Document Title	Requirements on AUTOSAR Features
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	294
Document Classification	Auxiliary
Document Version	1.2.1
Document Status	Final
Part of Release	4.1
Revision	2

Document Change History

Date	Version	Changed by	Change Description
31.10.2013	1.2.1	AUTOSAR Release Management	Name of document changed
06.03.2013	1.2.0	AUTOSAR Administration	Complete rework of document, requirements scheme updated
20.12.2011	1.1.0	AUTOSAR Administration	Corrected wrong usage of term “module short name”
30.11.2009	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Scope of Document

This document describes all features of AUTOSAR including Basic Software (BSW) and the RTE.

The features are grouped according to the architecture of AUTOSAR Basic Software and RTE.

2 How to read this document

Each requirement has its unique identifier starting with the prefix “RS_BRF_” (for “Basic AutosaR Features”). For any review annotations, remarks or questions please refer to this unique ID rather than chapter or page numbers!

2.1 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in TPS_STDT_00078 (see **[TPS_STDT]**).

In requirements, the following specific semantics shall be used (based on the Internet Engineering Task Force IETF).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as:

- SHALL: This word means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
- MUST: This word means that the definition is an absolute requirement of the specification due to legal issues.
- MUST NOT: This phrase means that the definition is an absolute prohibition of the specification due to legal constraints.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate

with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Acronyms and Abbreviations

All acronyms and abbreviations used throughout this document are included in the official AUTOSAR glossary **[GLOSSARY]**. For respective explanation please see there.

3 Requirements Tracing

The following table references the requirements specified in **[RS_MAIN]** and links to the fulfilments of these.

Requirement	Description	Satisfied by
-	-	RS_BRF_01024
-	-	RS_BRF_01032
-	-	RS_BRF_01120
-	-	RS_BRF_01136
-	-	RS_BRF_01144
-	-	RS_BRF_01152
RS_Main_00010	AUTOSAR shall provide a software platform to support the development of safety related systems.	RS_BRF_00057, RS_BRF_00110, RS_BRF_00113, RS_BRF_00129, RS_BRF_00131, RS_BRF_00241, RS_BRF_01168, RS_BRF_01232, RS_BRF_01240, RS_BRF_01248, RS_BRF_02040, RS_BRF_02048, RS_BRF_02056, RS_BRF_02064, RS_BRF_02096, RS_BRF_02104
RS_Main_00011	AUTOSAR shall support the development of reliable systems	RS_BRF_00113, RS_BRF_00129, RS_BRF_01440, RS_BRF_01464, RS_BRF_01600, RS_BRF_01608, RS_BRF_01840, RS_BRF_01848, RS_BRF_01936, RS_BRF_01944, RS_BRF_02000, RS_BRF_02168, RS_BRF_02176, RS_BRF_02216, RS_BRF_02224
RS_Main_00030	AUTOSAR shall support development processes for safety related systems	RS_BRF_NA_1
RS_Main_00060	AUTOSAR shall provide a standardized software interface for communication between Software Components	RS_BRF_01176, RS_BRF_01280, RS_BRF_01288, RS_BRF_01296, RS_BRF_01304, RS_BRF_01312, RS_BRF_01320, RS_BRF_01328, RS_BRF_01336, RS_BRF_01344, RS_BRF_01352, RS_BRF_01360, RS_BRF_01368, RS_BRF_01376, RS_BRF_01384, RS_BRF_01392, RS_BRF_01400
RS_Main_00080	AUTOSAR shall provide means to describe a component model for application software	RS_BRF_NA_3
RS_Main_00100	AUTOSAR shall provide standardized basic software	RS_BRF_00057, RS_BRF_01000, RS_BRF_01040, RS_BRF_01048, RS_BRF_01056, RS_BRF_01072, RS_BRF_01160, RS_BRF_01168, RS_BRF_01200, RS_BRF_01208, RS_BRF_01216, RS_BRF_01224, RS_BRF_01232, RS_BRF_01240, RS_BRF_01248, RS_BRF_01256, RS_BRF_01264, RS_BRF_01272, RS_BRF_02040
RS_Main_00120	AUTOSAR shall provide means to assure interoperability of AUTOSAR implementations (ICC1 level) on application level (RTE) and bus level.	RS_BRF_NA_2
RS_Main_00130	AUTOSAR shall provide an abstraction from hardware	RS_BRF_01008, RS_BRF_01792, RS_BRF_01800, RS_BRF_01808, RS_BRF_01856, RS_BRF_01864, RS_BRF_01872, RS_BRF_01880, RS_BRF_01888,

		RS_BRF_01896, RS_BRF_01904, RS_BRF_01912, RS_BRF_01920, RS_BRF_01928, RS_BRF_01936, RS_BRF_01944, RS_BRF_01968, RS_BRF_01976, RS_BRF_01984, RS_BRF_01992
RS_Main_00140	AUTOSAR shall provide network independent communication mechanisms for applications	RS_BRF_01288
RS_Main_00150	AUTOSAR shall support the reallocation of Software Components	RS_BRF_01416, RS_BRF_01432, RS_BRF_01832, RS_BRF_01960
RS_Main_00160	AUTOSAR shall provide means to describe interfaces of the entire system.	RS_BRF_NA_3
RS_Main_00170	AUTOSAR shall provide secure access to ECU	RS_BRF_01456, RS_BRF_02008, RS_BRF_02016, RS_BRF_02024, RS_BRF_02032, RS_BRF_02136, RS_BRF_02208
RS_Main_00180	AUTOSAR shall provide mechanisms to protect intellectual property in a shared development process	RS_BRF_NA_3
RS_Main_00190	AUTOSAR shall support interoperability with non-AUTOSAR software on the same ECU	RS_BRF_02280
RS_Main_00200	AUTOSAR specifications shall allow resource efficient implementations	RS_BRF_01088, RS_BRF_01128, RS_BRF_01184
RS_Main_00210	AUTOSAR shall support interoperability with non-AUTOSAR ECUs in a network	RS_BRF_02288
RS_Main_00220	The functional interfaces of AUTOSAR shall be specified in C90	RS_BRF_01056, RS_BRF_02080
RS_Main_00230	AUTOSAR shall support network topologies including gateways	RS_BRF_01576, RS_BRF_01584
RS_Main_00250	AUTOSAR process shall provide a predefinition of typical roles and activities in work-share model	RS_BRF_NA_3
RS_Main_00251	AUTOSAR process shall support roles and rights in a work-share model	RS_BRF_NA_3
RS_Main_00260	AUTOSAR shall provide diagnostics means during runtime, for production and services purposes	RS_BRF_01112, RS_BRF_01440, RS_BRF_01720, RS_BRF_01736, RS_BRF_01760, RS_BRF_02144, RS_BRF_02152, RS_BRF_02160, RS_BRF_02168, RS_BRF_02176, RS_BRF_02184, RS_BRF_02192, RS_BRF_02200, RS_BRF_02208, RS_BRF_02216
RS_Main_00270	AUTOSAR shall provide mitigation strategies towards new releases	RS_BRF_NA_2

RS_Main_00280	AUTOSAR shall provide a communication interface to the infotainment systems	RS_BRF_01784
RS_Main_00290	AUTOSAR shall support the verification of its specifications	RS_BRF_NA_1
RS_Main_00300	AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups	RS_BRF_NA_3
RS_Main_00310	AUTOSAR shall support hierarchical design methods	RS_BRF_NA_3
RS_Main_00320	AUTOSAR shall provide formats to specify all aspects necessary to integrate a Software Component on an ECU	RS_BRF_NA_3
RS_Main_00330	AUTOSAR shall support the principle of information hiding	RS_BRF_01016
RS_Main_00340	AUTOSAR shall support the observance of timing requirements	RS_BRF_NA_3
RS_Main_00350	AUTOSAR specifications shall be analyzable and support according methods to demonstrate the achievement of safety related properties.	RS_BRF_NA_1
RS_Main_00360	AUTOSAR shall support management of vehicle diversity	RS_BRF_NA_3
RS_Main_00400	AUTOSAR shall provide a layered software architecture	RS_BRF_01000, RS_BRF_01008, RS_BRF_01064, RS_BRF_01192, RS_BRF_01408, RS_BRF_01800
RS_Main_00410	AUTOSAR shall provide specifications for routines commonly used by Software Components to support sharing and optimization	RS_BRF_02072, RS_BRF_02080, RS_BRF_02088, RS_BRF_02096, RS_BRF_02104, RS_BRF_02112, RS_BRF_02120, RS_BRF_02128
RS_Main_00420	AUTOSAR shall consolidate existing basic software functionality of the automotive domain	RS_BRF_01184, RS_BRF_01200, RS_BRF_01680, RS_BRF_01688, RS_BRF_01696, RS_BRF_02144, RS_BRF_02264
RS_Main_00430	AUTOSAR shall support automotive communication systems	RS_BRF_01424, RS_BRF_01544, RS_BRF_01552, RS_BRF_01560, RS_BRF_01568, RS_BRF_01576, RS_BRF_01584, RS_BRF_01592, RS_BRF_01600, RS_BRF_01608, RS_BRF_01616, RS_BRF_01624, RS_BRF_01632, RS_BRF_01640, RS_BRF_01648, RS_BRF_01656, RS_BRF_01664, RS_BRF_01672, RS_BRF_01680, RS_BRF_01688, RS_BRF_01696, RS_BRF_01704, RS_BRF_01712, RS_BRF_01720, RS_BRF_01728, RS_BRF_01736, RS_BRF_01744, RS_BRF_01752, RS_BRF_01760, RS_BRF_01768, RS_BRF_01776, RS_BRF_01784

RS_Main_00435	AUTOSAR shall support automotive microcontrollers	RS_BRF_00057, RS_BRF_00206, RS_BRF_01080, RS_BRF_01168, RS_BRF_01432, RS_BRF_01856, RS_BRF_01864, RS_BRF_01872, RS_BRF_01880, RS_BRF_01888, RS_BRF_01896, RS_BRF_01904, RS_BRF_01912, RS_BRF_01920, RS_BRF_01928, RS_BRF_01936, RS_BRF_01944
RS_Main_00440	AUTOSAR shall standardize access to non-volatile memory	RS_BRF_01416, RS_BRF_01800, RS_BRF_01808, RS_BRF_01816, RS_BRF_01824, RS_BRF_01832, RS_BRF_01840, RS_BRF_01848, RS_BRF_01928
RS_Main_00450	AUTOSAR shall standardize access to general purpose I/O	RS_BRF_01080, RS_BRF_01864, RS_BRF_01872, RS_BRF_01880, RS_BRF_01888, RS_BRF_01896, RS_BRF_01952, RS_BRF_02000
RS_Main_00460	AUTOSAR shall standardize methods to organize mode management on SWC, ECU and system level	RS_BRF_01088, RS_BRF_01096, RS_BRF_01104, RS_BRF_01184, RS_BRF_01448, RS_BRF_01472, RS_BRF_01480, RS_BRF_01488, RS_BRF_01496, RS_BRF_01504, RS_BRF_01512, RS_BRF_01520, RS_BRF_01528, RS_BRF_01536, RS_BRF_01664, RS_BRF_01672, RS_BRF_01680, RS_BRF_01688, RS_BRF_01696, RS_BRF_01952, RS_BRF_02216
RS_Main_00480	AUTOSAR shall support the test of implementations	RS_BRF_02224, RS_BRF_02232, RS_BRF_02240, RS_BRF_02248, RS_BRF_02256, RS_BRF_02264, RS_BRF_02272
RS_Main_00490	AUTOSAR processes shall be compliant to ISO26262	RS_BRF_NA_1

4 Requirements Specification

4.1 System and Architecture

4.1.1 [RS_BRF_01000] AUTOSAR architecture shall organize the BSW in a hardware independent and a hardware dependent layer

⌈

Type:	Valid
Description:	AUTOSAR architecture (AUTOSAR Layered Software Architecture) shall organize the BSW in a hardware independent layer and a hardware dependent layer which base on each other
Rationale:	Make as many modules as possible portable between processor architectures. Additionally, establish a clear dependency between modules. This also encapsulates internal behavior of the hardware dependent layer from upper layers
Use Case:	Reuse implementation of shadow buffer strategies for non-volatile RAM management on all processor architectures
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00400, RS_Main_00100)

4.1.2 [RS_BRF_01008] AUTOSAR shall organize the hardware dependent layer in a microcontroller independent and a microcontroller dependent layer

⌈

Type:	Valid
Description:	AUTOSAR shall organize the hardware dependent layer in a microcontroller independent and a microcontroller dependent layer which base on each other
Rationale:	By moving all microcontroller dependencies to a separate layer, more modules are portable between processor architectures as long as the external peripheral devices are the same. As a result, the microcontroller dependent layer can be kept as small as possible. This also encapsulates internal behavior of the microcontroller dependent layer from upper layers
Use Case:	Keep strategies how to best look-up CAN identifiers out of the microcontroller dependent layer and thus be able to re-use implementations on other microcontrollers
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130, RS_Main_00400)

4.1.1 [RS_BRF_01016] AUTOSAR shall provide a modular design inside software layers

⌈

Type:	Valid

Description:	In each layer, AUTOSAR shall separate the complete functionality into disjunct parts which are implemented in modules and separately specified (loose coupling, high coherence). Specification of a BSW-Module defines all upper and lower external interfaces and module behavior
Rationale:	A modular design inside software layers with defined interfaces - encapsulates internal behavior - reduces complexity - increases maintainability - improves portability and - eases testability
Use Case:	Provide separate modules for FlexRay and CAN
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00330)

4.1.2 [RS_BRF_01024] AUTOSAR shall provide naming rules for public symbols

⌈

Type:	Valid
Description:	AUTOSAR shall provide naming rules that apply for all publicly visible symbols of Basic Software Modules, RTE and libraries. This includes especially naming rules for function names, types and constants
Rationale:	Avoid name clashes during system integration. Provide a consistent uniform interface to the user
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋()

4.1.3 [RS_BRF_01032] AUTOSAR modules shall provide meta data information

⌈

Type:	Valid
Description:	AUTOSAR modules shall provide meta data information to identify a module on source and object level. This includes e. g. version information, supplier information …
Rationale:	Allow the integrator to supervise and identify the set of Basic Software Modules during build time and run-time of the system
Use Case:	Reject compilation of modules from incompatible AUTOSAR versions or configuration builds
Dependencies:	--
Supporting Material:	--

⌋()

4.1.4 [RS_BRF_01040] AUTOSAR shall allow multiple instantiation of Basic Software Modules where appropriate

⌈

Type:	Valid
Description:	AUTOSAR shall allow multiple instantiation of Basic Software Modules where appropriate
Rationale:	Support directly connected hardware of same type but with different access methods
Use Case:	Systems with full and basic CAN
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.1.5 [RS_BRF_01048] AUTOSAR module design shall support modules to cooperate in a multitasking environment

⌈

Type:	Valid
Description:	AUTOSAR modules shall be designed such that they consider parallel activity of other AUTOSAR modules and cooperate in a multitasking environment
Rationale:	An AUTOSAR module have to consider that other Basic Software Modules need ECU resource like computation power, interrupt responsiveness, etc. in parallel to fulfill timing restrictions of the overall system
Use Case:	Avoid busy waits within interrupt handlers
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.1.6 [RS_BRF_01056] AUTOSAR BSW modules shall provide standardized interfaces

⌈

Type:	Valid
Description:	AUTOSAR BSW modules shall specify standardized application programming interfaces based on the C language
Rationale:	Allow upper layer modules, services or integrator code to access standardized functionality of a BSW module via C90 functions
Use Case:	All standardized interfaces of Basic Software Modules
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00220, RS_Main_00100)

4.1.7 [RS_BRF_01064] AUTOSAR BSW shall provide callback functions in order to access upper layer modules

⌈

Type:	Valid
Description:	AUTOSAR BSW shall provide callback functions in order to access upper layer modules
Rationale:	In order to activate functionality in an upper layer module a lower layer module has to specify callback functions that have to be implemented by the upper layer
Use Case:	Notify reception of communication data to upper layer
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00400)

4.1.8 [RS_BRF_01072] AUTOSAR BSW shall provide callout functions in order to implement certain functionality in integrator code

⌈

Type:	Valid
Description:	AUTOSAR BSW shall provide callout functions in order to implement certain functionality in integrator code
Rationale:	In order to allow programmable customization of a module’s behavior a module can provide callout functions to integrator code
Use Case:	Implementation of protection hook, callout during start-up and shutdown
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.1.9 [RS_BRF_01080] AUTOSAR shall allow access to internal and external peripheral devices

⌈

Type:	Valid
Description:	AUTOSAR shall allow access to peripheral devices which are either directly linked to the MCU (internal devices), or via an I/O bus (external devices)
Rationale:	Although microcontrollers come with a variety of on-chip internal devices, here is the need to increase the number of devices by connecting them to an I/O bus. AUTOSAR needs to support both
Use Case:	Internal EEPROM and additional EEPROM on SPI
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435)

4.1.10 [RS_BRF_01088] AUTOSAR shall offer interfaces which allow to express high level application communication needs

⌈

Type:	Valid
Description:	AUTOSAR shall offer interfaces which allow applications (functionality organized in separate software components and spread over several ECUs) to express communication needs on an abstract level, and then organize the communication needs accordingly (so-called Partial Networking)
Rationale:	This abstract level allows to abstract from any bus or software component mapping dependencies
Use Case:	Request all communication needed for a light management in a car
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460, RS_Main_00200)

4.1.11 [RS_BRF_01096] AUTOSAR shall support start-up and shutdown of ECUs

⌈

Type:	Valid
Description:	AUTOSAR BSW and RTE shall define how to start-up ECUs, and how to shut them down if needed. This includes initialization/deinitialization of Basic Software Modules and hardware
Rationale:	Basic functionality of any IT system
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.1.12 [RS_BRF_01104] AUTOSAR shall support sleep and wake-up of ECUs and buses

⌈

Type:	Valid
Description:	AUTOSAR BSW and RTE shall define how to set ECUs and buses to sleep, and how to wake them up
Rationale:	Basic functionality of any embedded battery powered system
Use Case:	Parked car
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.1.13 [RS_BRF_01112] AUTOSAR shall offer interfaces to boot loaders

⌈

Type:	Valid

Description:	AUTOSAR shall offer interfaces which allow outside boot loader software to interact with AUTOSAR
Rationale:	Boot loaders differ widely and are therefore not part of the AUTOSAR specification. There is however the need to interact with boot loaders
Use Case:	Diagnostic request to reflash an ECU
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.1.14 [RS_BRF_01120] AUTOSAR shall support re-flashing of configured BSW data

⌈

Type:	Valid
Description:	AUTOSAR shall define which configurable BSW data items are allowed to be re-flashed after system start up
Rationale:	Reflashing of BSW data allows to use the same ECU in higher quantities because it can be used inside more cars, thus reducing cost
Use Case:	Adapt ECUs to country specific regulations, adapt to right/left hand steering
Dependencies:	--
Supporting Material:	--

⌋() // RS_Main_00360

4.1.15 [RS_BRF_01128] AUTOSAR shall allow software components to be started before all BSW modules are initialized

⌈

Type:	Valid
Description:	AUTOSAR shall define rules to allow software components to be started before all BSW modules and all parts of the RTE are initialized
Rationale:	If parts of the BSW are not used by all software components on the ECU, and if some software components only run in specific situations, not initializing the BSW parts which are exclusively used by the latter software components may significantly reduce power consumption. Also, system start up time for the other software components is reduced, allowing faster reaction
Use Case:	Burglar alarm on LIN bus which is periodically checked by the ECU. Only if the alarm fires, the CAN bus initialization needs to be performed to inform the rest of the system
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00200)

4.1.16 [RS_BRF_01136] AUTOSAR shall enable BSW modules to choose from different sets of initialization parameters

⌈

Type:	Valid
Description:	AUTOSAR shall support BSW modules to be able to choose from different sets of initialization parameters
Rationale:	Especially for the attached hardware, the BSW software needs to be prepared to run with slightly different settings of the hardware parameters
Use Case:	Prepare the CAN bus software to initialize the bus with different speed settings
Dependencies:	--
Supporting Material:	--

⌋() // RS_Main_00360

4.1.17 [RS_BRF_01144] AUTOSAR shall support configuration parameters which allow to trade interrupt response time against runtime

⌈

Type:	Valid
Description:	AUTOSAR shall support configuration parameters which allow to trade interrupt response time against overall runtime
Rationale:	The decision how many actions are performed within an interrupt, and how many actions are assigned to a decoupled task, cannot be done in general. It very much depends on the requirements on response time to external events and the overall system load. AUTOSAR therefore needs to allow the system integrator to make the trade-offs
Use Case:	In case of a CAN interrupt, the complete data handling can be done in the interrupt. With respect to runtime, this is the most efficient solution. However, as a result, other interrupts will be blocked out longer, increasing interrupt response time. As an alternative, data can be passed on to an asynchronous running task, and data handling can be performed there. However, this will need additional run time and maybe more intermediate storage
Dependencies:	--
Supporting Material:	--

⌋()

4.1.18 [RS_BRF_01152] AUTOSAR shall support limited dynamic reconfiguration

⌈

Type:	Valid
Description:	AUTOSAR shall support dynamic reconfiguration of BSW modules as long as the configuration changes have been part of the overall set of configuration options used to generate the BSW module. To be able to do this, AUTOSAR shall clearly define for each BSW module to which extent reconfiguration may take place
Rationale:	Although AUTOSAR is a statically configured system, a certain amount of reconfiguration is necessary to adapt to changing environment. To keep the system static, all possible configuration modification have to be present in the generated BSW module code
Use Case:	Change bus communication speed
Dependencies:	--
Supporting Material:	--

⌋() // RS_Main_00360

4.1.19 [RS_BRF_00206] AUTOSAR shall support multi-core MCUs

⌈

Type:	Valid
Description:	AUTOSAR shall support to use multi-core MCUs with only one common binary managing one or more of the MCU cores
Rationale:	Having one common binary for a multi-core MCU has the following benefits - Enable efficient parallelization of functions - Support sharing of peripherals - Upward and downward scalability in number of cores - Support migration of strongly integrated single applications from single to multi-core
Use Case:	High performance computing applications (e. g. signal processing applications) Integration of formerly separated applications into one multi-core ECU
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435)

4.1.20 [RS_BRF_01160] AUTOSAR shall support BSW distribution on multi-core MCUs

⌈

Type:	Valid
Description:	AUTOSAR shall define rules how BSW modules can be distributed to cores in multi-core MCUs, including rules to allow the BSW modules to run on several cores in parallel
Rationale:	Usage of multi-core MCUs will only result in optimal performance gains if as much BSW as possible is run locally on the same core as the caller. However, the requirements on certain modules – e. g. a centralized buffering for diagnostic data – does not always allow this. Therefore, AUTOSAR needs to establish rules to allow for the maximum possible performance, and give adequate configuration support
Use Case:	Run one communication bus on one core and another communication bus on another core, but still support a gateway between the buses
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.1.21 [RS_BRF_01168] AUTOSAR BSW and RTE shall support MCUs with memory write protection

⌈

Type:	Valid
Description:	AUTOSAR BSW and RTE shall support MCUs which offer memory protection which catches illegal write accesses

Rationale:	MCUs which support memory protection are one possibility to implement important parts of safety concepts. They are available from different vendors and used in automotive environment. Therefore, they need to be supported by AUTOSAR.
Use Case:	To combine functionality of several applications (represented by Software Components) of different ASIL level on the same ECU it is necessary to make sure that a lower level application cannot illegally overwrite memory of the higher level application(s).
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00435, RS_Main_00100)

4.1.22 [RS_BRF_00057] AUTOSAR shall define a memory mapping mechanism

⌈

Type:	Valid
Description:	AUTOSAR shall define a memory mapping mechanism which allows collecting data contributions of application software and Basic Software in separate memory segments
Rationale:	Use special micro-controller properties like fast and slow memory areas, memory protection capabilities, etc.
Use Case:	Collect all data of an OSApplication and protect it with memory protection hardware support
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00435, RS_Main_00100)

4.1.23 [RS_BRF_01176] The RTE shall be the only interfacing layer between software components and the BSW

⌈

Type:	Valid
Description:	In the AUTOSAR Layered Software Architecture, the RTE shall be the only interfacing layer between software components and the BSW Note: The I/O Hardware Abstraction and Complex Drivers interface to software components like software components, thus the RTE is still the only interfacing layer to the BSW
Rationale:	Installing a clear borderline between application and BSW, and centralizing the necessary adaptations from the not-yet-mapped software components to the specific properties of an ECU BSW
Use Case:	Mapping of port names to function calls inside the BSW with the correct parameters
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.1.1 [RS_BRF_01184] AUTOSAR shall support different methods of degradation

⌈

Type:	Valid
Description:	AUTOSAR shall support different standardized methods to degrade the functionality of an AUTOSAR system
Rationale:	Depending on specific states of an ECU or of a complete system, either the full functionality cannot be available any more (example: hardware problem) or need not be available any more (example: parked car). AUTOSAR must support system and ECU degradation to properly react to such states. Main reason is to save energy
Use Case:	Partial Networking, Pretended Networking, ECU Degradation Shut off complete buses (network management), remove nodes from buses to save power (partial networking, pretended networking), halt ECUs while no activity is going on
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460, RS_Main_00420, RS_Main_00200)

4.1.2 [RS_BRF_01192] AUTOSAR shall document all architectural constraints which exist to use the RTE and the BSW

⌈

Type:	Valid
Description:	AUTOSAR shall document all architectural constraints which exist to use the RTE and the BSW
Rationale:	AUTOSAR is specified with clear use-cases in mind. This needs to be documented to avoid usage of AUTOSAR where it is not suitable
Use Case:	Constraint that AUTOSAR is designed for 16bit processors upwards, constraint that for multi-core systems it is necessary to access all memory from all cores etc.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00400)

4.2 Operating System

4.2.1 [RS_BRF_01200] AUTOSAR OS shall be backwards compatible to OSEK OS

⌈

Type:	Valid
Description:	AUTOSAR OS shall be backwards compatible to ISO 17356-3 (OSEK). This means, that all functionality of OSEK OS can be found in AUTOSAR OS. It also means that only extensions to OSEK OS need to be specified as separate features
Rationale:	OSEK OS is an established standard in the automotive industry and in use and proven in a large number of ECUs. Therefore, it shall be reused for AUTOSAR
Use Case:	OSEK OS was an important factor in migration to AUTOSAR
Dependencies:	--
Supporting Material:	Specification of OSEK (ISO 17356-3)

⌋(RS_Main_00100, RS_Main_00420)

4.2.2 [RS_BRF_01208] AUTOSAR OS shall support to start lists of tasks regularly

⌈

Type:	Valid
Description:	AUTOSAR OS shall support to start tasks based on a static list which describes in which order the tasks shall be activated, and at which counter value
Rationale:	This is the typical way strictly timed systems are implemented
Use Case:	Periodic systems which do not consist of one task, but a sequence of tasks with need to be started constantly, either based on timing, or on angle values
Dependencies:	--
Supporting Material:	Note: such static lists are called ScheduleTables

⌋(RS_Main_00100)

4.2.3 [RS_BRF_01216] AUTOSAR OS shall support to synchronize ScheduleTables to an outside time source

⌈

Type:	Valid
Description:	AUTOSAR shall offer interfaces to synchronize ScheduleTables to outside time values
Rationale:	If the time source which governs the ScheduleTable is not available to the OS, it must be possible to run the ScheduleTable on a local time source, and offer interfaces to actively resynchronize the ScheduleTable based on the outside time source
Use Case:	Synchronize ScheduleTables to the FlexRay bus time
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.2.4 [RS_BRF_01224] AUTOSAR OS shall support timing protection

⌈

Type:	Valid
Description:	AUTOSAR OS shall offer functionality to limit runtime and activation frequency of tasks and interrupts
Rationale:	This is a pre-requirement to catch problems with interrupt lines (babbling idiot) and certain programming bugs
Use Case:	Disable an interrupt line if this interrupt line fires too often
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.2.5 [RS_BRF_01232] AUTOSAR OS shall support isolation and protection of application software

⌈

Type:	Valid
Description:	AUTOSAR OS shall support to organize all objects handled by the OS such that they can be assigned to different entities (OSApplications) and that access between OSApplications is restricted. This includes usage of hardware memory protection. Note: Assignment of Software Components to OSApplications needs to be done outside the OS
Rationale:	This is a pre-requirement to install protection mechanisms for higher level BSW and Software Components
Use Case:	Usage of memory protection properties of microcontrollers to catch erroneous write access of software components
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00100)

4.2.6 [RS_BRF_01240] AUTOSAR OS shall support communication between OSApplications

⌈

Type:	Valid
Description:	AUTOSAR shall offer a communication mechanism to transfer data between OSApplications
Rationale:	With OSApplications protected against each other, and in multi-core systems, the OS needs to offer functionality to transport data between OSApplications
Use Case:	When a port is established between Software Components in different OSApplications, the RTE needs to use this functionality to transport port data
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00100)

4.2.7 [RS_BRF_01248] AUTOSAR OS shall support to terminate and restart OSApplications

⌈

Type:	Valid
Description:	AUTOSAR OS shall support to terminate and – if wanted - restart OSApplications
Rationale:	If an OSApplication encounters an error, the error strategy of the ECU needs to decide if this OSApplication can be permitted to continue working, and eventually terminate or terminate and restart the OSApplication. The OS needs to offer the necessary functionality
Use Case:	Memory protection error in an OSApplication which cannot be salvaged without terminating the OSApplication
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00100)

4.2.8 [RS_BRF_01256] AUTOSAR OS shall offer support to switch off cores

⌈

Type:	Valid
Description:	If configured and supported by hardware, AUTOSAR OS shall support to switch off cores if no task or interrupt is ready to run
Rationale:	A core does not need to run if there is no activity going on. This is detected inside the OS when the OS goes into the internal idle state. At this point, switching off the core will save energy
Use Case:	Energy saving
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.2.9 [RS_BRF_01264] AUTOSAR OS shall support multi-core deadlock free mutual exclusion

⌈

Type:	Valid
Description:	AUTOSAR shall support multi-core deadlock free mutual exclusion which is safe against multiple nested usage across cores
Rationale:	In a multi-core system a mutual exclusion mechanism is needed to synchronize different cores. In order to keep system integrity, this mutual exclusion mechanism shall be deadlock free
Use Case:	Concurrent access to DEM fault memory
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.2.10 [RS_BRF_01272] AUTOSAR OS shall offer functionality to allow Software Components time measurement

⌈

Type:	Valid
Description:	AUTOSAR OS shall offer functionality to allow Software Components time measurement, that is measure the time between two specific calls to the OS
Rationale:	Time can anyway be used by the OS to schedule tasks (see OSEK specification). This gives an easy means to Software Components to measure time and such the behavior of the Software Component
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00100)

4.3 Runtime Environment (RTE)

4.3.1 [RS_BRF_01280] AUTOSAR RTE shall offer the external interfaces between Software Components and between Software Components and BSW

⌈

Type:	Valid
Description:	The RTE shall architecturally separate the Software Components from the rest of the system by offering all interfaces to Software Components located on the same ECU. These interfaces are necessary to connect ports between Software Components and between Software Components and BSW. This type of interface is called 'AUTOSAR interface' and encompasses all types of ports
Rationale:	Encapsulation of Software Components from their environment
Use Case:	Integrate Software Components in different hardware architectures
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.2 [RS_BRF_01288] AUTOSAR RTE interfaces shall be independent of the addressee

⌈

Type:	Valid
Description:	The RTE shall offer generic interfaces which are independent of the fact if the addressed entity is the BSW, if the addressed entity is a Software Component on the same core of the ECU, a different core of the ECU, or on a different ECU
Rationale:	Necessary to allow Software Components to be retargeted to different ECUs
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00140, RS_Main_00060)

4.3.3 [RS_BRF_01296] AUTOSAR RTE shall support and handle single and multiple instantiation of Software Components

⌈

Type:	Valid
Description:	The RTE shall offer generic interfaces which support single and multiple instantiation of Software Components and which are transparent for the addressee
Rationale:	Addressees (Software Components, BSW) are written without knowing if the originator is instantiated. As a consequence, the originating Software Component has to take into account multiple instantiation. The RTE has to handle the instantiation and hide it from the Basic Software
Use Case:	--
Dependencies:	--

Supporting Material:	--

⌋(RS_Main_00060)

4.3.4 [RS_BRF_01304] AUTOSAR RTE shall support broadcast communication

⌈

Type:	Valid
Description:	AUTOSAR RTE shall support data broadcast (sender/receiver) communication including support of queuing and non-queuing strategies on the receiver side
Rationale:	Support 1:n communication
Use Case:	The same data can be used by different Software Components
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.5 [RS_BRF_01312] AUTOSAR RTE shall support procedure-call communication

⌈

Type:	Valid
Description:	AUTOSAR RTE shall support calling of subroutines (client/server call, including remote procedure calls)
Rationale:	Requesting synchronous functionality or data
Use Case:	Call of system services like NVRAM manager services
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.6 [RS_BRF_01320] AUTOSAR RTE shall schedule SWC and BSW modules

⌈

Type:	Valid
Description:	The RTE shall support scheduling of executable entities (runnable entities) defined inside Software Components and the BSW Note: in case of Software Components the executable entities are called ‘runnables’, in case of the BSW ‘main function’. To shorten the description, the term ‘runnables’ is used for both
Rationale:	The runnable entities which need to be run based on certain events within the system need to be mapped to OS objects and started according to the application needs
Use Case:	Start all runnable entities which shall be scheduled in a certain period to one OS task and execute them in a defined order
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.7 [RS_BRF_01328] AUTOSAR RTE shall support scheduling of executable entities on defined events

⌈

Type:	Valid
Description:	The RTE shall support a set of events which can be used to start executable entities, and offer the software components and BSW the necessary interfaces. Note: The offered events shall be based on current automotive requirements
Rationale:	Executable entities do need a reason to be run, in the simplest case a periodic trigger. This needs to be organized and handled by the RTE
Use Case:	Start executable entities because of periodic events, because data has arrived, because an error has occurred etc.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.8 [RS_BRF_01336] AUTOSAR RTE shall only run software component runnables inside tasks

⌈

Type:	Valid
Description:	If RTE is called by the BSW inside an interrupt, it shall not pass the interrupt on to the software component, but instead memorize actions to be initiated based on the interrupt, and perform these actions outside the interrupt in a task
Rationale:	Software components have to be independent of specific hardware. They cannot be implemented such that parts of it may be run in interrupt and handle the resulting restrictions (interrupt lock timing requirements etc.). Therefore, the RTE needs to decouple BSW and application
Use Case:	--
Dependencies:	--
Supporting Material:	

⌋(RS_Main_00060)

4.3.9 [RS_BRF_01344] AUTOSAR RTE shall support Software Component global data

⌈

Type:	Valid
Description:	The RTE shall support Software Component global data for each instance of a Software Component and offer all necessary interfaces to access such data; including the necessary implicit protection mechanisms against concurrent access
Rationale:	Global data which is used by several independent runnables cannot be assumed to be ECU global, because the Software Components do not know which of them are mapped to the same ECU. However, not all data is local

	to one runnable. Therefore, the RTE needs to support data which is shared between runnables of the same Software Component
Use Case:	Store received input data in a local storage which can be read to be processed later by other runnables
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.10 [RS_BRF_01352] AUTOSAR RTE shall offer direct read/write data access, and alternatively pre-read data before a runnable is called and post-write data after the runnable returns

⌈

Type:	Valid
Description:	For Sender-Receiver communication and internal variables, the RTE shall offer read or write data accesses which have immediate effect during the ongoing execution of the runnable and read or write data accesses which pre-read or post-write the data at the execution of the runnable. Note: this is called implicit and explicit data communication
Rationale:	There are different strategies to work on data: either to have a complete data set at start of a runnable, or to get data when needed. The same is true for writing data. AUTOSAR needs to support both strategies
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.11 [RS_BRF_01360] AUTOSAR RTE shall support explicit protection mechanisms against concurrent access

⌈

Type:	Valid
Description:	The RTE shall support explicit protection mechanisms against concurrent access which can be used by Software Components and BSW
Rationale:	Whereas protection mechanisms are implicitly done for internal variables, this is not sufficient if other use cases than internal variables are involved
Use Case:	Protect a non-reentrant subroutine against concurrent access. Protect a list of internal variables against concurrent access
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.12 [RS_BRF_01368] AUTOSAR RTE shall support calibration data

⌈

Type:	Valid
Description:	The RTE shall support calibration data and offer the necessary interfaces to

	Software Components
Rationale:	Calibration data is a standard means in automotive industry to adapt applications to environmental conditions
Use Case:	Adapt motor management to the characteristics of a specific engine
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.13 [RS_BRF_01376] AUTOSAR RTE shall support automatic re-scaling and conversion of port data elements

⌈

Type:	Valid
Description:	The RTE shall support automatic re-scaling and conversion of port data elements, if configured
Rationale:	Software Components may use different ranges or scaling to represent data. When they are created, it is not known to Software Components with which other Software Components they interact, and what scaling/representation they choose. Re-scaling therefore needs to be done automatically in the RTE
Use Case:	Temperatures in Celsius and Fahrenheit, different ranges like temperatures represented with base -40 Celsius or -50 Celsius etc.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.14 [RS_BRF_01384] AUTOSAR RTE shall support automatic range checks of data

⌈

Type:	Valid
Description:	The RTE shall support automatic range checks of data, if configured
Rationale:	Detect range violation of data and react properly
Use Case:	Especially if re-scaling of data is in place, the check of data on valid range is crucial for a working system
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.15 [RS_BRF_01392] AUTOSAR RTE shall support a bypass implementation

⌈

Type:	Valid
Description:	AUTOSAR shall provide support for implementation of bypass A bypass consists of directly reading/modifying/writing data managed by the RTE for the purpose of testing or rapid prototyping

Rationale:	To support the integration on a standard AUTOSAR software of different implementations of bypass tools and software
Use Case:	A Rapid Prototyping tool/software vendor provides an implementation that can be integrated on a standard RTE. A Tier 1 supplier integrates the Rapid Prototyping software in an AUTOSAR ECU. The Rapid Prototyping tool is used by an OEM on the ECU provided by the Tier 1 supplier to evaluate/test new control algorithms
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.3.16 [RS_BRF_01400] AUTOSAR RTE shall offer configurable test hooks

⌈

Type:	Valid
Description:	For testing, the RTE shall offer configurable hooks which allow to be informed about actions taken inside the RTE
Rationale:	For testing and debugging, it needs to be possible to follow-up the actions of the RTE
Use Case:	Debugging of BSW and of Software Components, time measurement
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00060)

4.4 Services

4.4.1 [RS_BRF_01408] AUTOSAR shall provide a service layer that is accessible from each basic software layer

⌈

Type:	Valid
Description:	General management functionality shall be provided in the services layer of the architecture. These services are standardized interfaces which are mostly MCU and hardware independent. If applicable, they are made accessible to the application via the RTE as standardized AUTOSAR interfaces. In this case the interface to the basic software is called a system service
Rationale:	Management functionality must be available to all modules and layers of the system
Use Case:	Time service
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00400)

4.4.2 [RS_BRF_01416] AUTOSAR services shall support standardized handling of non-volatile memory data

⌈

Type:	Valid
Description:	The NV-memory service shall be the main interface to make persistent application data available to application software and Basic Software Modules. This includes read, write and erase access. The provided interface shall allow concurrent access and shall be independent from the underlying hardware
Rationale:	Portability of software components needs flexible assignment of available memory to different memory areas and/or memory hardware
Use Case:	Providing configuration data to application software
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00440, RS_Main_00150)

4.4.3 [RS_BRF_01424] AUTOSAR services shall support communication services

⌈

Type:	Valid
Description:	AUTOSAR shall support communication services that provide a unified protocol and hardware independent interface of the communication stack to the RTE
Rationale:	Hide protocol and message properties from application
Use Case:	Diagnostic communication services, COM
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.4.4 [RS_BRF_01432] AUTOSAR services shall support system time services

⌈

Type:	Valid
Description:	AUTOSAR shall provide time services for applications and Basic Software modules. Time services shall provide measurement of absolute time
Rationale:	Provide global basic system time information
Use Case:	Time service for high precision local time measurement, providing synchronized time base, e. g. originating from FlexRay or TTCAN
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435, RS_Main_00150)

4.4.5 [RS_BRF_01440] AUTOSAR services shall support system diagnostic functionality

⌈

Type:	Valid
Description:	AUTOSAR shall provide basic diagnostic services for applications and Basic Software modules to detect or report errors and react on fault modes. These services are mostly MCU and hardware independent
Rationale:	Provide global basic error management and handling functionality
Use Case:	Diagnostic error management, function inhibition management
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260, RS_Main_00011)

4.4.6 [RS_BRF_01448] AUTOSAR services shall support mode and state management

⌈

Type:	Valid
Description:	AUTOSAR shall provide basic mode and state management services for applications and Basic Software modules
Rationale:	Basic system management functionality
Use Case:	Management of ECU States and Basic Software Modes, management of communication modes, network management
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.4.7 [RS_BRF_01456] AUTOSAR services shall provide system wide cryptographic functionality

⌈

Type:	Valid
Description:	AUTOSAR shall provide unified cryptographic service interfaces for applications. These interfaces allow for basic operation of e. g. encryption, hash computation, key exchange … The cryptographic services are independent from specific cryptographic algorithms
Rationale:	Provide unified encryption functionality to software applications
Use Case:	Crypto service manager
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170)

4.4.8 [RS_BRF_01464] AUTOSAR services shall support standardized handling of watchdogs

⌈

Type:	Valid
Description:	The watchdog service shall be the main interface to supervise timing behavior of application software. The provided interface shall allow concurrent access and shall be independent from the underlying hardware
Rationale:	Portability of software components needs flexible assignment of available hardware resources
Use Case:	Detect end-less looping software components
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011)

4.5 Mode Management

4.5.1 [RS_BRF_01472] AUTOSAR shall support modes

⌈

Type:	Valid
Description:	AUTOSAR RTE and BSW shall support a mode management which offers modes which can be requested (mode user) and switched (mode manager), and where the mode users can react on a mode change
Rationale:	Modes are a means to indicate different states of software. BSW and Software Components need to be able to set modes, and to react on mode changes in a standardized way.
Use Case:	Different stages of system start-up. Switching of LIN schedule tables. Setting the application in a limp-home state based on errors.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.2 [RS_BRF_01480] AUTOSAR shall support software component local modes, ECU global modes, and system wide modes

⌈

Type:	Valid
Description:	AUTOSAR shall support different scopes of modes: software component local, ECU global, and system wide
Rationale:	As modes are a means to indicate different states of software, the involved entities interested in the specific mode may be one software component, several software components (on the same ECU, or on different ECUs), or BSW and software components. All these different scopes need to be supported.
Use Case:	Software component local: setting the application in a limp-home state based on a detected fault ECU global: Different stages of system start-up. System wide: act on low battery
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.3 [RS_BRF_01488] AUTOSAR RTE and BSW shall support standardized modes for ECU start up, ECU shut down with restart, and for putting an ECU to sleep

⌈

Type:	Valid
Description:	AUTOSAR shall support standardized modes for ECU start up, ECU shut down with restart, and putting an ECU to sleep. This includes description of the activities which need to take place when such a mode is reached
Rationale:	As these states of an ECU are central and therefore cannot be individually defined and coordinated by software components, they need to be offered by

	the BSW and RTE
Use Case:	While one software component may currently be mainly idle and thus willing to set an ECU to sleep (request the SLEEP mode), other software components may not have reached the state. The actions which need to be performed if an ECU goes e. g. in SLEEP mode need to be standardized and clearly documented
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.4 [RS_BRF_01496] AUTOSAR shall standardize how events which move an ECU out of the SLEEP mode are handled

⌈

Type:	Valid
Description:	In case an ECU is set in SLEEP mode, different external sources can cause the ECU to be moved out of the SLEEP mode. The mode management shall handle these different external sources in a standardized way. This includes e. g.: - Engine Off Time - Battery Charge Monitoring - HVAC Auxiliary Engine heater - Security/Theft monitoring
Rationale:	Individual handling will complicate the system and harm extendibility of AUTOSAR
Use Case:	Handle a CAN wake up and a wake up caused by a timer interrupt similar
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.5 [RS_BRF_01504] AUTOSAR shall handle memory corruption resulting from ECU sleep

⌈

Type:	Valid
Description:	In case an ECU is set in SLEEP mode, the mode management shall create a memory checksum, and based on the checksum check when returning from SLEEP if the memory content is still valid
Rationale:	The time an ECU is in the state SLEEP is not limited. As a result, the memory may have become invalid. This needs to be checked to detect memory errors, and take necessary action
Use Case:	Do not continue with normal operation in case of memory failure during sleep, but restart the ECU instead
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.6 [RS_BRF_01512] AUTOSAR mode management shall support standardized modes for handling of communication buses

⌈

Type:	Valid
Description:	AUTOSAR mode management shall support standardized modes for handling of communication buses. Management has to take into account the application and Basic Software communication needs and the bus hardware state/availability
Rationale:	Communication buses are shared objects between application and Basic Software (e. g. diagnostic) and need to be centrally managed in order to provide the capability to switch off buses
Use Case:	While one software component may currently not need communication on a specific bus channel but access is still request by diagnostic communication. In this case the mode management is not allowed to put the bus to sleep
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.7 [RS_BRF_01520] AUTOSAR RTE shall automatically adapt the runnable management on a mode switch

⌈

Type:	Valid
Description:	The RTE shall, defined by configuration, automatically adapt the management of runnables on a mode switch: it shall set events which start runnables to active or passive, and start configured runnables when entering or leaving a mode
Rationale:	Processing needs may change when a mode switch signals a different state. The RTE shall then adapt the scheduling of runnables accordingly.
Use Case:	When changing in a limp-home state, do not restart runnables which depend on full availability of all hardware resources, and start runnables instead which depend on restricted hardware functionality
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.8 [RS_BRF_01528] AUTOSAR mode management shall perform actions based on the evaluation of configured rules

⌈

Type:	Valid
Description:	AUTOSAR mode management shall offer to configure rules and evaluate them during runtime. Based on the result, it shall be possible to execute configurable actions, including the ability to switch a mode. This shall be a generic functionality which can be tailored to application and Basic Software needs. All restrictions with respect to actions need to be documented
Rationale:	Besides ECU global modes like STARTUP, modes are user defined. The BSW and RTE can therefore not define rules or actions, but needs to offer a highly configurable method to define rules and actions as result of the rules

Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.5.9 [RS_BRF_01536] For system wide modes, AUTOSAR mode management shall forward ECU local mode requests to all involved ECUs

⌈

Type:	Valid
Description:	For system wide modes, In case of a mode requests which originates on one ECU, but where the mode users are spread over several ECUs, AUTOSAR mode management shall forward such a mode request to all involved ECUs
Rationale:	Necessary to support system global modes
Use Case:	Distributing information if Ignition is on or off between ECUs
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460)

4.6 Communication via Bus

4.6.1 [RS_BRF_01544] AUTOSAR communication shall define transmission and reception of communication data

⌈

Type:	Valid
Description:	AUTOSAR communication shall define the way how communication data is handled, how data is transmitted, and how an indication of data is transformed into a data reception
Rationale:	Exchange of data
Use Case:	Exchange of data within the vehicle network or within an ECU internally
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.2 [RS_BRF_01552] AUTOSAR communication shall separate bus independent functionality from bus dependent functionality

⌈

Type:	Valid
Description:	AUTOSAR communication shall separate bus independent functionality from bus dependent functionality
Rationale:	As many modules as possible shall be re-usable for all buses. This reduces implementation effort and supports modularization. Additionally, the RTE should not need to cope with bus specifica and therefore needs to have a bus independent entry to the communication stack
Use Case:	Filling data into entities which are later transferred etc.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.3 [RS_BRF_01560] AUTOSAR communication shall support mapping of signals into transferrable protocol data units

⌈

Type:	Valid
Description:	AUTOSAR communication shall handle the mapping of (application) data types (signals) into data entities which are suitable to be transferred on the communication bus (I-PDUs)
Rationale:	Format and size of signals do often not fit to the properties of a bus. Data (signals) needs to be combined or even chopped into parts before transfer and on the receiver side handled accordingly
Use Case:	Store a Boolean value in an IPDU as an 1-bit-value, which in the function interface is passed as an integer value
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.4 [RS_BRF_01568] AUTOSAR communication stack shall support fixed size and dynamic size signals

⌈

Type:	Valid
Description:	AUTOSAR communication shall support signals of fixed size, and signals of dynamic size with an upper limit on size
Rationale:	Exchange information of fixed and/or dynamic length using an automotive communications bus
Use Case:	Simple boolean value (clam15 on/off) as example for fixed size, data from a cluster instrument (often dynamic size)
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.5 [RS_BRF_01576] AUTOSAR communication shall support a signal gateway

⌈

Type:	Valid
Description:	AUTOSAR communication shall support a signal gateway which receives signals and sends them out again unchanged
Rationale:	The receiver of a signal may reside on a different bus which is connected via a gateway ECU. The BSW on the gateway ECU needs to receive the signal and retransmit it without application involvement
Use Case:	Signal creator on an ECU which is exclusively connected to a CAN bus, signal consumer on a FlexRay bus connected via a gateway ECU
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00230, RS_Main_00430)

4.6.6 [RS_BRF_01584] AUTOSAR communication shall support an IPDU gateway

⌈

Type:	Valid
Description:	AUTOSAR communication shall support an IPDU gateway which receives IPDUs and directly sends them out again
Rationale:	Efficient gatewaying of complete protocol data units from one bus to another
Use Case:	Gateway between two CAN buses. It could be used in some cases as an efficient alternative for a signal gateway
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00230, RS_Main_00430)

4.6.7 [RS_BRF_01592] AUTOSAR communication shall offer data transfer on user request, time based, and requested via the underlying bus

⌈

Type:	Valid
Description:	AUTOSAR communication shall offer data transfer on user request, time based, and requested remotely via the underlying bus, or if necessary a combination of these basic methods
Rationale:	Different kind of buses traditionally schedule data transfer differently. These different methods need to be supported by AUTOSAR
Use Case:	Data on LIN is transferred according to the LIN schedule table handled inside LIN. CAN data is mostly scheduled periodically, sometimes on event (user request). On the reception side, data may be polled periodically or received on event. J1939 request management
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.8 [RS_BRF_01600] AUTOSAR communication shall support time-out handling

⌈

Type:	Valid
Description:	AUTOSAR communication shall support time-out handling for data transmission and data reception
Rationale:	If data is transmitted, and the transmission is not acknowledged in time by the bus, or if data reception is expected and does not occur in time, this shall be detected by the BSW
Use Case:	Detect missing signals, especially when data should arrive periodically, which indicates problems on the sending node
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00430)

4.6.9 [RS_BRF_01608] AUTOSAR communication shall support to filter signals

⌈

Type:	Valid
Description:	AUTOSAR communication shall support to filter signals such that unchanged and/or implausible data is not forwarded to a receiver
Rationale:	Signals occupy a number of bits in an IPDU. The value range allowed for the signal may be smaller than the range representable in the bits. AUTOSAR shall offer the functionality to discard signal data which is outside the allowed range
Use Case:	Discard implausible temperature values and/or filter unchanged data
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00430)

4.6.10 [RS_BRF_01616] AUTOSAR communication shall support initial values for signals

⌈

Type:	Valid
Description:	AUTOSAR communication shall support initial values for signals
Rationale:	Signals may be read before they are written. To indicate this, an initial value may be specified which can be detected by the application. Likely, if data has for some time not arrived in time, the initial value is used by the application in a limp-home state
Use Case:	See rationale
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.11 [RS_BRF_01624] AUTOSAR communication shall support data conversion between big endian and little endian data representation

⌈

Type:	Valid
Description:	AUTOSAR communication shall support data conversion between big endian and little endian data representation
Rationale:	Different ECUs may have different data representation. As software components are written hardware independent and are not aware about data representation on the other side of a sender / receiver connection, they cannot handle differences in data representation. AUTOSAR therefore is responsible to handle the conversion internally. Note: only big endian and little endian is supported
Use Case:	See rationale
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.12 [RS_BRF_01632] AUTOSAR communication shall support data consistency of groups of signals

⌈

Type:	Valid
Description:	AUTOSAR communication shall support data consistency when sending a configured group of signals
Rationale:	If a signal group is send out by the application, a modification of an individual signal of a group shall only come into effect after the group update has been completed. Thus, the transferred values are consistent
Use Case:	Consistent values of several signals which depend on each other
Dependencies:	--

Supporting Material:	--

⌋(RS_Main_00430)

4.6.13 [RS_BRF_01640] AUTOSAR communication shall support transmit and receive cancelation

⌈

Type:	Valid
Description:	AUTOSAR communication shall support to cancel transmissions and receptions if they are still in progress
Rationale:	To speed up clean-ups: all pending transmission requests can be cancelled
Use Case:	Shorten waiting time in case of shutdown
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.14 [RS_BRF_01648] AUTOSAR communication shall support transfer of data sizes larger than the maximum transmission unit of the underlying bus

⌈

Type:	Valid
Description:	AUTOSAR communication shall support to transfer data sizes which are not restricted by network packet sizes offered by a specific bus
Rationale:	Applications are written bus independent and cannot take into account restrictions of the bus the data communication is finally mapped to
Use Case:	Transfer more than 8 bytes on the CAN bus
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.6.15 [RS_BRF_01656] AUTOSAR communication shall support XCP

⌈

Type:	Valid
Description:	AUTOSAR communication shall support XCP as specified by ASAM
Rationale:	XCP is the most widely used protocol for testing in automotive
Use Case:	Testing
Dependencies:	--
Supporting Material:	XCP specification of ASAM

⌋(RS_Main_00430)

4.6.16 [RS_BRF_01664] AUTOSAR communication shall support a state management of buses

⌈

Type:	Valid
Description:	AUTOSAR communication shall support a state management which keeps track of user requests to use bus communication, and detects if a bus is currently not used for communication
Rationale:	Software components have individual needs with respect to bus communication, which need to be coordinated
Use Case:	Detect for each bus if communication is requested, and inform if the state of a bus changes
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460, RS_Main_00430)

4.6.17 [RS_BRF_01672] AUTOSAR communication state management shall support dynamic bus access limitation

⌈

Type:	Valid
Description:	AUTOSAR communication state management shall support to restrict and reopen access to buses during runtime
Rationale:	An ECU may be forced to restrict bus access in case of specific internal states
Use Case:	Diagnostic may block bus access for other accessors and only keep it open for diagnostic to satisfy legal requirements
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460, RS_Main_00430)

4.6.18 [RS_BRF_01680] AUTOSAR communication shall support mechanism to keep a bus awake, and to be kept awake by a bus

⌈

Type:	Valid
Description:	AUTOSAR shall support mechanism to keep the bus awake, thus indicating usage of a bus. Likely, AUTOSAR shall support mechanism to be kept awake by the bus, thus indicating usage of a bus by other bus nodes. Note: this functionality is commonly called Network Management
Rationale:	Unless supported otherwise by hardware, a bus can only be put to sleep if all nodes on a bus agree. Therefore, a special bus specific protocol is needed to reach agreement: a node has to indicate if it needs the bus, or if it does not need the bus any more
Use Case:	Energy saving: switch off buses
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00420, RS_Main_00430, RS_Main_00460)

4.6.19 [RS_BRF_01688] AUTOSAR communication shall support to put buses synchronously to sleep

⌈

Type:	Valid
Description:	AUTOSAR network management shall support to put more than one bus synchronously to sleep
Rationale:	Buses may be logically linked. In this case, it shall be possible to configure that more than one bus act the same with respect to bus sleep
Use Case:	Legacy: if a software was written such that it assumed that all nodes reside on the same bus, and nodes are now distributes to two buses
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00420, RS_Main_00430, RS_Main_00460)

4.6.20 [RS_BRF_01696] AUTOSAR communication shall support selective shutdown of nodes while bus communication is active

⌈

Type:	Valid
Description:	AUTOSAR network management shall set a bus to sleep if all communication has internally been stopped even if other bus nodes are still active, if it is possible to do this without interfering with the other nodes
Rationale:	With special hardware support it is possible to remove a node from a bus although other nodes still use the bus
Use Case:	Power saving
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00420, RS_Main_00430, RS_Main_00460)

4.7 Communication buses

4.7.1 [RS_BRF_01704] AUTOSAR communication shall support the CAN communication bus

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the CAN communication bus with 11 bit CAN identifiers, 29 bit CAN identifiers, and 11 bit CAN identifiers which are extended by software bits in the CAN data part
Rationale:	All these methods are in current use
Use Case:	--
Dependencies:	--
Supporting Material:	ISO 11898

⌋(RS_Main_00430)

4.7.2 [RS_BRF_01712] AUTOSAR communication shall support the adaptable speed offered by CAN FD

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the ability of CAN FD to transport data at a higher speed than the header information
Rationale:	The current usage of CAN buses already often uses up the available bandwidth of standard CAN. CAN FD offers the ability to transport data faster and thus increase the bus bandwidth
Use Case:	See rationale
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.7.3 [RS_BRF_01720] AUTOSAR communication shall support the standardized transport protocol for Diagnostics over CAN

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the standardized protocol for Diagnostic communication over CAN (ISO15765-2) to support diagnostics as well as other applications which need to transfer long data blocks
Rationale:	Legal requirement for diagnostics
Use Case:	--
Dependencies:	--
Supporting Material:	Specification of ISO15765-2 (DoCAN, Part 2: Transport protocol and network layer services)

⌋(RS_Main_00260, RS_Main_00430)

4.7.4 [RS_BRF_01728] AUTOSAR communication shall support J1939 transport protocol

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the J1939 transport protocol as established standard for trucks
Rationale:	J1939 is the de-facto standard for trucks
Use Case:	--
Dependencies:	--
Supporting Material:	Specification of J1939-21

⌋(RS_Main_00430)

4.7.5 [RS_BRF_01736] AUTOSAR communication shall support dynamic allocation of addresses as requested by J1939 network management

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the dynamic allocation of addresses necessary to support J1939 network management
Rationale:	J1939 is the de-facto standard for trucks
Use Case:	--
Dependencies:	--
Supporting Material:	Specification of J1939-21

⌋(RS_Main_00260, RS_Main_00430)

4.7.6 [RS_BRF_01744] AUTOSAR communication shall support TTCAN

⌈

Type:	Valid
Description:	AUTOSAR communication shall support TTCAN as a superset of the CAN communication
Rationale:	TTCAN is a superset of CAN which adds a time-triggered mechanism on top of CAN. Differences between CAN communication and TTCAN communication shall be restricted to the hardware dependent architectural layer. All hardware independent modules assigned to CAN will work the same with CAN as with TTCAN
Use Case:	--
Dependencies:	--
Supporting Material:	ISO 11898-4 (CAN, Part 4: Time-triggered communication)

⌋(RS_Main_00430)

4.7.7 [RS_BRF_01752] AUTOSAR communication shall support FlexRay

⌈

Type:	Valid
Description:	AUTOSAR communication shall support FlexRay as specified by the FlexRay consortium
Rationale:	The FlexRay bus is widely used in automotive
Use Case:	--
Dependencies:	--
Supporting Material:	Specification of FlexRay

⌋(RS_Main_00430)

4.7.8 [RS_BRF_01760] AUTOSAR communication shall support the standardized transport protocol for Diagnostics on FlexRay

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the standardized protocol for diagnostics on FlexRay (ISO10681-2) to support diagnostics as well as other applications which need to transfer long data blocks
Rationale:	Legal requirement for diagnostics
Use Case:	--
Dependencies:	--
Supporting Material:	ISO 10681-2 (FlexRay ISO-TP), FlexRay AUTOSAR-TP

⌋(RS_Main_00260, RS_Main_00430)

4.7.9 [RS_BRF_01768] AUTOSAR communication shall support LIN

⌈

Type:	Valid
Description:	AUTOSAR communication shall support LIN 2.1 as specified by the LIN consortium
Rationale:	LIN is widely used in automotive
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.7.10 [RS_BRF_01776] AUTOSAR communication shall support Ethernet

⌈

Type:	Valid
Description:	AUTOSAR communication shall support Ethernet as an alternative bus for high data rates
Rationale:	Ethernet is in widely used outside the automotive industry and has lately overcome problems which formerly have precluded usage in automotive
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00430)

4.7.11 [RS_BRF_01784] AUTOSAR communication shall support the IP protocol stack

⌈

Type:	Valid
Description:	AUTOSAR communication shall support the IP protocol stack. This includes e. g. IPv4, IPv6, UDP, TCP, ARP, DHCP, ICMP
Rationale:	Ethernet is mostly used to connect to the outside world (non-AUTOSAR bus nodes). They use these protocols, therefore they need to be available inside AUTOSAR
Use Case:	Connect to a non-AUTOSAR telematics device, battery charging
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00280, RS_Main_00430)

4.7.12 [RS_BRF_01792] AUTOSAR shall support SPI

⌈

Type:	Valid
Description:	AUTOSAR shall support SPI as an I/O bus to connect external devices
Rationale:	SPI is the standard bus to connect external devices. Note: because the SPI bus is not used to directly connect ECUs, it does not need state management or network management
Use Case:	Communication with external EEPROM, external CAN etc.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130)

4.8 Memory Stack

4.8.1 [RS_BRF_01800] AUTOSAR non-volatile memory functionality shall be divided into a hardware dependent and independent layer

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory functionality shall be divided into a hardware dependent and independent layer
Rationale:	Access to persistent data shall be independent from the type of the actually used hardware. This enables portability of application software
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130, RS_Main_00400, RS_Main_00440)

4.8.2 [RS_BRF_01808] AUTOSAR non-volatile memory handling shall support different kinds of memory hardware

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory handling shall support different kinds of memory hardware, e. g. Flash memory and EEPROM. This also includes support for internal and external memory devices
Rationale:	Access to persistent data shall be independent from the type of the actually used hardware and how this hardware is connected. This enables portability of application software
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130, RS_Main_00440)

4.8.3 [RS_BRF_01816] AUTOSAR non-volatile memory functionality shall organize persistent data based on logical memory blocks

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory functionality shall organize persistent data based on logical memory blocks
Rationale:	Allow an ECU local efficient organization of persistent data, in particular with regard to grouping for read and write jobs. Abstract from device specific addressing and segmentation of memory
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00440)

4.8.4 [RS_BRF_01824] AUTOSAR non-volatile memory functionality shall provide a mapping of non-volatile memory into random access memory

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory functionality shall provide a mapping of non-volatile memory into random access memory. Loading and storing of data into the memory device shall be handled asynchronously and decoupled from high level application access
Rationale:	Non-volatile memory data has to be provided as random access memory to applications in order to make the data easily readable and writable by Basic Software Modules or application software
Use Case:	Reading configuration data required for BSW module initialization. Saving application state during shutdown
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00440)

4.8.5 [RS_BRF_01832] AUTOSAR non-volatile memory shall handle logical memory blocks independent of its physical address

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory shall handle logical memory blocks independent of its physical address
Rationale:	Make persistent data manageable independent from the location or base address inside a specific memory device
Use Case:	Wear leveling
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00150, RS_Main_00440)

4.8.6 [RS_BRF_01840] AUTOSAR non-volatile memory functionality shall secure integrity of memory blocks

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory handling shall be able to detect corrupted memory data and act appropriately. This could be achieved e. g. by error correction data or saving the data to multiple redundant memory areas together with a suitable data digest, identifying data correctness. Where possible and reasonable, fault reaction shall be transparent to the application
Rationale:	Writing or reading memory blocks may fail due to faulty hardware
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00440)

4.8.7 [RS_BRF_01848] AUTOSAR non-volatile memory functionality shall provide mechanisms to enhance hardware reliability

⌈

Type:	Valid
Description:	AUTOSAR non-volatile memory functionality shall provide mechanisms to enhance hardware reliability. Spread write access across physical address space to reduce individual memory cell wear level
Rationale:	Non-volatile memory hardware may have limited lifetime
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00440)

4.9 Microcontroller Abstraction and I/O

4.9.1 [RS_BRF_01856] AUTOSAR microcontroller abstraction shall provide access to internal MCU configuration

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide read and write access to internal MCU configuration registers, such register could also be memory mapped
Rationale:	Provide standardized access to common internal MCU resources required for proper efficient initialization and operation of the MCU
Use Case:	Access to memory protection unit, MCU clock generation, MCU power states
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435, RS_Main_00130)

4.9.2 [RS_BRF_01864] AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to digital I/O ports

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide mapping of binary signals to digital I/O ports of the MCU and/or otherwise accessible I/O-ports
Rationale:	Allow clean decoupling of functional processing of signals and accessing their related hardware. This enables portability of application software
Use Case:	Control externally connected TTL hardware components
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435, RS_Main_00130)

4.9.3 [RS_BRF_01872] AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to analog/digital converter ports

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide mapping of discrete signals to analog/digital converter ports
Rationale:	Allow clean decoupling of functional processing of signals and accessing their related hardware. This enables portability of application software
Use Case:	Read-out externally connected analog sensors
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435, RS_Main_00130)

4.9.4 [RS_BRF_01880] AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to pulse-width modulation controlled ports

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to pulse-width modulation controlled ports
Rationale:	Allow clean decoupling of functional processing of signals and accessing their related hardware. This enables portability of application software
Use Case:	Dim interior light
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435, RS_Main_00130)

4.9.5 [RS_BRF_01888] AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to an output compare unit

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to an output compare unit
Rationale:	Allow clean decoupling of functional processing of signals and accessing their related hardware. This enables portability of application software
Use Case:	counter based wave generation
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435, RS_Main_00130)

4.9.6 [RS_BRF_01896] AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to input capture units

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide mapping of I/O signals to input capture units
Rationale:	Allow clean decoupling of functional processing of signals and accessing their related hardware. This enables portability of application software
Use Case:	access to capture control unit, input capture units, counter based frequency measurement
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00435, RS_Main_00130)

4.9.7 [RS_BRF_01904] AUTOSAR microcontroller abstraction shall provide access to hardware timers

⌈

Type:	Valid

Description:	AUTOSAR microcontroller abstraction shall provide access to hardware timers
Rationale:	Allow clean decoupling of functional processing of timer values and accessing their related hardware. This enables portability of application software
Use Case:	High precision time measurement, periodic interrupt generation, alarm clock of ECU State Manager
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435, RS_Main_00130)

4.9.8 [RS_BRF_01912] AUTOSAR microcontroller abstraction shall provide access to SPI

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide access to SPI as a selected I/O bus for external devices
Rationale:	Allow clean decoupling of management and operation of externally connected hardware. This enables portability of the service layer
Use Case:	Connect SPI accessible external memory devices
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435, RS_Main_00130)

4.9.9 [RS_BRF_01920] AUTOSAR microcontroller abstraction shall provide access to communication bus controllers

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide access to communication bus controllers
Rationale:	Allow clean decoupling of management and operation of communication and accessing the related hardware. This enables portability of the service layer
Use Case:	Connect external communication transceivers
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00435, RS_Main_00130)

4.9.10 [RS_BRF_01928] AUTOSAR microcontroller abstraction shall provide access to non-volatile memory hardware

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide access to non-volatile memory hardware
Rationale:	Allow clean decoupling of the usage of persistent data from respective storage hardware. This enables portability of the service layer

Use Case:	Internal or external EEPROM hardware
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00440, RS_Main_00435, RS_Main_00130)

4.9.11 [RS_BRF_01936] AUTOSAR microcontroller abstraction shall provide access to MCU internal and external hardware watchdogs

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide management and handling of MCU internal and external hardware watchdogs
Rationale:	Allow clean decoupling of logical processing from related hardware access. This enables portability of application software
Use Case:	Supervise life signs and correct timing behavior of periodic functionality
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00435, RS_Main_00130)

4.9.12 [RS_BRF_01944] AUTOSAR microcontroller abstraction shall provide access to communication bus watchdog hardware

⌈

Type:	Valid
Description:	AUTOSAR microcontroller abstraction shall provide access to communication bus watchdog hardware
Rationale:	Allow clean decoupling of logical processing from related hardware access. This enables portability of application software
Use Case:	Detect communication timeouts
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00435, RS_Main_00130)

4.9.13 [RS_BRF_01952] AUTOSAR IO Hardware Abstraction shall support standardized modes for connected I/O devices

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall support standardized modes for connected I/O devices. Management has to take into account the application and Basic Software I/O access needs
Rationale:	Hardware is not in use in certain states like start-up, shutdown of the system or in certain fault states
Use Case:	e. g. energy saving states, controlling power state of connected hardware
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00450, RS_Main_00460)

4.9.14 [RS_BRF_01960] AUTOSAR IO Hardware Abstraction shall provide mapping of I/O signals between domain specific and hardware specific units

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall provide mapping of I/O signals between domain specific units and hardware specific representation. This involves e. g. range checking, range and resolution conversion, including signal processing like debouncing or frequency domain filtering of pulse-code modulated signals
Rationale:	Enhances portability of software components between different hardware platforms with different raw representations of the signal on I/O interface level
Use Case:	A/D conversion of an analog signal (voltage) into its corresponding temperature
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00150)

4.9.15 [RS_BRF_01968] AUTOSAR IO Hardware Abstraction shall support edge triggered I/O signals

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall support edge triggered I/O signals. This involves e. g. notification of upper layer functionality about occurrence of an edge and/or measurement edge based metrics (e. g. pulse-width, duty-cycle, pulse period, …)
Rationale:	This is typical for event based systems
Use Case:	Measurement of the period time between two falling or rising edges
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130)

4.9.16 [RS_BRF_01976] AUTOSAR IO Hardware Abstraction shall support level triggered I/O signals

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall support level triggered I/O signals. This involves notification of upper layer functionality if a signal reaches a certain active or inactive level
Rationale:	
Use Case:	Periodically sample a continuous signal
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130)

4.9.17 [RS_BRF_01984] AUTOSAR IO Hardware Abstraction shall support time domain I/O signals

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall support handling of time discrete properties of transient and periodic physical I/O signals
Rationale:	Handling of the absolute value of a physical signal at certain points in time is a common task of signal processing
Use Case:	Analyzing absolute value of an A/D converted signal stream
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130)

4.9.18 [RS_BRF_01992] AUTOSAR IO Hardware Abstraction shall support frequency domain I/O signals

⌈

Type:	Valid
Description:	AUTOSAR IO Hardware Abstraction shall support handling frequency domain properties of periodic physical I/O signals
Rationale:	Handling of frequency domain related properties of a signal is a common task of signal processing
Use Case:	PWM, event counting by input capture units
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00130)

4.9.19 [RS_BRF_02000] AUTOSAR IO Hardware Abstraction shall protect hardware against illegal operation

⌈

Type:	Valid
Description:	Protect hardware against deterioration induced by systematic failure. Switching of digital I/O could e. g. lead to short-circuit or over-loading of external hardware. Protection can be achieved e. g. by monitoring of feedback signals or internal checks for plausibility of input signals
Rationale:	Maintain reliability and durability of the hardware
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00450)

4.10 Security

4.10.1 [RS_BRF_02008] AUTOSAR shall provide mechanisms to protect the system from unauthorized read access

⌈

Type:	Valid
Description:	If considered appropriate, the system (ECU, communication, I/O) and its data have to be protected against unauthorized read access. This typically involves data encryption mechanisms
Rationale:	Secure access to confidential data.
Use Case:	Storage of personal or private data
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170)

4.10.2 [RS_BRF_02016] AUTOSAR shall provide mechanisms to protect the system from unauthorized modification

⌈

Type:	Valid
Description:	If considered appropriate, the system and its data have to be protected against unauthorized modification. This typically involves authentication and signature mechanisms
Rationale:	Secure integrity of data
Use Case:	Prohibit unauthorized modification of an emission control systems. Protect immobilizer code or vehicle identification number inside NV memory after end-of-line programming
Dependencies:	--
Supporting Material:	e. g. Regulation EC 692/2008

⌋(RS_Main_00170)

4.10.3 [RS_BRF_02024] AUTOSAR shall provide mechanisms to protect the system from unauthorized use

⌈

Type:	Valid
Description:	If considered appropriate, the system and its functionality have to be protected against unauthorized activation or use. This typically involves authentication and signature mechanisms
Rationale:	Secure availability of the system by preventing damage or fraudulent use
Use Case:	Homologation directives that require mechanisms to prevent unauthorized use of the vehicle
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170)

4.10.4 [RS_BRF_02032] AUTOSAR security shall allow integration of cryptographic primitives into the cryptographic service manager

⌈

Type:	Valid
Description:	Security goals typically require the involvement of cryptographic principles like data encryption, hash-number computation, key handling. AUTOSAR shall support integration of OEM specific implementations of the related cryptographic primitives through a standardized interface
Rationale:	Prevent code duplication of common primitive cryptographic operations. Allow OEM specific selection of cryptographic algorithms
Use Case:	Block/stream encryption, message authentication, validation/verification of signatures, exchange of private/public key, …
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170)

4.11 Safety

4.11.1 [RS_BRF_02040] AUTOSAR BSW and RTE shall ensure data consistency

⌈

Type:	Valid
Description:	AUTOSAR shall ensure data consistency of internal data of BSW and RTE, and of data which is shared between several modules and especially different cores
Rationale:	Multi-core systems have to provide consistent access to shared data. (e. g. mechanisms for mutual exclusion, atomic sequences, etc.)
Use Case:	Multi-core systems, multi-tasking systems
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00100)

4.11.2 [RS_BRF_02048] AUTOSAR shall support usage of hardware memory protection features to enhance safety

⌈

Type:	Valid
Description:	If adequate memory protection mechanisms are supported by hardware, AUTOSAR shall support the usage of these hardware mechanisms in such a way that memory used by SW-Cs and BSW modules can be protected from illegal or erroneous access
Rationale:	Only if it can be shown that different groups of software components do not interfere, the groups of software components can be evaluated separately with respect to their safety requirements
Use Case:	Combine software components of different ASIL level on the same ECU
Dependencies:	--
Supporting Material:	ISO 26262-6:2011, Annex D (Freedom from interference between software elements)

⌋(RS_Main_00010)

4.11.3 [RS_BRF_00129] AUTOSAR shall support data corruption detection and protection

⌈

Type:	Valid
Description:	AUTOSAR shall check data in RAM and non-volatile memory to detect data corruption where applicable. This can be done by software (e. g. by means of checksums) or by special hardware support (e. g. redundancy controller or parity checking) Protection against hardware faults is assumed to be solved e. g. by standard ECC-correction mechanisms
Rationale:	Enable AUTOSAR to handle its internal data in a safe manner
Use Case:	Requestors of fault-tolerant data protection (RAM-test, flash test) such as: 1. ECU state manager: ECU state data;

	2. DEM, FIM: current errors detected.
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00010)

4.11.4 [RS_BRF_00131] AUTOSAR shall support program flow monitoring

⌈

Type:	Valid
Description:	AUTOSAR shall support logical and temporal program flow monitoring to detect if program flow control is violated. AUTOSAR shall offer support for ensuring that the program flow monitoring mechanisms are working properly
Rationale:	Using flow control to detect if a software components runs wild is an established safety feature Using program flow control to detect if a runnable (or a sequence of runnables) is executed out of order or not at all is a well established safety feature
Use Case:	To detect a defective program sequence. A defective program sequence exists, if the individual elements of a program (for example, software modules, subprograms or commands) are processed in the wrong sequence or period of time, or if the clock of the processor is faulty
Dependencies:	--
Supporting Material:	ISO 26262-5:2011 Annex D, ISO 26262-6:2011

⌋(RS_Main_00010)

4.11.5 [RS_BRF_02056] AUTOSAR OS shall support timing protection

⌈

Type:	Valid
Description:	If configured, AUTOSAR OS shall support to supervise runtime of tasks and interrupts, together with frequency of task and interrupt activation, to detect and react if a task or an interrupt consume more runtime than configured
Rationale:	Systems are usually evaluated based on assumptions concerning runtime and frequency of tasks and interrupts. The violation of these assumptions may lead to the violation of the safety goals
Use Case:	Stop application parts which violate runtime constraints
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010)

4.11.6 [RS_BRF_02064] AUTOSAR shall use hardware communication data integrity mechanisms

⌈

Type:	Valid
Description:	AUTOSAR shall use data integrity mechanisms which are offered by communication hardware such that major fault models described in ISO 26262 are covered

Rationale:	Cover the ISO26262 cases like: - Failure of communication peer - Message corruption - Message delay - Message loss - Unintended message repetition - Resequencing - Insertion of message and - Masquerading
Use Case:	Exchanging of information between elements executed on different ECUs including signals, data, messages, etc. Information can be exchanged using I/O-devices, data busses, etc.
Dependencies:	--
Supporting Material:	ISO 26262-5:2011 Annex D, ISO 26262-6:2011 Annex D

⌋(RS_Main_00010)

4.11.7 [RS_BRF_00110] AUTOSAR shall offer methods to protect safety related data communication against corruption

⌈

Type:	Valid
Description:	All currently supported communication stacks (CAN, LIN, FlexRay, Ethernet) shall have a communication protection that detects corruption of communication. This includes checks whether a signal is received in sequence or not
Rationale:	To detect when data exchanged between different ECUs is corrupted or wrongly routed
Use Case:	Two SW-Cs on two ECUs exchange safety-related data
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010)

4.11.8 [RS_BRF_00113] AUTOSAR shall detect signal time-outs

⌈

Type:	Valid
Description:	AUTOSAR shall provide a mechanism that detects if periodic signals are not exchanged within a defined time interval (time-out). This can be used for detecting errors in the communication system (loss of messages). If a message is coming too late, even if it is correct (correct sequence number, checksum etc.), it shall be considered as an error. The actual handling shall be up to the application (e. g. error report, request resend, …)
Rationale:	Time-outs are commonly used to determine if a communication system is functioning or if an individual ECU is communicating. Failure to receive a message from a particular ECU means loss of information or functionality
Use Case:	The behavior of an anti-skid system might become erroneous if its operation is based on outdated sensor values. The continuous updates of the sensor values can be monitored using a communication watchdog
Dependencies:	--

Supporting Material:	

⌋(RS_Main_00011, RS_Main_00010)

4.11.9 [RS_BRF_00241] AUTOSAR shall support redundant multiple communication links

⌈

Type:	Valid
Description:	AUTOSAR shall support multiple communication links
Rationale:	1. If in a given system there is redundant communication HW (like two independent CAN buses, or one CAN and one FlexRay buses), then to provide fault tolerance, one can use a safety protocol on each channel (with data protected with checksum, address id, counter and timeout for example). This enables the receiver to do e. g. 1oo2 voting (take one of two correct received messages) 2. If one channel completely fails the second channel may be used for reduced functionality communications
Use Case:	Tolerate faults on one of the channels
Dependencies:	
Supporting Material:	--

⌋(RS_Main_00010)

4.12 Libraries

4.12.1 [RS_BRF_02072] AUTOSAR shall provide generic functionality which is in wide use in the automotive domain as libraries

⌈

Type:	Valid
Description:	AUTOSAR shall provide generic algorithms which are in wide use in the automotive domain as libraries
Rationale:	Having common automotive algorithms available as a library implementation to applications reduces code duplication, speeds-up application development and increases implementation reliability
Use Case:	Mathematical libraries, safety and security libraries supporting algorithms
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00410)

4.12.2 [RS_BRF_02080] AUTOSAR libraries shall use C interfaces

⌈

Type:	Valid
Description:	Library functionality shall be accessible via C interfaces with prototypes provided in standard header files. This also includes publication of library specifics types. Accordingly a library can only access other library functionality
Rationale:	Usage of libraries is based on an implementor's design decision and therefore cannot be under control of system configuration. Hence, mechanisms like RTE or services from Basic Software Modules are not available
Use Case:	Making library code accessible for BSW and SW-C
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00220, RS_Main_00410)

4.12.3 [RS_BRF_02088] AUTOSAR library functionality shall be reentrant

⌈

Type:	Valid
Description:	AUTOSAR library functionality shall be reentrant
Rationale:	Libraries must be stateless in order to be accessible in parallel from different layers, tasks or cores of the system. Library functionality is not under state control of the system, and therefore can have neither initialization nor shutdown behavior
Use Case:	Making library code accessible for BSW and SW-C at any time in an operational life cycle
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00410)

4.12.4 [RS_BRF_02096] AUTOSAR shall provide checksum computation of cyclic redundancy check sums as a library

⌈

Type:	Valid
Description:	An AUTOSAR library shall provide standard implementations for computation of cyclic redundancy checks. This includes different implementations for an algorithm favoring either memory or computation time consumption
Rationale:	CRC computation is a common task to ensure integrity of data blocks
Use Case:	Computation of message digests in safe communication
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00410)

4.12.5 [RS_BRF_02104] AUTOSAR shall provide end-to-end protection support as a library

⌈

Type:	Valid
Description:	In order to support safe communication between application software components a library shall be provided that supports implementation of safe communication. This includes checking of signal integrity e. g. by checksums and sequence counters
Rationale:	Support integrity of communication data
Use Case:	Safety-related communication between too ECUs
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00010, RS_Main_00410)

4.12.6 [RS_BRF_02112] AUTOSAR shall support floating point arithmetic functions as a library

⌈

Type:	Valid
Description:	AUTOSAR shall support floating point arithmetic functions as a library
Rationale:	Mathematical computation is common to open and closed-loop control systems. Having a standard set of mathematical functions to implement common control applications reduces code duplication
Use Case:	Integration of control applications from different vendors on one ECU
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00410)

4.12.7 [RS_BRF_02120] AUTOSAR shall support fixed point arithmetic functions as a library

⌈

Type:	Valid
Description:	AUTOSAR shall support fixed point arithmetic functions as a library including extended functions like e. g. filtering, transcendent functions, sorting, etc.
Rationale:	Mathematical computation is common to open and closed-loop control systems. Having a standard set of mathematical functions to implement common control applications reduces code duplication
Use Case:	Integration of control applications from different vendors on one ECU
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00410)

4.12.8 [RS_BRF_02128] AUTOSAR shall provide arithmetic interpolation routines as a library

⌈

Type:	Valid
Description:	Interpolation routines that interpolate in 2D and 3D space shall be available as library functionality to applications
Rationale:	Interpolation between configured interpolation points is a common task in every instrumentation and control system
Use Case:	Adaptable control applications with externally configurable parameters
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00410)

4.12.9 [RS_BRF_02136] AUTOSAR shall provide cryptographic primitives as a library

⌈

Type:	Valid
Description:	AUTOSAR shall provide cryptographic primitives as a library
Rationale:	Basic Software might need to use cryptographic primitives
Use Case:	Authorize access for a jump to bootloader
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170)

4.13 Diagnostic and Error Handling

4.13.1 [RS_BRF_02144] AUTOSAR diagnostic shall provide standardized diagnostic services for external testers

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall provide standardized diagnostic services for externally connected tester hardware. Supported services include e. g. access to internal event memories, execution of diagnostic related services, ECU reset…
Rationale:	Allow repair technicians access to state of vehicle sub-systems
Use Case:	Read and control recorded diagnostic trouble codes
Dependencies:	--
Supporting Material:	ISO 15031-5 (OBD), ISO 14229-1 (UDS)

⌋(RS_Main_00260, RS_Main_00420)

4.13.2 [RS_BRF_02152] AUTOSAR diagnostic shall provide standardized bootloader interaction

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall provide standardized bootloader interaction. This includes to save all relevant information, perform necessary mode changes and a jump to a specific bootloader (e. g. system or OEM bootloader)
Rationale:	Restarting the ECU might be necessary after certain configuration changes
Use Case:	Reflashing parts of ECU memory
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.13.3 [RS_BRF_02160] AUTOSAR diagnostic shall allow external testers to control active functionality of the ECU

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall allow external testers to switch the ECU into different states (e. g. for programming, diagnostic mode) in order to control/deactivate functionality on the ECU
Rationale:	Some diagnostic operations require disabling interfering standard functionality of the ECU
Use Case:	Disabling the ECU functionality during a programming cycle, disable faulty and/or dangerous functionality (like airbag ignition)
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.13.4 [RS_BRF_02168] AUTOSAR diagnostics shall provide a central classification and handling of abnormal operative conditions

⌈

Type:	Valid
Description:	AUTOSAR diagnostics shall provide a central classification of abnormal operative conditions. This includes classification, filtering and debouncing faults from application and Basic Software and handling of set and reset conditions of fault conditions depending on the system state
Rationale:	Ease handling of faults from Basic Software and application software
Use Case:	Classification of hardware errors prior to creating garage related trouble
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00260)

4.13.5 [RS_BRF_02176] AUTOSAR diagnostics shall distinguish between defined abnormal operative conditions and unexpected exceptions from intended behavior

⌈

Type:	Valid
Description:	AUTOSAR diagnostics shall distinguish between defined abnormal operative conditions and exceptions from intended and expected behavior
Rationale:	Faults from systematic errors have to be handled differently than predefined abnormal operative conditions
Use Case:	Ignoring defect sensor value against unknown handling of illegal configuration
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00260)

4.13.6 [RS_BRF_02184] AUTOSAR diagnostics shall provide central storage to document occurrences of fault conditions

⌈

Type:	Valid
Description:	AUTOSAR diagnostics shall provide central storage to document occurrences of fault conditions from Basic Software and application software
Rationale:	Support retrieval of fault conditions by repair technician
Use Case:	Recording a diagnostic trouble code
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.13.7 [RS_BRF_02192] AUTOSAR diagnostic management shall be bus independent

⌈

Type:	Valid
Description:	AUTOSAR diagnostic management shall be bus independent
Rationale:	Keeping Basic Software Modules and application software portable. Note: This does not include the transport protocol used to communicate standardized diagnostic messages. The transport protocol is normally bus-dependent (e. g. ISO 15765-3 Diagnostic communication over CAN, ISO 13400-2 Diagnostic communication over IP, …)
Use Case:	--
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.13.8 [RS_BRF_02200] AUTOSAR diagnostic shall provide external access to internal configuration and calibration data

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall provide external access to ECU internal memory and/or configuration data. This includes post-build configuration by communication with external testers, access to vehicle specific information (like vehicle information number)
Rationale:	Adopt the ECU to varying environmental conditions
Use Case:	End of line configuration/calibration
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00260)

4.13.9 [RS_BRF_02208] AUTOSAR diagnostic shall use authentication mechanisms to secure external access

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall use authentication mechanisms to secure external access
Rationale:	Prevent unauthorized modification or manipulation of configuration or calibration data
Use Case:	Prevent unauthorized tuning
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00170, RS_Main_00260)

4.13.10 [RS_BRF_02216] AUTOSAR diagnostic shall allow runtime degradation of faulty functionality to maintain minimum ECU/vehicle operability

⌈

Type:	Valid
Description:	AUTOSAR diagnostic shall allow runtime degradation of faulty functionality by static configuration of functionality clusters, accessible by application software
Rationale:	Maintain minimum ECU/vehicle operability in case of defect sensor values that inhibit normal performance characteristics but still allows for backup operation
Use Case:	Limp home mode
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00460, RS_Main_00260, RS_Main_00011)

4.14 Test and Debugging

4.14.1 [RS_BRF_02224] AUTOSAR shall support run-time hardware tests

⌈

Type:	Valid
Description:	AUTOSAR shall support mechanisms for scheduling regular tests that intend to detect hardware failure. These checks can be performed e. g. in constant intervals, at idle time or as part of power-on/power-off tests
Rationale:	Assure integrity of hardware
Use Case:	Flash test, RAM test, core test
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00011, RS_Main_00480)

4.14.2 [RS_BRF_02232] AUTOSAR shall support development with run-time assertion checks

⌈

Type:	Valid
Description:	AUTOSAR shall support development with run-time assertion checks that detect e. g. violation of interface contracts or invalid state changes
Rationale:	Early detection of violated interface constraints helps to avoid consecutive failures which might be difficult to trace back to the original fault during development and/or integration
Use Case:	Detect illegal communication channel numbers
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00480)

4.14.3 [RS_BRF_02240] AUTOSAR debugging shall provide relevant internal data of Basic Software to the developer

⌈

Type:	Valid
Description:	AUTOSAR debugging shall provide relevant internal data from RTE and BSW modules to the developer in order to make behavior of the system transparent during run-time debugging. Debugging shall be as far as possible independent of other Basic Software Modules in order to allow each module to act as a debugging target
Rationale:	This might help e. g. to shorten debugging time in the case that source code of a module is for certain reasons not available. Help debugging interaction between on-site integrator and a remote module supplier. Get a trace of internal system behavior immediately before failure
Use Case:	Debugging of a faulty system
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00480)

4.14.4 [RS_BRF_02248] AUTOSAR debugging shall offer methods to influence behavior of a Basic Software Module

⌈

Type:	Valid
Description:	AUTOSAR debugging shall offer methods to influence behavior of a Basic Software Module e. g. by modifying internal data of the target module
Rationale:	Modifying the behavior of a module that does not behave as expected can help the integrator to prove or reject a given hypothesis on the root cause of a failure. This also provides the possibility to inject certain faults to compare system behavior against reported sporadic failures
Use Case:	Debugging of a faulty system, fault injection
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00480)

4.14.5 [RS_BRF_02256] AUTOSAR debugging shall support runtime and post mortem debugging

⌈

Type:	Valid
Description:	Interpretation of AUTOSAR debugging data shall be possible during runtime and post mortem
Rationale:	Analysis of module internal states gives valuable insight on a faulty system before and after actual system failure
Use Case:	Debugging of a faulty or even failed system, e. g. during start-up
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00480)

4.14.6 [RS_BRF_02264] AUTOSAR shall support XCP for setting measurement and calibration data

⌈

Type:	Valid
Description:	AUTOSAR shall support automotive standards for setting measurement and calibration data like XCP
Rationale:	XCP provides bus independent access to measurement and calibration data during development, prototyping and test of ECUs
Use Case:	Unknown
Dependencies:	--
Supporting Material:	ASAM MCD-1 XCP

⌋(RS_Main_00420, RS_Main_00480)

4.14.7 [RS_BRF_02272] AUTOSAR shall offer tracing of application software behavior

⌈

Type:	Valid
Description:	AUTOSAR shall offer configurable tracing of application software behavior by recording RTE activity and logging events from the Basic Software Modules DET and DEM in order to be able to consolidate fault reports with supervised application behavior
Rationale:	Provide insight on actions taken inside SW-Cs during development and production phase of an ECU
Use Case:	Debugging support, model based test, test automation
Dependencies:	--
Supporting Material:	

⌋(RS_Main_00480)

4.15 Integration and Migration

4.15.1 [RS_BRF_02280] AUTOSAR shall support non-AUTOSAR BSW modules

⌈

Type:	Valid
Description:	AUTOSAR shall define under which conditions code for which no AUTOSAR-supplied BSW module specification exists can run inside the AUTOSAR BSW and interact with RTE, software components and AUTOSAR-defined BSW modules Note: this functionality is often called Complex Driver (CDD)
Rationale:	AUTOSAR can never be complete: there may be functionality which is rarely used such that it is not worthwhile to create an AUTOSAR specification, or functionality which is so new that an AUTOSAR specification cannot exist yet. For such cases, rules shall exist which allow non-AUTOSAR functionality to be integrated in an AUTOSAR-ECU
Use Case:	MCU-to-MCU communication via shared memory
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00190)

4.15.2 [RS_BRF_02288] Generic interfaces in AUTOSAR shall support Complex Drivers

⌈

Type:	Valid
Description:	In case an AUTOSAR BSW module supports multiple underlying BSW modules of the same interface type, the interface names shall be generic, and interfaces and configuration shall be designed to be used by non AUTOSAR-defined BSW modules (CDDs) as well
Rationale:	Acting otherwise would without any benefit seriously harm the possibilities to integrate CDDs
Use Case:	Introduction of a new communication bus in the AUTOSAR architecture using CDDs
Dependencies:	--
Supporting Material:	--

⌋(RS_Main_00210)

5 Not applicable requirements

[RS_BRF_NA_1] ⌈This requirement references all process related main requirements which are not applicable for the Basic AUTOSAR Feature list.⌋(RS_Main_00030, RS_Main_00490, RS_Main_00290, RS_Main_00350)

[RS_BRF_NA_2] ⌈This requirement references all non-functional main requirements which are not applicable for the Basic AUTOSAR Feature list. ⌋(RS_Main_00120, RS_Main_00270)

[RS_BRF_NA_3] ⌈This requirement references all methodology related main requirements which are currently not applicable to the current subset of BSW and RTE features. A future version of this document, with methodology related features integrated, is expected to implement them.⌋(RS_Main_00160, RS_Main_00180, RS_Main_00300, RS_Main_00080, RS_Main_00310, RS_Main_00320, RS_Main_00340, RS_Main_00360, RS_Main_00250, RS_Main_00251)

6 References

[GLOSSARY] AUTOSAR Glossary, AUTOSAR_TR_Glossary.pdf

[ISO 10681] Road vehicles -- Communication on FlexRay

[ISO 11898] Road vehicles -- Controller area network (CAN)

[ISO 13400] Road vehicles -- Diagnostic communication over Internet Protocol (DoIP)

[ISO 14229] Road vehicles -- Unified diagnostic services (UDS)

[ISO 15031] Road vehicles -- Communication between vehicle and external equipment for emissions-related diagnostics

[ISO 15765] Road vehicles -- Diagnostic communication over Controller Area Network (DoCAN, UDS on CAN)

[ISO 17356] Road vehicles -- Part 3: OSEK/VDX Operating System (OS)

[ISO 26262] Road vehicles -- Functional safety

[ISO 27145] Road vehicles -- Implementation of WWH-OBD communication requirements

[RS_MAIN] AUTOSAR Main Requirements, AUTOSAR_RS_Main.pdf

[SAE J1939] Serial Control and Communications Heavy Duty Vehicle Network

[TPS_STDT] AUTOSAR Standardization Template, AUTOSAR_TPS_StandardizationTemplate.pdf