOSAR shall support the development of safety related systems	[RS_MANI_00028] [RS_MANI_00032] [RS_MANI_00040]
[RS_Main_00011]	AUTOSAR shall support the development of reliable systems	[RS_MANI_00032]
[RS_Main_00049]	AUTOSAR shall provide an Execution Management for running multiple applications	[RS_MANI_00006] [RS_MANI_00007] [RS_MANI_00050]
[RS_Main_00050]	AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows	[RS_MANI_00006] [RS_MANI_00007] [RS_MANI_00050]
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_MANI_00003] [RS_MANI_00064] [RS_MANI_00065] [RS_MANI_00066] [RS_MANI_00067]
[RS_Main_00080]	AUTOSAR shall provide means to describe a component model for Application Software	[RS_MANI_00002] [RS_MANI_00003] [RS_MANI_00004] [RS_MANI_00005]

▽

△

Requirement	Description	Satisfied by
[RS_Main_00140]	AUTOSAR shall provide network independent communication mechanisms for applications	[RS_MANI_00002] [RS_MANI_00064] [RS_MANI_00065] [RS_MANI_00066]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_MANI_00001] [RS_MANI_00002] [RS_MANI_00003] [RS_MANI_00004] [RS_MANI_00035]
[RS_Main_00161]	AUTOSAR shall provide a unified way to describe software systems deployed to Adaptive and / or Classic platforms	[RS_MANI_00026] [RS_MANI_00029]
[RS_Main_00190]	AUTOSAR shall support standardized interoperability with non-AUTOSAR software	[RS_MANI_00003] [RS_MANI_00016]
[RS_Main_00230]	AUTOSAR shall support network topologies including gateways	[RS_MANI_00018]
[RS_Main_00260]	AUTOSAR shall provide diagnostics means during runtime, for production and services purposes	[RS_MANI_00005] [RS_MANI_00061]
[RS_Main_00280]	AUTOSAR shall support standardized automotive communication protocols	[RS_MANI_00008] [RS_MANI_00009] [RS_MANI_00011] [RS_MANI_00019] [RS_MANI_00024] [RS_MANI_00025] [RS_MANI_00038] [RS_MANI_00040] [RS_MANI_00067]
[RS_Main_00300]	AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups	[RS_MANI_00004] [RS_MANI_00030]
[RS_Main_00310]	AUTOSAR shall support hierarchical Application Software design methods	[RS_MANI_00004] [RS_MANI_00026]
[RS_Main_00320]	AUTOSAR shall provide formats to specify system development	[RS_MANI_00006] [RS_MANI_00007] [RS_MANI_00008] [RS_MANI_00009] [RS_MANI_00011] [RS_MANI_00017] [RS_MANI_00050] [RS_MANI_00060]
[RS_Main_00340]	AUTOSAR shall support the continuous timing requirement analysis	[RS_MANI_00032]
[RS_Main_00420]	AUTOSAR shall use established software standards and consolidate de-facto standards for basic software functionality	[RS_MANI_00040] [RS_MANI_00062]
[RS_Main_00435]	AUTOSAR shall support automotive microcontrollers	[RS_MANI_00020]
[RS_Main_00440]	AUTOSAR shall standardize access to non-volatile memory	[RS_MANI_00027]
[RS_Main_00445]	AUTOSAR shall standardize access to crypto-specific HW and SW	[RS_MANI_00031]
[RS_Main_00460]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[RS_MANI_00021] [RS_MANI_00041]
[RS_Main_00491]	AUTOSAR shall provide means for logging	[RS_MANI_00037] [RS_MANI_00040]
[RS_Main_00503]	AUTOSAR shall support change of communication and application software at runtime.	[RS_MANI_00015] [RS_MANI_00020]
[RS_Main_00510]	AUTOSAR shall support secure onboard communication	[RS_MANI_00031] [RS_MANI_00036] [RS_MANI_00040]

▽

△

Requirement	Description	Satisfied by
[RS_Main_00513]	AUTOSAR shall support language bindings for different programming languages	[RS_MANI_00039]
[RS_Main_00514]	AUTOSAR shall support the development of secure systems	[RS_MANI_00031] [RS_MANI_00034] [RS_MANI_00036] [RS_MANI_00040] [RS_MANI_00060] [RS_MANI_00068] [RS_MANI_00069]
[RS_Main_00652]	AUTOSAR shall support the translation between signal-based and service-oriented communication	[RS_MANI_00063]
[RS_Main_01001]	AUTOSAR shall support intra ECU communication	[RS_MANI_00014]
[RS_Main_01002]	AUTOSAR shall support service-oriented communication	[RS_MANI_00064] [RS_MANI_00065] [RS_MANI_00066]
[RS_Main_01005]	AUTOSAR shall establish communication paths dynamically	[RS_MANI_00008] [RS_MANI_00009] [RS_MANI_00011] [RS_MANI_00014] [RS_MANI_00019] [RS_MANI_00024] [RS_MANI_00038]
[RS_Main_01008]	AUTOSAR shall provide secure communication with off-board entities	[RS_MANI_00036]

Table 4.1: RequirementsTracing

5 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Specification of Manifest
AUTOSAR_TPS_ManifestSpecification

[3] Glossary
AUTOSAR_TR_Glossary

[4] Main Requirements
AUTOSAR_RS_Main

6 Change History of this Document

6.1 Change History of this Document according to the original version of the Document

Number	Heading
[RS_MANI_00001]	Adaptive AUTOSAR Application
[RS_MANI_00002]	Declaration of provided and required services in an application
[RS_MANI_00003]	Specification of service interfaces
[RS_MANI_00004]	Support of application design
[RS_MANI_00005]	Configuration of diagnostic capabilities of an application
[RS_MANI_00006]	Support of application deployment
[RS_MANI_00007]	Configuration of application startup behavior
[RS_MANI_00008]	Service interface deployment to a transport layer mechanism
[RS_MANI_00009]	Service instance configuration on the network-level
[RS_MANI_00011]	Instantiation of provided and required services in an application
[RS_MANI_00014]	User defined transport layer mechanisms
[RS_MANI_00015]	Definition of the nature of a manifest
[RS_MANI_00016]	Usage of data types specifically on the AUTOSAR adaptive platform
[RS_MANI_00017]	Specification of the mapping of Service Interfaces
[RS_MANI_00018]	Network connections of the machine
[RS_MANI_00019]	Service discovery message exchange configuration
[RS_MANI_00020]	Hardware resources of the machine
[RS_MANI_00021]	Description of machine states
[RS_MANI_00022]	Adaptive Platform configuration
[RS_MANI_00023]	Adaptive Module configuration
[RS_MANI_00024]	SOME/IP transport layer mechanisms
[RS_MANI_00025]	Definition and configuration of serialization

Table 6.1: Added Requirements in original Version

6.2 Change History of this document according to AUTOSAR Release 17-10

6.2.1 Added Traceables in 17-10

Number	Heading
[RS_MANI_00026]	Software Component System Design
[RS_MANI_00027]	Support for access to persistent data
[RS_MANI_00028]	Configuration of Safety protection
[RS_MANI_00029]	Mapping description between Signal-based communication and Service-Oriented communication
[RS_MANI_00030]	Definition of optional elements in composite data structures
[RS_MANI_00031]	Interaction with Crypto Software
[RS_MANI_00032]	Support for platform health management supervision
[RS_MANI_00033]	Interaction with web services based on the REST pattern
[RS_MANI_00034]	Specification of capabilities
[RS_MANI_00035]	Definition of an uploadable software package
[RS_MANI_00036]	Configuration of security protection

Table 6.2: Added Traceables in 17-10

6.2.2 Changed Traceables in 17-10

Number	Heading
[RS_MANI_00001]	Adaptive AUTOSAR Application

Table 6.3: Changed Traceables in 17-10

6.2.3 Deleted Traceables in 17-10

6.3 Change History of this document according to AUTOSAR Release 18-03

6.3.1 Added Traceables in 18-03

Number	Heading
[RS_MANI_00037]	Configuration of logging and tracing
[RS_MANI_00038]	DDS transport layer mechanisms
[RS_MANI_00039]	Usage of implementation specific data types
[RS_MANI_00040]	Support for access to synchronized time
[RS_MANI_00041]	Configuration of function groups

Table 6.4: Added Traceables in 18-03

6.3.2 Changed Traceables in 18-03

Number	Heading
[RS_MANI_00014]	User defined transport layer mechanisms

Table 6.5: Changed Traceables in 18-03

6.3.3 Deleted Traceables in 18-03

6.4 Change History of this document according to AUTOSAR Release 18-10

6.4.1 Added Traceables in 18-10

Number	Heading
[RS_MANI_00050]	Support of Deterministic Client

Table 6.6: Added Traceables in 18-10

6.4.2 Changed Traceables in 18-10

6.4.3 Deleted Traceables in 18-10

6.5 Change History of this document according to AUTOSAR Release 19-03

6.5.1 Added Traceables in 19-03

Number	Heading
[RS_MANI_00060]	Support of Identity and Access Management
[RS_MANI_00061]	Support of Diagnostic Interfaces
[RS_MANI_00062]	Support for Partial Networking

Table 6.7: Added Traceables in 19-03

6.5.2 Changed Traceables in 19-03

6.5.3 Deleted Traceables in 19-03

6.6 Change History of this document according to AUTOSAR Release 19-11

6.6.1 Added Traceables in 19-11

Number	Heading
[RS_MANI_00063]	The Manifest specification shall support the translation between signal-based and service-oriented communication
[RS_MANI_00064]	Service contract version for a service interface
[RS_MANI_00065]	Service contract versioning for all Transport Deployment Protocols
[RS_MANI_00066]	Service Versioning Blacklist

Table 6.8: Added Traceables in 19-11

6.6.2 Changed Traceables in 19-11

6.6.3 Deleted Traceables in 19-11

6.7 Change History of this document according to AUTOSAR Release R20-11

6.7.1 Added Traceables in R20-11

Number	Heading
[RS_MANI_00067]	Raw data stream deployment
[RS_MANI_00068]	Support for security event reporting interface definitions
[RS_MANI_00069]	Support for deployment definition of the Intrusion Detection System Manager

Table 6.9: Added Traceables in R20-11

6.7.2 Changed Traceables in R20-11

Number	Heading
[RS_MANI_00034]	Specification of intents

Table 6.10: Changed Traceables in R20-11

6.7.3 Deleted Traceables in R20-11

6.8 Change History of this document according to AUTOSAR Release R21-11

6.8.1 Added Traceables in R21-11

6.8.2 Changed Traceables in R21-11

6.8.3 Deleted Traceables in R21-11

Number	Heading
[RS_MANI_00033]	Interaction with web services based on the REST pattern

Table 6.11: Deleted Traceables in R21-11