Document Title	Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	986

Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R22-11

Document Change History

Date	Release	Changed by	Description
2022-11-24	R22-11	AUTOSAR Release Management	• renamed safety goal to top level safety requirement • added safety need for recovery upon failure • added support to include extracted requirements from component requirement specification
2021-11-25	R21-11	AUTOSAR Release Management	• add classic platform requirements chapter • add requirements for CP WDG, CP OS, CP E2E • rework top level safety requirements structure • add TLSR RS_SAF_00006 • update PHM, EM, SM requirements • update functional safety requirements to be AUTOSAR Platform and Foundation
2020-11-30	R20-11	AUTOSAR Release Management	• Initial release • Functional safety requirements for the AUTOSAR Adaptive Platform • Technical safety requirements for PHM, EM, SM, OS, PER, CM and UCM

Disclaimer

Contents

1 Scope of Document 4

2 How to Read This Document 5

2.1 Document Conventions 5
2.2 Conventions used 5
2.2.1 Requirement Identifier Coding 6

3 Acronyms and abbreviations 8

4 Requirements Specification 9

4.1 Top Level Safety Requirements 9
4.2 Functional Safety Requirements 11
4.3 Technical Safety Requirements 17
4.3.1 AUTOSAR Foundation 18
4.3.1.1 Health Monitoring (HM) 18
4.3.2 AUTOSAR AdaptivePlatform 19
4.3.2.1 Functional Cluster: Platform Health Management (PHM) 19
4.3.2.2 Functional Cluster: Execution Management (EM) . . 20
4.3.2.3 Functional Cluster: State Management (SM) 23
4.3.2.4 Operating System (OS) 24
4.3.2.5 Functional Cluster: Persistency (PER) 26
4.3.2.6 Functional Cluster: Communication Management (CM) 27
4.3.2.7 Functional Cluster: Update and Configuration Management (UCM) 29
4.3.3 AUTOSAR ClassicPlatform 31
4.3.3.1 Basic Software: Watchdog Manager (WDGM) 31
4.3.3.2 Basic Software: Operating System (OS) 33
4.3.3.3 E2E Protection 34

5 Requirements Tracing 36

6 References 38

1 Scope of Document

This document specifies safety requirements on the AUTOSAR Platform, the AUTOSAR Adaptive Platform in particular. This document elaborates the high level safety requirements written in RS_Main. It makes use of the intended functionality described in EXP_PlatformDesign document. The functional safety requirements are derived from the top level safety requirements and hazards mentioned in EXP_SafetyOverview. Technical safety requirements towards the AUTOSAR functional cluster and safety relevant applications are derived from the functional safety requirements.

The AUTOSAR Classic Platform is not in scope.

No ASIL Ratings

The AUTOSAR consortium, especially the AUTOSAR Adaptive Platform Working Groups are only providing an architecture definition, descriptions of the functional blocks and a *proof of concept* implementation, it is not possible to add an ASIL rating to any requirement in this scope as described in ISO26262[1].

2 How to Read This Document

This document contains functional safety requirements which are generic and do not mention specific solutions/components of AUTOSAR. The technical safety requirements are then derived from functional safety requirements, which mention the specific responsibilities of AUTOSAR components. Each requirement has its unique identifier starting with the prefix "RS_SAF_" (for "Safety Requirement").

Technical Safety Requirements are partly extracted from the dedicated target component requirement specification and will therefore not have the prefix "RS_SAF_". Not all technical requirements have been consolidated yet, the goal is to have all technical requirements being part of the target requirements specification and only included here for to have a complete safety requirement catalog.

2.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([2]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([2]).

2.2 Conventions used

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 follows.

Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the adjective "LEGALLY REQUIRED", means that the definition is an absolute requirement of the specification due to legal issues.
- MUST NOT: This phrase, or the phrase "MUST NOT", means that the definition is an absolute prohibition of the specification due to legal issues.
- SHALL: This phrase, or the adjective "REQUIRED", 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.
- SHOULD: This word, or the adjective "RECOMMENDED", means 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", means 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, SHALL 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, SHALL be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2.1 Requirement Identifier Coding

The unique identifier for safety requirements shall consist of

- a document identifier
- an identifier to distinguish functional safety requirements and technical safety requirements
- an identifier to identify a target component (either a Functional Cluster in the AUTOSAR Adaptive Platform or a Basic Software Component in the AUTOSAR Classic Platform)
- a requirement number

The coding pattern used in this requirements specification is `RS_SAF_<Z><YY><XX>`, where

`Z` is a single digit number, describing whether the requirement is a

`0` safety goal or top level safety requirement functional safety requirement, where

`YY` *is reserved*

`XX` is a double digit number

`1` functional safety requirement for the AUTOSAR Adaptive Platform, where

`YY` *is reserved*

`XX` is a double digit number

2 technical safety requirement for the AUTOSAR Adaptive Platform, where

YY is a double digit number, describing whether the requirement addresses

00 *reserved*

11 Platform Health Management (PHM)

12 Execution Management (EM)

13 State Management (SM)

14 Operating System (OS)

15 Persistency (PER)

16 Communication Management (CM)

17 Update and Configuration Management (UCM)

and

XX is a double digit number

3 technical safety requirement for the AUTOSAR Classic Platform, where

YY is a double digit number, describing whether the requirement addresses

00 *reserved*

11 Watchdog Manager (WDGM)

12 Operating System (OS)

13 E2E Protection (E2E)

and

XX is a double digit number

4-9 reserved for future use

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to RS_Safety that are not included in the AUTOSAR Glossary [3].

Abbreviation / Acronym:	Description:
PHM	Platform Health Management
EM	Execution Management
SM	State Management
OS	Operating System
PER	Persistency
CM	Communication Management
UCM	Update and Configuration Management
S2S	Signal to Service
SG	Safety Goal

Table 3.1: Acronyms and Abbreviations

4 Requirements Specification

This chapter contains top level safety requirements for AUTOSAR in 4.1. Functional safety requirements in 4.2 are derived from these requirements. The sub-chapter 4.3 contains technical safety requirements which are derived from the functional safety requirements.

4.1 Top Level Safety Requirements

[RS_SAF_00001]{DRAFT} **Safe Execution** ⌈

Description:	AUTOSAR shall provide supporting mechanisms to monitor the control flow and manage the execution order of multiple applications with mixed safety criticality.
Rationale:	To ensure freedom from interference with respect to timing [1] and data processing.
AppliesTo:	FO
Supporting Material:	ISO26262 [1]

⌋(*RS_Main_00010*, *RS_Main_00011*, *RS_Main_00012*, *RS_Main_00030*)

[RS_SAF_00002]{DRAFT} **Safe Configuration** ⌈

Description:	AUTOSAR shall provide mechanisms to support correct configuration during the entire driving cycle of the vehicle.
Rationale:	AUTOSAR needs to provide measures and mechanisms to keep the configuration consistent through out the whole driving cycle of the vehicle.
AppliesTo:	FO
Supporting Material:	ISO 26262 [1]

⌋(*RS_Main_00010*, *RS_Main_00011*, *RS_Main_00012*, *RS_Main_00030*)

[RS_SAF_00003]{DRAFT} **Safe Update or Safe Upgrade** ⌈

Description:	AUTOSAR shall provide mechanisms to support correct update and upgrade of multiple platform and non-platform applications with mixed criticality.
Rationale:	AUTOSAR supports updatability during the life cycle of the machine and therefore the platform is responsible to ensure that these updates are performed correctly and safe.
AppliesTo:	FO
Supporting Material:	ISO 26262 [1]

⌋(*RS_Main_00010*, *RS_Main_00011*, *RS_Main_00012*, *RS_Main_00030*, *RS_Main_-00150*)

[RS_SAF_00004]{DRAFT} **Safe Exchange of Information** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe exchange (transmission and reception) of information between safety critical applications.
Rationale:	In a vehicle several ECUs with several software components are interrelating with each other to fulfill a goal or functionality. AUTOSAR provides standardized interfaces and mechanisms to achieve safe communication between these components. Safe communication with elements outside of the vehicle is also in scope.
AppliesTo:	FO
Supporting Material:	ISO 26262 [1]

⌋*(RS_Main_00010, RS_Main_00011, RS_Main_00012, RS_Main_00030)*

[RS_SAF_00005]{DRAFT} **Detection of Data Corruption** ⌈

Description:	AUTOSAR shall provide mechanisms to detect faults and failures while processing data, communicating with other systems or system elements.
Rationale:	Mechanisms to detect faults and failures are required to achieve higher safety ratings and increase product quality. A list of potential failures is described in EXP_SafetyOverview [4] and ISO 26262 [1]. Incorrect specification or configuration is a potential source of failure.
AppliesTo:	FO
Supporting Material:	ISO 26262 [1]

⌋*(RS_Main_00010)*

[RS_SAF_00006]{DRAFT} **Safe Storage** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe storage for applications.
Rationale:	Many applications need to store and retrieve data from persistent or volatile memory. If the Application is safety critical, data elements need to be identified correctly and the data itself shall be checked to ensure that it has not been altered.
AppliesTo:	FO
Supporting Material:	ISO 26262 [1]

⌋*(RS_Main_00010)*

[RS_SAF_00007]{DRAFT} **Recovery upon failure** ⌈

Description:	AUTOSAR shall Monitor, detect and provide means to react on detectable failures.
Rationale:	AUTOSAR expected to be capable of • restarting applications in case of failures • restarting a machine/ECU in case of failures • recover last known configuration in case of update failures
AppliesTo:	CP, AP
Supporting Material:	–

⌋*(RS_Main_00010)*

4.2 Functional Safety Requirements

[RS_SAF_10001]{DRAFT} **AUTOSAR shall provide mechanisms to support safe initialization of software components.** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe initialization of software components.
Rationale:	Safe initialization of the underlying hardware and the AUTOSAR Adaptive Platform functional cluster and services and the application software is required to ensure intended functionality.
Use Case:	SUC_02
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋*(RS_SAF_00001, RS_SAF_00002)*

[RS_SAF_10002]{DRAFT} **AUTOSAR shall provide mechanisms to support safe verification mechanisms of platform basic software modules, functional-clusters, software components, applications, services and their respective configuration data.** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe verification mechanisms of platform basic software modules, functional-clusters, software components, applications, services and their respective configuration data.
Rationale:	Due to the random hardware failures in the memory unit the data integrity is required to be verified to ensure no loss of data has occurred over time during operation, stand-by or powered off and has not been tampered with. Note: Not with respect to cybersecurity.
Use Case:	SUC_02, SUC_06

▽

△

AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00002*, *RS_SAF_00003*)

[RS_SAF_10005]{DRAFT} **AUTOSAR shall provide mechanisms to support safe shutdown and termination of applications, software components, basic software modules and services.** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe shutdown and termination of applications, software components, basic software modules and services.
Rationale:	Before termination of applications and services and/or shut-down of the AUTOSAR Adaptive Platform or the whole ECU, the dependent applications have to be terminated properly in the right order to prevent conflicts or failures or unexpected behavior. Ensure safe degradation, fault evacuation and fault containment.
Use Case:	SUC_01, SUC_06
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00001*, *RS_SAF_00003*)

[RS_SAF_10006]{DRAFT} **AUTOSAR shall provide mechanisms to support safe transition of states in a basic software module, software component, application or service life cycle.** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe transition of states in a basic software module, software component, application or service life cycle.
Rationale:	AUTOSAR Adaptive Platform is responsible for managing and monitoring the internal states of the application.
Use Case:	SUC_01, SUC_06
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋()

[RS_SAF_10008]{DRAFT} **AUTOSAR shall provide mechanisms to support safe resource management for the AUTOSAR Adaptive Platform functional-clusters,**

applications and services and AUTOSAR Classic Platform basic software modules and software components. ⌈

Description:	AUTOSAR shall provide mechanisms to support safe resource management for the AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.
Rationale:	The functional clusters, applications and services of the AUTOSAR Adaptive Platform shall be ensured with adequate resources and availability to that resource in the expected time with sufficient freedom from interference. No unexpected or unhandled exception shall prevent access or delay access to a required and properly managed and authorized resource. Resources are - among other - CPU, runtime, memory consumption, net bandwidth, peripherals (like ADC, DAC, Timer) . . .
Use Case:	SUC_01
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00001*, *RS_SAF_00002*, *RS_SAF_00004*)

[RS_SAF_10014]{DRAFT} **AUTOSAR shall provide an interface to support safe communication for basic software module, software component, application or services.** ⌈

Description:	AUTOSAR shall provide an interface to support safe communication for basic software module, software component, application or services.
Rationale:	In a vehicle several ECUs with several software components are interrelating with each other to fulfill a goal or functionality. AUTOSAR Adaptive Platform provides standardized interfaces and mechanisms to achieve safe communication between these components. Safe communication with elements outside of the vehicle is also in scope.
Use Case:	SUC_03, SUC_04, SUC_05
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00004*)

[RS_SAF_10027]{DRAFT} **AUTOSAR shall provide mechanisms to prevent the loss of a valid configuration.** ⌈

Description:	AUTOSAR shall provide mechanisms to prevent the loss of a valid configuration on either machine or vehicle level.
Rationale:	AUTOSAR Adaptive Platform should provide mechanisms to switch back to the latest working configuration

▽

△

Use Case:	SUC_02, SUC_06
AppliesTo:	CP,AP
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00002*, *RS_SAF_00007*)

[RS_SAF_10028]{DRAFT} **AUTOSAR shall provide mechanisms to support dependable scheduling of AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.** ⌈

Description:	AUTOSAR shall provide mechanisms to support dependable scheduling of AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.
Rationale:	Dependable scheduling is required to ensure the proper time-allocation for all the available functional-clusters, applications and services.
Use Case:	SUC_01
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00001*, *RS_SAF_00002*)

[RS_SAF_10030]{DRAFT} **AUTOSAR shall provide mechanisms to support safe program execution.** ⌈

Description:	AUTOSAR shall provide mechanisms to support safe program execution.
Rationale:	The AUTOSAR Adaptive Platform shall offer flow monitoring mechanisms to detect and ensure that the intended program flow of functional-clusters and services as well as for user-applications and user-services is not violated.
Use Case:	SUC_01
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00001*)

[RS_SAF_10031]{DRAFT} **AUTOSAR shall provide mechanisms to detect program execution time violation** ⌈

Description:	AUTOSAR shall provide mechanisms to detect program execution time violation
Rationale:	All the timing constraints of the functional-clusters, applications and services need to be supervised and monitored.
Use Case:	SUC_01, SUC_06
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00001*)

[RS_SAF_10037]{DRAFT} **AUTOSAR shall provide mechanisms to prevent unintended alteration of data.** ⌈

Description:	AUTOSAR shall provide mechanisms to prevent unintended alteration of data.
Rationale:	To achieve freedom from interference in systems running applications with mixed safety criticality, protection of data against unintended alteration is required.
Use Case:	SUC_06
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00002*, *RS_SAF_00003*, *RS_SAF_00004*)

[RS_SAF_10038]{DRAFT} **AUTOSAR shall provide mechanisms to support that the safety relevant software is only updated/upgraded in a state that cannot cause a hazardous situation.** ⌈

Description:	AUTOSAR shall provide mechanisms to support that the safety relevant software is only updated/upgraded in a state that cannot cause a hazardous situation.
Rationale:	The update of safety critical application should be done when the car is stationary and at a safe location e.g. a parking garage.
Use Case:	SUC_02
AppliesTo:	FO
Dependencies:	–
Supporting Material:	–

⌋(*RS_SAF_00003*)

[RS_SAF_10039]{DRAFT} **AUTOSAR shall support mechanisms to detect unintended alteration of data.** ⌈

Description:	There shall be a safety mechanism that detects communication errors. The mechanism shall be fully built-in in AUTOSAR (including AUTOSAR configuration and corresponding AUTOSAR basic software module). There shall be a support for all currently supported communication stacks (CAN, LIN, FlexRay, Ethernet).
Rationale:	To ensure safe data exchange between software components that fulfills ISO 26262-6:2018 D.2.4, while using a QM communication stack. D.2.4 defines following failure modes of the exchange of information: • • repetition of information; • loss of information; • delay of information; • insertion of information; • masquerade or incorrect addressing of information; • incorrect sequence of information; • corruption of information; • asymmetric information sent from a sender to multiple receivers; • information from a sender received by only a subset of the receivers; • blocking access to a communication channel
Use Case:	SW-Cs or Adaptive Applications on different ECUs, Machines or Partitions exchange safety related data, using QM communication stack
AppliesTo:	CP, AP
Dependencies:	–
Supporting Material:	ISO 26262-6:2018 D.2.4[1]

⌋*(RS_SAF_00002, RS_SAF_00003, RS_SAF_00004)*

[RS_SAF_10040]{DRAFT} **AUTOSAR shall support data recovery mechanisms.** ⌈

Description:	AUTOSAR shall support data recovery mechanisms
Rationale:	Applications want to recover altered data.
Use Case:	SUC_06
AppliesTo:	CP, AP
Dependencies:	–
Supporting Material:	ISO 26262[1]

⌋*(RS_SAF_00006)*

[RS_SAF_10041]{DRAFT} **AUTOSAR shall allow integrators to select and configure the set of safety mechanisms to detect communication faults.** ⌈

Description:	Based on individual safety concepts, AUTOSAR integrators need to individually configure the required mechanism to fulfill the safety requirements.
Rationale:	The AUTOSAR Platform is designed to be used in various applications. It is possible that for specific applications, a particular type of fault will not occur. Therefore, it is reasonable to have the configurability such that integrators may freely select the set of mechanisms to be deployed.
Use Case:	A hi-level design change or new information requires a different communication protection mechanism. An integrator can select the proper protection by changing the manifest or description file.
AppliesTo:	CP, AP
Dependencies:	–
Supporting Material:	–

⌋*(RS_SAF_00004, RS_SAF_00005)*

[RS_SAF_10042]{DRAFT} **AUTOSAR shall provide mechanisms to detect time synchronization violations.** ⌈

Description:	AUTOSAR shall provide mechanisms to detect time synchronization violations.
Rationale:	Time synchronization is a critical functionality for a distributed system where functions are deployed and data is acquired asynchronously in various machines within a network and have to work collaboratively.
Use Case:	A sender is adding a timestamp within a critical message and the receiver is 'running behind' and cannot detect the message delay if time-synchronization is not working properly An important information from a sensor needs to be timestamped so that the information can be processed and fused with other data sources within the time domain.
AppliesTo:	CP, AP
Dependencies:	–
Supporting Material:	–

⌋*(RS_SAF_00004)*

4.3 Technical Safety Requirements

Some of the following requirements are extracted from the dedicated requirements specifications, the extracted requirements are not named *RS_SAF*. The source document is mentioned within the chapter. The requirements named *RS_SAF* are to be consolidated with the corresponsing target specification, so that this chapter does not contain any own requirements in the upcoming releases anymore.

4.3.1 AUTOSAR Foundation

4.3.1.1 Health Monitoring (HM)

[RS_HM_09125]{DRAFT} **Health Monitoring shall provide an Alive Supervision** ⌈

Description:	Health Monitoring shall check if the frequency of reaching a given Checkpoint in a Supervised Entity matches specified limits.
Rationale:	To detect if a periodic function is executed periodically according to specification/design.
AppliesTo:	CP, AP
Use Case:	A safety critical application with alive supervision get stuck at some point in time during execution. HM detects that the supervised application is not alive.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340, RS_SAF_10031)*

[RS_HM_09222]{DRAFT} **Health Monitoring shall provide a Logical Supervision** ⌈

Description:	Health Monitoring shall check if the sequence of Checkpoints in a Supervised Entity at runtime is the same as the one that is specified. This shall include: • start of if/else branch (decision node): exactly one of the code branches shall be entered, the choice is runtime-specific depending on logical condition • end of if/else branch (merge node): exactly one of the branches shall be reached so that the join is performed • fork of the flow into concurrent execution (fork node): all concurrent branches shall be entered • join of the flow of concurrent execution (join node): all concurrent branches shall be reached so that the join is performed.
Rationale:	To detect if the sequence in the execution is the same as specified/designed.
AppliesTo:	CP, AP
Use Case:	Supervision of any software components: application software components or platform components (e.g. execution manager, state manager).
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340, RS_SAF_10005, RS_SAF_10006, RS_SAF_10030)*

[RS_HM_09235]{DRAFT} **Health Monitoring shall provide a Deadline Supervision** ⌈

Description:	Health Monitoring shall check if the elapsed time between two Checkpoints is within the specified min and max limits, including the detection if the second Checkpoint never arrives.
Rationale:	To detect timeouts or loss of deadlines.
AppliesTo:	CP, AP
Use Case:	A safety critical application is developed to reach specific checkpoints in a defined time window and is suddenly not behaving as intended. PHM detects the violation.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340, RS_SAF_-10031)*

4.3.2 AUTOSAR AdaptivePlatform

4.3.2.1 Functional Cluster: Platform Health Management (PHM)

[RS_PHM_00115]{DRAFT} **If supervision of State Management fails then Platform Health Management shall trigger a watchdog reset.** ⌈

Description:	
Rationale:	State Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	SM is managing a safety critical application. Supervision of SM fails and is detected by PHM. PHM shall trigger a watchdog reset.
Dependencies:	SM
Supporting Material:	

⌋*(RS_SAF_10006, RS_SAF_10030, RS_SAF_10005)*

[RS_PHM_00116]{DRAFT} **If supervision of Execution Management fails then Platform Health Management shall trigger a watchdog reset.** ⌈

Description:	
Rationale:	Execution Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	EM is managing safety critical applications and supervision of EM fails and is detected by PHM. PHM shall trigger a watchdog reset.

▽

△

Dependencies:	EM
Supporting Material:	

⌋(*RS_SAF_10006*, *RS_SAF_10030*, *RS_SAF_10005*)

[RS_PHM_00117]{DRAFT} **Platform Health Management shall notify State Management in case an AUTOSAR Adaptive Platform functional cluster, application or service other than Execution Management and State Management fails.** ⌈

Description:	Platform Health Management shall notify State Management in case an AUTOSAR Adaptive Platform functional cluster, application or service other than Execution Management and State Management fails.
Rationale:	Recovery actions are coordinated in SM, the failures shall be reported to SM except if SM or EM themselves fail.
Use Case:	PHM supervises a safety critical application. This application fails. PHM detects the issue and reports to SM.
Dependencies:	–
Supporting Material:	

⌋(*RS_SAF_10005*, *RS_SAF_10006*)

4.3.2.2 Functional Cluster: Execution Management (EM)

[RS_EM_00002]{DRAFT} **`Execution Management` shall set-up one process for the execution of each `Modelled Process`.** ⌈

Description:	For each instance of an `Executable`, `Execution Management` shall allocate one POSIX process. Furthermore process specific properties (like priority, scheduling policy and access rights) shall be assigned based on the `Execution Manifest`.
Rationale:	Isolation of `Executable` instances from each other.
Use Case:	Safety and security related `Applications` require isolation.
Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_00010*, *RS_Main_00049*, *RS_Main_00080*, *RS_Main_00320*, *RS_Main_00150*, *RS_Main_00420*, *RS_SAF_10037*)

[RS_EM_00005]{DRAFT} **`Execution Management` shall support the configuration of OS resource budgets for `process` and groups of `processes`.** ⌈

Description:	Based on the `Execution Manifest`, `Execution Management` shall allocate OS resources to the `process`. The allocation shall be possible for single `process` and groups of `processs`.
Rationale:	Real-time guarantees shall be defined
Use Case:	Like `cgroups` (based on containers which contain one or more processes) and `ulimit`.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00002, RS_Main_00010, RS_Main_00106, RS_Main_00340, RS_Main_00150, RS_SAF_10008)*

[RS_EM_00008]{DRAFT} **`Execution Management` shall support the binding of all threads of a given process to a specified set of processor cores.** ⌈

Description:	`Execution Management` shall allow the binding of threads to specific set of processor cores based on configuration in the `Execution Manifest`. The binding granularity shall be at process level.
Rationale:	Mechanism to influence load balancing, reaction times, and latencies.
Use Case:	Assign two parallel threads to two processor cores to achieve true parallelism.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00010, RS_Main_00050, RS_Main_00106, RS_Main_00320, RS_Main_00501, RS_Main_00150, RS_SAF_10008)*

[RS_EM_00009]{DRAFT} **`Execution Management` shall ensure it is the sole entity starting `processes`.** ⌈

Description:	`Execution Management` is responsible for starting child `processes` and shall prevent such child `processes` from directly starting other processes.
Rationale:	`Execution Management` needs full control of starting applications to ensure required isolation of temporal and spatial properties. Only `Execution Management` shall start `processes`.
Use Case:	Segregation between applications with different safety and/or security properties.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00010, RS_Main_00011, RS_Main_00049, RS_Main_00150, RS_SAF_10001, RS_SAF_10008)*

[RS_EM_00050]{DRAFT} **`Execution Management` shall perform `Machine`-wide coordination of `processes`.** ⌈

Description:	`Execution Management` shall provide an API for a `process` to register its activities for being able to coordinate their execution.
Rationale:	Coordinated scheduling of activities across `Executables`.
Use Case:	Usage of computation resources within the running `processes` shall be managed in the `Machine` to ensure that activities can be coordinated across `processes`. Registration enables `Execution Management` to form the necessary `Machine`-wide view for the coordination.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00460, RS_SAF_10008)*

[RS_EM_00053]{DRAFT} **`Execution Management` shall provide APIs to the `process` to support deterministic redundant execution of `processes`.** ⌈

Description:	`Execution Management` shall provide APIs to support deterministic redundant execution of `processes`.
Rationale:	High ASIL systems require safety mechanism like software lockstep to be implemented on non-automotive grade microprocessors. The redundant execution shall guarantee deterministic, i.e. reproducible results.
Use Case:	Redundant execution of activities to implement software lockstep
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00010, RS_Main_00501, RS_SAF_10028)*

[RS_EM_00113]{DRAFT} **`Execution Management` shall support time-triggered execution.** ⌈

Description:	`Execution Management` shall facilitate time-triggered periodic execution.
Rationale:	Algorithms in processes can be time-triggered. The OS needs to provide mechanisms to allow the time-triggered execution of applications. The triggers need to contain at least external timers, but are not limited to.
Use Case:	Redundant execution of activities to implement software lockstep
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00010, RS_Main_00501, RS_SAF_10028)*

[RS_EM_00151]{DRAFT} **Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.** ⌈

Description:	Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.
Rationale:	EM manages process instantiation and termination of all the processes and therefore needs to be developed and executed according to the same safety standards as the highest rated safety application managed by EM in the system
AppliesTo:	AP
Use Case:	An ASIL C, B and QM Application is running on the adaptive Platform. EM shall execute the ASIL C, B and the QM application, therefore EM shall be implemented with an ASIL C.
Dependencies:	EM
Supporting Material:	–

⌋*(RS_SAF_10001)*

4.3.2.3 Functional Cluster: State Management (SM)

[RS_SM_00600]{DRAFT} **State Management shall be implemented at least according to the highest safety integrity level from any process that is managed by State Management.** ⌈

Description:	State Management shall be implemented at least according to the highest safety integrity level from any process that is managed by State Management.
Rationale:	SM manages state changes and recovery actions of all the processes and therefore needs to be developed and executed according to the same safety standards as the highest rated safety application managed by SM in the system
AppliesTo:	AP
Use Case:	An ASIL C, B and QM Application is running on the adaptive Platform. SM shall manage the ASIL C, B and the QM application, therefore SM shall be implemented with an ASIL C.
Dependencies:	SM
Supporting Material:	–

⌋*(RS_SAF_10001)*

[RS_SM_00601]{DRAFT} **State Management shall coordinate recovery actions.** ⌈

Description:	State Management shall coordinate recovery actions.
Rationale:	State Management is a central functional cluster to which Platform Health Management reports supervision failures and State Management decides which recovery action (e.g. functional group state change, notification to a safe application or even ECU reset) should be triggered.
AppliesTo:	AP
Use Case:	PHM supervises a safety critical application. This application fails. PHM detects the issue and reports to SM. SM coordinates the error recovery actions.
Dependencies:	SM, PHM
Supporting Material:	–

⌋*(RS_SAF_10005, RS_SAF_10006)*

4.3.2.4 Operating System (OS)

[RS_SAF_21401]{DRAFT} **The OS shall support a mechanism that prevents starvation of applications or processes on the basis of CPU usage (under the respect of available resources).** ⌈

Description:	The OS shall support a mechanism that prevents starvation of applications or processes on the basis of CPU usage (under the respect of available resources).
Rationale:	To achieve freedom from interference it is necessary to prevent processes from being adversely affected by other processes that are consuming of excessive resources.
Use Case:	A QM application and a ASIL B application are executed on the same core. OS ensures the defined amount of execution time for safety relevant application.
AppliesTo:	AP
Dependencies:	OS
Supporting Material:	–

⌋*(RS_SAF_10008, RS_SAF_10028, RS_SAF_10031)*

[RS_SAF_21402]{DRAFT} **The OS shall support resource reservation for memory in the interval [min,max]. If max is not specified it shall be considered as unlimited.** ⌈

Description:	The OS shall support resource reservation for memory in the interval [min,max]. If max is not specified it shall be considered as unlimited.
Rationale:	To achieve freedom from interference it is necessary to prevent processes from adversely affecting other processes, through consumption of excessive resources. To this end the OS mechanisms to configure minimum guarantees on available memory are necessary. Optionally a maximum can be configured - if not specified the process can consume all memory that is not otherwise reserved.
Use Case:	A QM application and a ASIL application are executed on the same machine. OS is ensuring all applications are only getting the defined amount of memory allocated.
AppliesTo:	AP
Dependencies:	OS
Supporting Material:	–

⌋*(RS_SAF_10008)*

[RS_SAF_21403]{DRAFT} **Operating System shall ensure that only allowed memory accesses are made.** ⌈

Description:	Operating System shall ensure that only allowed memory accesses are made.
Rationale:	To achieve freedom from interference it is necessary to prevent processes from adversely affected other processes. Access to private memory which is reserved for a process shall be protected against un-allowed accesses from other processes.
Use Case:	A QM application and a ASIL application are executed on the same machine. OS prevents the QM application from changing the memory assigned to the safety critical application.
AppliesTo:	AP
Dependencies:	OS
Supporting Material:	–

⌋*(RS_SAF_10008)*

4.3.2.5 Functional Cluster: Persistency (PER)

[RS_PER_00008]{DRAFT} **Persistency shall support detection of data corruption in persistent memory** ⌈

Description:	Persistency shall support detection of data corruption in persistently stored data. The corruption may be caused by systematic or random failures. To be able to detect corrupted data, some redundancy is needed, which can be anything from a checksum to a full copy. The actual mechanisms and the granularity of redundancy are subject to configuration.
Rationale:	Applications need to be sure to read valid data.
Use Case:	Notification to an Adaptive Application or functional cluster in case of corrupted data in persistent memory, which is essential for safety use cases. The detection of data corruption is also necessary to support data recovery mechanisms.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00011, RS_SAF_10039)*

[RS_PER_00009]{DRAFT} **Persistency shall support data recovery mechanisms if persistent data was corrupted** ⌈

Description:	Persistency shall support a recovery mechanism if corruption of persistently stored data was detected. To be able to recover corrupted data, a redundant copy of the data is needed. The actual mechanisms and the granularity of redundancy are subject to configuration. Persistency shall also support a notification of the application in case recovery took place.
Rationale:	Applications want to recover corrupted data.
Use Case:	If corruption of persistent data was detected it shall be possible to recover corrupted data.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00011, RS_SAF_10040)*

4.3.2.6 Functional Cluster: Communication Management (CM)

[RS_CM_00223]{DRAFT} **The Communication Management shall protect the transmission of events using E2E protocol. The E2E Protection has to be executed behind the event API.** ⌈

Description:	Application developers shall be able to have an E2E-protected event-based communication, regardless of the bus used.
Rationale:	It shall be ensured that communication failure modes introduced by the communication bus (on the E2E-protected serialized data) which are detectable by the E2E protocol are detected by Communication Management.Note: It depends on the used communication type (periodic/ non-periodic) and the application which failure modes are to be detected.
Use Case:	Application “A” receives an E2E-protected speed (as a part of an event). In case of a corruption or a loss, this is detected by a periodic polling by application and by E2E checks (CRC and a stuck-at counter), reported by Communication Management by E2E result. As a result, the application could enforce the safe state of its function, e.g. refusing to open tail gate.
Dependencies:	–
Supporting Material:	[5]

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00010, RS_SAF_10039)*

[RS_CM_00224]{DRAFT} **The communication management shall provide the E2E information of the received event to the application.** ⌈

Description:	The communication management shall provide the E2E information of the received event to the application.
Rationale:	In case of reception of invalid E2E check result, the application shall be able to perform an appropriate error handling. The access to the event data is identical for safety-related and non-safety-related data.
Use Case:	Application “A” polls gets invalid E2E check result and as a result it switches to a safe state.
Dependencies:	–
Supporting Material:	The provided E2E information shall be, for each event in the queue: E2E status, E2E state, and the sample. Note that in case applications are triggered, there may be a need of an application-level detection of timeouts. This is because in case of delay or loss, the event will not arrive and E2E check will not be performed.

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00010, RS_SAF_10039, RS_SAF_-10014)*

[RS_CM_00400]{DRAFT} **Communication Management shall protect the transmission of methods using E2E protocol.** ⌈

Description:	Communication Management shall, transparent to the application, protect the transmission of methods using E2E protocol.
Rationale:	It shall be ensured that communication failure modes introduced by the communication bus (on the E2E-protected serialized request or response data) which are detectable by the E2E protocol are detected at the client side by Communication Management. Note: It depends on the used communication type (periodic/ non-periodic) and the application which failure modes are to be detected.
Use Case:	E2E protected method calls in client-server based communication
Dependencies:	–
Supporting Material:	[5]

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00010, RS_SAF_10039)*

[RS_CM_00401]{DRAFT} **The communication management shall provide the E2E information of the received method call to the application.** ⌈

Description:	The communication management shall provide the E2E information of the received method call to the application.
Rationale:	In case of reception of invalid E2E check result, the application shall be able to propagate detected E2E failure modes to the response data provided to the client. The access to the request data is identical for safety-related and non-safety-related data.
Use Case:	Application “B” provides a method and this method is called by application “A” and receives with the request invalid E2E check result and as a result the same invalid E2E data are added to the response data
Dependencies:	–
Supporting Material:	The provided E2E information shall be E2E status, E2E state and object data.

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00010, RS_SAF_10039, RS_SAF_-10014)*

[RS_CM_00403]{DRAFT} **Communication management shall provide an interface to detect delay of E2E protected service responses at the client side by supervision of a predefined response deadline.** ⌈

Description:	Communication management shall provide an interface to detect delayed service responses at the client side by supervision of a predefined response deadline.
Rationale:	A delayed response shall be detected and the application can apply a safety related error reaction.
Use Case:	Client is sending a method call. Client is awaiting the response within 300ms. After reaching the deadline the fault is detected at client side.

▽

△

Dependencies:	–
Supporting Material:	–

⌋(*RS_Main_01002*, *RS_Main_00060*, *RS_Main_00010*, *RS_SAF_10039*, *RS_SAF_10014*)

[RS_CM_00404]{DRAFT} **The communication management shall provide the E2E information of the method response to the application.** ⌈

Description:	The communication management shall provide the E2E information of the method response to the application.
Rationale:	In case of reception of invalid E2E check result, the application shall be able to perform an appropriate error handling. The access to the response data is identical for safety-related and non-safety-related data.
Use Case:	Application “A” requests a method call and receives with the response an invalid E2E check result and as a result it switches to a safe state.
Dependencies:	–
Supporting Material:	Note, there may be a need of an application-level monitoring of a deadline to stop waiting for a response.

⌋(*RS_Main_01002*, *RS_Main_00060*, *RS_Main_00010*, *RS_SAF_10039*, *RS_SAF_10014*)

4.3.2.7 Functional Cluster: Update and Configuration Management (UCM)

[RS_UCM_00008]{DRAFT} **UCM shall support a recovery mechanism in case of failed activation** ⌈

Description:	UCM shall assure that, in case of failed update process, the system will recover to the state it was before the update process started.
Rationale:	A failed update shall not result in a loss of desired functionality of the AUTOSAR Adaptive Platform.
Use Case:	After a failed remote update the AUTOSAR Adaptive Platform recovers to the previous system state.
Dependencies:	[RS_UCM_00021]
Supporting Material:	–

⌋(*RS_Main_00150*, *RS_Main_00503*, *RS_Main_00011*, *RS_SAF_10027*)

[RS_UCM_00012]{DRAFT} **UCM shall check the consistency of transferred Software Package** ⌈

Description:	UCM shall check the consistency of the received Software Package.
Rationale:	AUTOSAR Adaptive Platform shall make sure that the Software Package can be installed safely.
Use Case:	To detect possible errors which might have occurred during creation of the Software Package, UCM shall check that provided Software Package meta-data and content match.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503, RS_SAF_10039)*

[RS_UCM_00027]{DRAFT} **UCM shall be able to safely recover from unexpected interruption.** ⌈

Description:	At startup, UCM shall be able to identify if some action was interrupted and exited in an uncontrolled way and needs to be reverted or finished to return the software into the previous state
Rationale:	UCM shall make sure that software should not be started up into inconsistent and not updatable state
Use Case:	After unexpected reset or crash UCM shall identify that there was an interruption while an action was on going and UCM shall handle this by reverting or by finishing the unfinished action.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503, RS_SAF_10027)*

[RS_UCM_00030]{DRAFT} **UCM shall be able to verify the updated software during activation** ⌈

Description:	UCM shall require the updated software to be executed and verified before declaring that SW was successfully activated.
Rationale:	UCM shall declare activation to be successful only after it detects that Execution Manager can execute the software successfully.
Use Case:	Ensuring that safety-critical application can be executed and thus monitored by the Platform Health Manager.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00503, RS_SAF_10002)*

[RS_UCM_00035]{DRAFT} **`UCM Master` shall coordinate software update in a vehicle across multiple `Electronic Control Units`** ⌈

Description:	`UCM Master` is responsible of coordinating the distribution and processing of `Software Packages`s in several `ECU`s
Rationale:	There could be dependencies between `Machine`s or `ECU`s that shall be resolved by a central entity in the vehicle and that require specific processing or activation ordering
Use Case:	Complete vehicle `Electronic Control Unit`s update
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00011, RS_Main_00503, RS_SAF_10027)*

[RS_UCM_00037]{DRAFT} **`UCM Master` shall ensure it is safe to perform any modification to the vehicle** ⌈

Description:	`UCM Master` shall start any update, removal or install of `Software Packages` depending of safety requirements and kind of `Software Package`
Rationale:	`Software Package` can have big or no impact on vehicle safety
Use Case:	Performing an update of autonomous driving features will require for instance to have vehicle standing still, closed doors, etc. but a simple service might not have to consider vehicle safety.
Dependencies:	–
Supporting Material:	–

⌋*(RS_Main_00011, RS_SAF_10038)*

4.3.3 AUTOSAR ClassicPlatform

4.3.3.1 Basic Software: Watchdog Manager (WDGM)

[RS_SAF_31101]{DRAFT} **Watchdog Manager inherits highest safety integrity level from Software Component.** ⌈

Description:	Watchdog Manager shall inherit at least the highest safety integrity level from any Software Component that is running on the platform.
Rationale:	Watchdog Manager is responsible for ensuring part of the safe execution of safety relevant software components/applications, it should at least be developed with the highest ASIL as the software component/application that is being supervised.

▽

△

Use Case:	An ASIL C, B and QM Application is running on the Classic Platform. WdgM shall supervise the ASIL C and B application, therefore WdgM shall be implemented with an ASIL C.
AppliesTo:	CP
Dependencies:	WdgIf, Wdg Drv
Supporting Material:	–

⌋(*RS_SAF_10001*)

[RS_SAF_31102]{DRAFT} **Watchdog Manager monitors aliveness.** ⌈

Description:	Watchdog Manager shall monitor the aliveness of safety relevant software components/applications and modules.
Rationale:	Alive Supervision is one of the mechanisms of Watchdog Manager by which it monitors safety relevant software components/applications and modules.
Use Case:	A safety critical functionality with alive supervision gets stuck at some point in time during execution. WdgM detects that the supervised application is not alive.
AppliesTo:	CP
Dependencies:	WdgIf, Wdg Drv
Supporting Material:	–

⌋(*RS_SAF_10031*)

[RS_SAF_31103]{DRAFT} **Watchdog Manager monitors control flow.** ⌈

Description:	Watchdog Manager shall monitor the control flow of safety relevant software components/applications and modules.
Rationale:	Logical Supervision is one of the mechanisms of Watchdog Manager by which it monitors safety relevant software components/applications and modules.
Use Case:	A safety relevant functionality is developed to follow a specific control flow and is suddenly not following the intended sequence. Watchdog Manager detects the control flow violation.
AppliesTo:	CP
Dependencies:	WdgIf, Wdg Drv
Supporting Material:	–

⌋(*RS_SAF_10005*, *RS_SAF_10006*, *RS_SAF_10030*)

[RS_SAF_31104]{DRAFT} **Watchdog Manager monitors deadline.** ⌈

Description:	Watchdog Manager shall monitor that the duration between the checkpoints of safety relevant software components/applications and modules are within the minimum and maximum configured time limits.

▽

△

Rationale:	Deadline Supervision is one of the mechanisms of Watchdog Manager by which it monitors safety relevant functionalities.
Use Case:	A safety critical application is developed to reach specific checkpoints in a defined time window and is suddenly not behaving as intended. WdgM detects the violation.
AppliesTo:	CP
Dependencies:	WdgIf, Wdg Drv
Supporting Material:	–

⌋(*RS_SAF_10031*)

4.3.3.2 Basic Software: Operating System (OS)

[RS_SAF_31201]{DRAFT} **Memory Protection of Applications** ⌈

Description:	The Operating System shall prevent applications from performing write accesses outside their assigned memory regions
Rationale:	To achieve freedom from interference it is necessary to prevent applications from adversely affecting other applications. Access to private memory which is reserved for applications shall be protected against un-allowed write accesses from other applications.
Use Case:	A QM application and a ASIL application are executed on the same machine. OS prevents the QM application from changing the memory assigned to the safety critical application.
AppliesTo:	CP
Dependencies:	OSEK OS
Supporting Material:	–

⌋(*RS_SAF_10037*)

[RS_SAF_31202]{DRAFT} **Timing Protection of Applications** ⌈

Description:	The Operating System shall not allow a timing fault in any application to propagate. A timing fault may be caused by • exceeding a statically/pre-runtime specified execution time budget • exceeding a statically/pre-runtime specified blocking time budget • exceeding a statically/pre-runtime specified arrival rate
Rationale:	A timing fault in one application might trigger a chain of timing faults and this shall be prevented.
Use Case:	A QM application and a ASIL application are executed on the same machine. OS prevents the QM application from propagating its delay to the safety critical application.

▽

△

AppliesTo:	CP
Dependencies:	OSEK OS
Supporting Material:	–

⌋(*RS_SAF_10031*)

4.3.3.3 E2E Protection

[RS_SAF_31301]{DRAFT} **E2E Protection with E2E Transformer and E2E Library** ⌈

Description:	Communication Service, E2E Transformer and E2E Library shall provide mechanisms for detection of errors during the exchange of information among software components, by considering all faults listed in the ISO standard (ISO 26262:6-2018 D.2.4). **The Result of the E2E check needs to be published to the application.** If E2E Transformer is used RTE Interfaces need to be developed according to the same ASIL Level as the Application and data being transformed.
Rationale:	This requirement is created initially to fulfill the goal of AUTOSAR in supporting the development of safety-related systems by offering safety measures and mechanisms. As users may build project-specific applications, it is only possible for AUTOSAR to provide the safe exchange of information. ISO 26262 is mentioned and to be followed, as it is the international standard for functional safety of E/E systems for automotive.
Use Case:	Two ASIL rated applications on different control devices shall exchange information through a component (HW or SW) with a lower rated ASIL. E2E Transformer and E2E Library shall support safety mechanisms like a counter, a checksum and a timestamp to allow the ASIL applications or the E2E Transformer and E2E Library implementations to detect and ensure that the information has been transmitted correctly, in time and in-order.
AppliesTo:	CP
Dependencies:	E2E Library, E2E Transformer, RTE, SWC
Supporting Material:	–

⌋(*RS_SAF_10014*, *RS_SAF_10037*)

[RS_SAF_31302]{DRAFT} **Allow integrators to configure safety mechanisms to detect communication faults** ⌈

Description:	Communication Service, E2E Transformer and E2E Library shall, based on individual safety concepts, allow integrators to select and configure the set of safety mechanisms to detect communication faults.

▽

△

Rationale:	Different communication buses and data information may have different needs to be protected by the E2E. Therefore, it is reasonable to have the configurability (pre-deployment) such that integrators may freely select the set of mechanisms to be deployed.
Use Case:	A hi-level design change or new information requires a different communication protection mechanism. An integrator can select the proper protection by changing the Manifest.
AppliesTo:	CP
Dependencies:	E2E Library, E2E Transformer, RTE, SWC
Supporting Material:	–

⌋(*RS_SAF_10001*)

5 Requirements Tracing

The following table references the requirements specified in [6] and links to the fulfillment of these.

Feature	Description	Satisfied by
[RS_Main_00010]	Safety Mechanisms	[RS_SAF_00001] [RS_SAF_00002] [RS_SAF_00003] [RS_SAF_00004] [RS_SAF_00005] [RS_SAF_00006] [RS_SAF_00007]
[RS_Main_00011]	Mechanisms for Reliable Systems	[RS_SAF_00001] [RS_SAF_00002] [RS_SAF_00003] [RS_SAF_00004]
[RS_Main_00012]	Highly Available Systems Support	[RS_SAF_00001] [RS_SAF_00002] [RS_SAF_00003] [RS_SAF_00004]
[RS_Main_00030]	Safety Related Process Support	[RS_SAF_00001] [RS_SAF_00002] [RS_SAF_00003] [RS_SAF_00004]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_SAF_00003]
[RS_SAF_00001]	Safe Execution	[RS_SAF_10001] [RS_SAF_10005] [RS_SAF_10008] [RS_SAF_10028] [RS_SAF_10030] [RS_SAF_10031]
[RS_SAF_00002]	Safe Configuration	[RS_SAF_10001] [RS_SAF_10002] [RS_SAF_10008] [RS_SAF_10027] [RS_SAF_10028] [RS_SAF_10037] [RS_SAF_10039]
[RS_SAF_00003]	Safe Update or Safe Upgrade	[RS_SAF_10002] [RS_SAF_10005] [RS_SAF_10037] [RS_SAF_10038] [RS_SAF_10039]
[RS_SAF_00004]	Safe Exchange of Information	[RS_SAF_10008] [RS_SAF_10014] [RS_SAF_10037] [RS_SAF_10039] [RS_SAF_10041] [RS_SAF_10042]
[RS_SAF_00005]	Detection of Data Corruption	[RS_SAF_10041]
[RS_SAF_00006]	Safe Storage	[RS_SAF_10040]
[RS_SAF_00007]	Recovery upon failure	[RS_SAF_10027]

[RS_SAF_10001]	AUTOSAR shall provide mechanisms to support safe initialization of software components.	[RS_SAF_31101] [RS_SAF_31302]
[RS_SAF_10005]	AUTOSAR shall provide mechanisms to support safe shutdown and termination of applications, software components, basic software modules and services.	[RS_SAF_31103]
[RS_SAF_10006]	AUTOSAR shall provide mechanisms to support safe transition of states in a basic software module, software component, application or service life cycle.	[RS_SAF_31103]
[RS_SAF_10008]	AUTOSAR shall provide mechanisms to support safe resource management for the AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.	[RS_SAF_21401] [RS_SAF_21402] [RS_SAF_21403]
[RS_SAF_10014]	AUTOSAR shall provide an interface to support safe communication for basic software module, software component, application or services.	[RS_SAF_31301]
[RS_SAF_10028]	AUTOSAR shall provide mechanisms to support dependable scheduling of AUTOSAR Adaptive Platform functional-clusters, applications and services and AUTOSAR Classic Platform basic software modules and software components.	[RS_SAF_21401]
[RS_SAF_10030]	AUTOSAR shall provide mechanisms to support safe program execution.	[RS_SAF_31103]
[RS_SAF_10031]	AUTOSAR shall provide mechanisms to detect program execution time violation	[RS_SAF_21401] [RS_SAF_31102] [RS_SAF_31104] [RS_SAF_31202]
[RS_SAF_10037]	AUTOSAR shall provide mechanisms to prevent unintended alteration of data.	[RS_SAF_31201] [RS_SAF_31301]

6 References

[1] ISO 26262:2018 (all parts) – Road vehicles – Functional Safety
http://www.iso.org

[2] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[3] Glossary
AUTOSAR_TR_Glossary

[4] Explanation of Safety Overview
AUTOSAR_EXP_SafetyOverview

[5] Requirements on E2E
AUTOSAR_RS_E2E

[6] Main Requirements
AUTOSAR_RS_Main