Document Title	Explanation of Safety Overview
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	895

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	18-03

Document Change History			
Date	**Release**	**Changed by**	**Description**
2018-03-29	18-03	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

Bibliography

[1] Virtual Functional Bus
AUTOSAR_EXP_VFB

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture

[3] AUTOSAR Introduction
AUTOSAR_Introduction.pdf

[4] Explanation of Adaptive Platform Design
AUTOSAR_EXP_PlatformDesign

[5] ISO 26262 (Part 1-10) – Road vehicles – Functional Safety, First edition
http://www.iso.org

[6] Utilization of Crypto Services
AUTOSAR_EXP_UtilizationOfCryptoServices

[7] Specification of Operating System Interface
AUTOSAR_SWS_OperatingSystemInterface

[8] Design guidelines for using parallel processing technologies on Adaptive Platform
AUTOSAR_EXP_ParallelProcessingGuidelines

[9] Specification of Platform Types
AUTOSAR_SWS_PlatformTypes

[10] Specification of Execution Management
AUTOSAR_SWS_ExecutionManagement

[11] Explanation of ara::com API
AUTOSAR_EXP_ARAComAPI

[12] Specification of Persistency
AUTOSAR_SWS_Persistency

[13] Specification of Health Management for Adaptive Platform
AUTOSAR_SWS_HealthManagement

[14] Specification of Identity and Access Management
AUTOSAR_SWS_IdentityAndAccessManagement

[15] Specification of RESTful communication
AUTOSAR_SWS_REST

[16] Specification of Time Synchronization for Adaptive Platform
AUTOSAR_SWS_TimeSync

[17] Specification of Log and Trace for Adaptive Platform
AUTOSAR_SWS_AdaptiveLogAndTrace

[18] Specification of Crypto Interface
AUTOSAR_SWS_CryptoInterface

[19] Specification of ECU State Manager
AUTOSAR_SWS_ECUStateManager

[20] Specification of Diagnostic Communication Manager
AUTOSAR_SWS_DiagnosticCommunicationManager

[21] Specification of Network Management Interface
AUTOSAR_SWS_NetworkManagementInterface

[22] Specification of Update and Configuration Management
AUTOSAR_SWS_UpdateAndConfigManagement

[23] Methodology for Adaptive Platform
AUTOSAR_TR_AdaptiveMethodology

[24] Mapping mixed-criticality applications on multi-core architectures

[25] Specification of SW-C End-to-End Communication Protection Library
AUTOSAR_SWS_E2ELibrary

[26] Requirements on E2E
AUTOSAR_RS_E2E

[27] Specification of Communication Management
AUTOSAR_SWS_CommunicationManagement

Known Limitations

AUTOSAR specifications may contain exemplary items, like exemplary reference models, use-cases, scenarios, 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.

The chapter 5

- Functional Safety Concept and initial Functional Safety Requirements

is still in development and open discussion and should not be considered mature or final. The chapters

- Technical Safety Concept
- Safety Requirements
- Validation of Safety requirements

are scheduled for the later releases.

SEooC according to ISO26262 part 10

Whether the AUTOSAR Adaptive Platform architecture definition itself can be considered being a SEooC according to ISO26262 part 10 is still not verified yet. Either way, following the ISO26262 part 10 SEooC definition as a guideline for this document to create reusable content and similarities to a proper "Safety Manual" could be considered as an agreeable starting point. The safety goal of the AUTOSAR Adaptive Platform architecture is to enable and support systems up to ASIL-D.

No ASIL ratings

The AUTOSAR consortium, especially the Adaptive AUTOSAR workgroups are only providing an architecture definition, descriptions of the functional blocks and - in the best case - a *proof of concept* implementation, it is not possible to add concrete ASIL ratings to each architectural item in this scope. It is only possible to give the reader and final user some hints on how to combine the architectural items to achieve a safe architecture in his own very specific context: considering the underlying hardware, the products safety goals and metrics as well as the development processes.

Completeness

This document might not cover all possible scenarios in which the AUTOSAR Adaptive Platform could be used. The safety related requirements are derived from some specific use cases and to the best knowledge of all the members of the Adaptive AUTOSAR workgroups, contributers and reviewers.

1 Introduction

1.1 Motivation

Functional safety is a system characteristic which is taken into account from the beginning, as it may influence system and software architectural design decisions. Therefore, the AUTOSAR Adaptive Platform specifications include requirements related to functional safety. Aspects such as complexity of the system design can be relevant for the achievement of functional safety in the automotive field.

Software is one parameter that can influence complexity on system level. New techniques and concepts for software development can be used in order to minimize complexity and therefore can ease the achievement of functional safety. AUTOSAR Adaptive Platform supports the development of safety-related systems by offering safety measures and mechanisms.

However, AUTOSAR Adaptive Platform is not a complete safe solution. The objective of this safety overview is to derive safety requirements from the top level safety goals and assumed use-cases to allocate them to the architectural elements of the item, or to any external measures. The use of the AUTOSAR Adaptive Platform does not imply ISO26262 part 10 compliance. It is still possible to build unsafe systems using the AUTOSAR safety measures and mechanisms. The Architecture of the AUTOSAR Adaptive Platform can only be considered to be an Safety Element out of Context (SEooC).

Information about AUTOSAR Adaptive Platform functional safety mechanisms and measures is currently distributed throughout the referenced documentation. Unless one knows how functional safety mechanisms are supported and where the necessary information is specifically located, it is difficult to evaluate how a safety-relevant system can be implemented using AUTOSAR efficiently. This explanatory document summarizes the key points related to functional safety in AUTOSAR and explains how the functional safety mechanisms and measures can be used.

1.2 Scope

This document shall be explanatory and help the functional safety engineer to identify functional safety related topics within the AUTOSAR Adaptive Platform. The content of this document is structured into separate chapters as follows:

- AUTOSAR Adaptive Platform objectives, use-cases and scenarios
- System definition and context
- Top Level Safety Goals
- Functional Safety Concept and initial Functional Safety Requirements

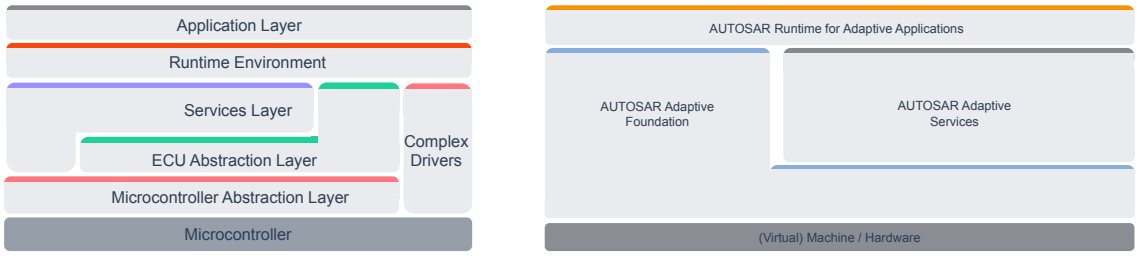
1.3 Intended Audience

This document gives an overview of the functional safety measures and mechanisms of the AUTOSAR Adaptive Platform and their implementation to those involved in the development of safety-relevant (ECU) systems. Therefore, this document is intended for the users of the AUTOSAR Adaptive Platform, including people involved in safety analysis.

2 Objectives of Adaptive AUTOSAR

2.1 Design Objectives

The overall design objectives of the AUTOSAR Adaptive Platform are similar to those of the well known and established AUTOSAR Classic Platform. The AUTOSAR Adaptive Platform is still providing an abstraction layer for the software developers AUTOSAR Runtime for Adaptive Applications (ARA) so that AUTOSAR Adaptive Platform applications could be exchanged between ECUs and developed, from a systematic viewpoint similar to the AUTOSAR Classic Platform BSW and VFB Layer - as it is described in AUTOSAR Classic Platform architecture in [1][2].



(a) AUTOSAR Classic Platform **(b) AUTOSAR Adaptive Platform**

Figure 2.1: AUTOSAR layered architectures [3]

2.2 Scenarios

Scenarios for the AUTOSAR Adaptive Platform are in particular - but not only - automotive grade electronic control units from the following domains:

- Autonomous Driving (from driver assistance to fully autonomous driving)
- Gateways
- Body-Domain Controller
- Infotainment-systems
- etc.

To resolve the requirements for more processing power for image and sensor processing, multi-sensor data-fusion or machine-learning as well as enhanced multimedia capabilities like 2D/3D graphics acceleration, video and audio processing the AUTOSAR Adaptive Platform shall support high performance computation units and accelerators.

The second major objective is to allow dynamic software upgrades and more flexible development and deployment of applications and services within the vehicle.

The third - and for the functional safety engineer most important - objective is the capability to execute applications with mixed criticality, from QM to ASIL-D within one partition while maintaining freedom from interference. If the System contains several

partitions, which may not even be ISO26262 compliant at all, like infotainment-systems, freedom from interference is still required but **not** within the scope of the AUTOSAR Adaptive Platform architecture and standards.

For more details regarding the objectives of AUTOSAR especially the AUTOSAR Adaptive Platform please have a look into the AUTOSAR Introduction presentation [3] and the explanatory AUTOSAR Adaptive Platform Design document [4].

2.3 Example Scenario: HAD

The Highly Autonomous Driving (HAD) scenario has been chosen to investigate the safety capabilities of the AUTOSAR Adaptive Platform. This scenario does not only cover the requirement for high performance computing and dynamic software updates but also the corresponding highest safety case: ASIL-D according to ISO26262 [5]. The system design on vehicle level is assumed to contain several sensors (e.g. odometry, GPS) or Sensor-ECUs (e.g. radar, lidar, vision). The vehicle is expected to have at least one ADAS-ECU for the AD functionality where adaptive AUTOSAR could be integrated, not only on that ADAS-ECU, but also on the Sensor-ECUs or any other before mentioned domain controller.

2.3.1 AUTOSAR Adaptive Platform features

The HAD scenario and the resulting HAD-applications require the following capabilities from the underlining AUTOSAR Adaptive Platform Foundation Libraries and Services as shown in figure 2.1b - besides the specialized HAD applications of course:

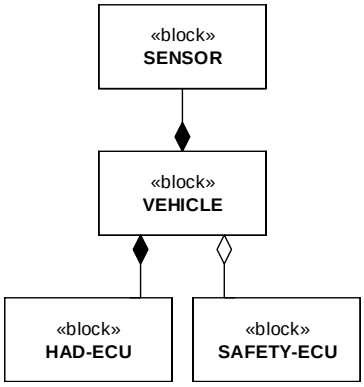
- safe and secure boot
- execution of applications
- scheduling of applications
- application state management: start, stop, halt, etc.
- runtime behavior monitoring: processing time, bus load, memory consumption, etc.
- access to application data
- persistent data storage
- configuration of ECU and application data
- update of deployed applications
- deployment of new applications
- system monitoring

- send and receive messages through vehicle networks: e.g CAN, CAN-FD, FlexRay, ETH

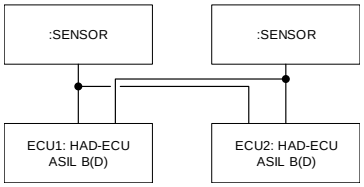
This feature list is not only related to the mentioned HAD scenario and could be applied to other domain specific ECUs too and comes so far without any further deep application and safety analysis on these topics.

2.3.2 Target HW

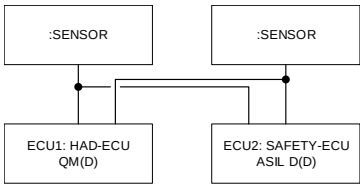
At the time of the initial definition of the AUTOSAR Adaptive Platform typical high performance processing units are not always reaching the safety rating of ASIL-D by itself, therefore several simple systematic designs have been considered to be able to reach ASIL-B or ASIL-D by proper decomposition. The AUTOSAR Adaptive Platform architecture can only support the actual System or Hardware developer to achieve the specific safety targets.



(a) Exemplary simplified vehicle system



(b) Systematic redundancy



(c) Decomposition with safety checker

Figure 2.2: Systematic approaches on vehicle level

The system design is **not** part of the AUTOSAR Adaptive Platform specification but both options are valid use-cases. It is up to the product developer to choose an proper system design to achieve the safety goals.

2.3.3 HW capabilities

Memory Protection mechanisms, Error-Correction and Built-In Self-Tests are required HW capabilities from the known AUTOSAR Classic Platform world to achieve a safety rating.

2.3.3.1 Hypervisor

To support proper partitioning for mixed criticality or to be able to execute non ISO26262 compliant software on the same system a well established feature like virtual machine monitor (VMM) or hypervisor could be used, see figure 2.3. The hypervisor should utilize special hardware capabilities to achieve the optimal performance ratings for e.g. safety, security and domain specific guests, like HAD. The safety classification of the hypervisor must be equal to the highest ASIL rating on the layers above.

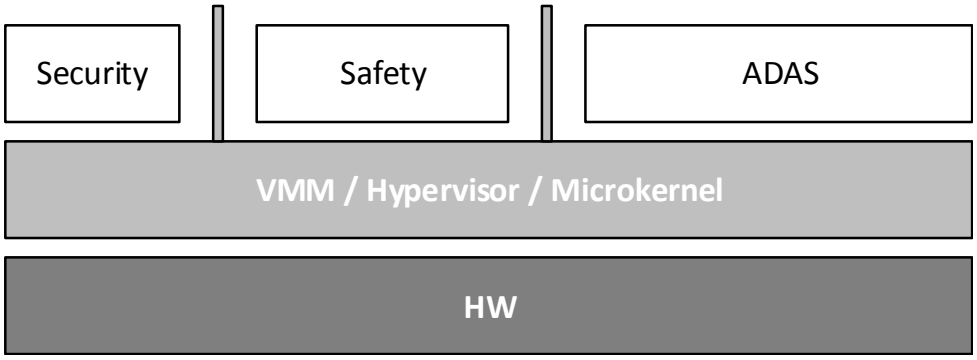


Figure 2.3: Virtual Machine Monitor / Hypervisor / Microkernel

This usage scenario is within the objectives of the AUTOSAR Adaptive Platform but not part of its architectural specification as seen in 2.1 and further in 3.3.

2.3.4 Safety Decomposition Strategies

To achieve the targeted ASIL rating an optimal software decomposition could be considered. Therefore a software component could be split into safety related and non-safety related parts or a mechanism known as *Software Lockstep* as described in section 2.3.4.3 could be used on.

2.3.4.1 Safety Checker

A safety decomposition strategy could be to integrate the control flow or an value checker on a dedicated safety island. This could be a special safety ECU, a safety processor or safety partition on the same IC (SoC). Properly authenticated and with end-to-end (E2E) protection a status flag or values could be sent to this safety checker, which decides if the result matches the expectations 2.2c.

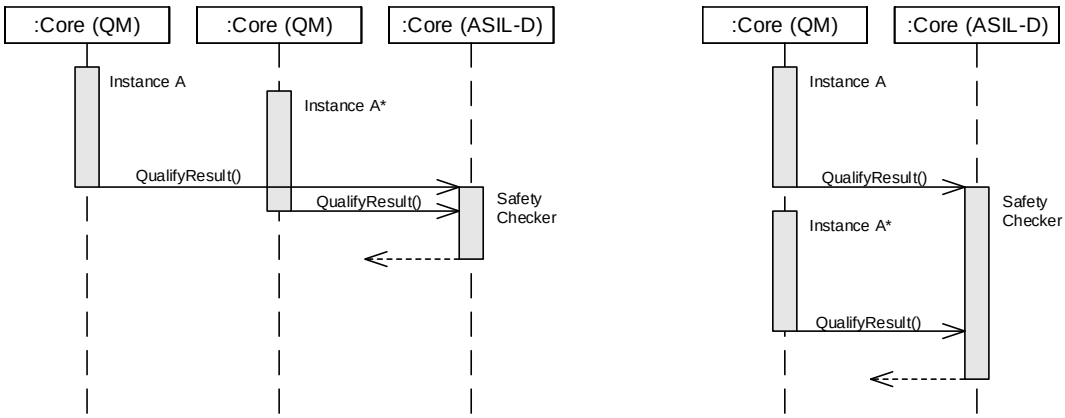
2.3.4.2 Self-Test-Library

In some cases a self-test library (STL) is required to be scheduled properly to test if the hardware is still working within the right boundaries and if the internal hardware or

software mechanisms are still active and working. Typically such an STL can be implemented as a normal AUTOSAR Adaptive Platform application and scheduled within the Fault Tolerant Time Interval (FTTI).

2.3.4.3 SW Lockstep

SW Lockstep could be required if an application needs to be executed on a HW which does not support the same classification and proper a ASIL decomposition is not possible (e.g. to achieve HW independent deployment). The easiest solution is just to run the application delayed on different cores, and therefore it is a scheduling strategy and a small extension to the applications might be required: the voting system. Such a *voter* or safety-checker shall control if both applications came to the same conclusion within the expected time.



(a) simultaneous execution **(b) repeated execution**

Figure 2.4: Safe scheduling

2.3.5 Security Capabilities

For autonomous driving security is expected to have a greater impact than in the past. Not only that communication channels and communication partners needs to be authenticated and verified, they also need to be safe. The security related topics of the AUTOSAR Adaptive Platform can be found in [6]. Some security related features are:

- secure boot
- authentication of communication partners within the vehicle network as well as with the off-board world
- secure key exchange
- secure key storage

3 System Description

3.1 Item Under Investigation

The Item under investigation in this explanatory document is the AUTOSAR Adaptive Platform Architecture running in a context roughly described in the previous chapter. Since the AUTOSAR Adaptive Platform Architecture will eventually be a piece of software declared as a SEooC, the platform it will be executed on needs to be investigated too, in order to derive some safety requirements which will finally be satisfied by software features as described and defined in the AUTOSAR Adaptive Platform architecture.

Modern ECUs contain highly modular embedded software, which can consist of both non-safety-related and safety-related software components, which perform functions with different ASIL ratings. According to ISO 26262, if the embedded software consists of software components with different ASIL ratings, then either the entire software must be developed according to the highest ASIL, or freedom from interference shall be ensured for software components with a higher ASIL rating from elements with a lower ASIL rating.

3.1.1 ECU

In a typical safety compliant ECU it can be assumed that, besides an microprocessor (uP or SoC) dynamic and persistent memory, it will be equipped with a Power Management Integrated Circuit (PMIC), Watchdog and some on-board-sensors or drivers as well as several input output channels, e.g. digital, analog or for communication via a vehicle bus like Ethernet, CAN or FlexRay.

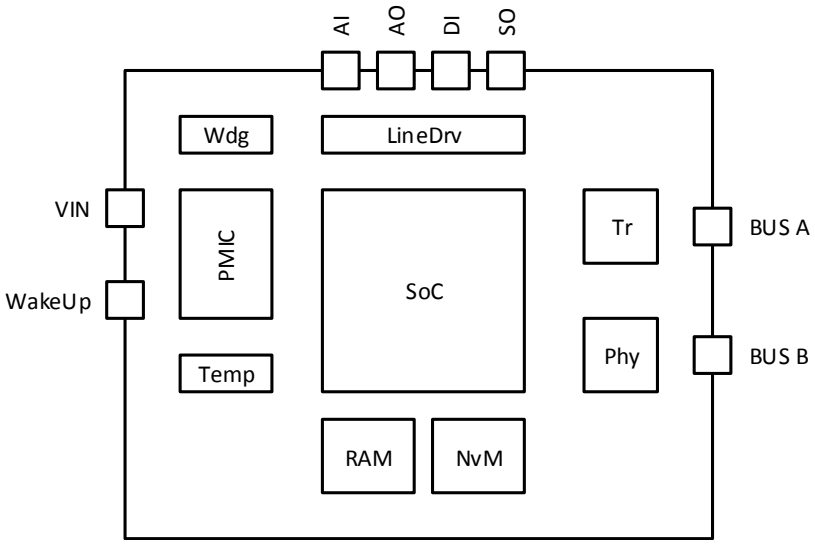


Figure 3.1: Exemplary draft of a common simple ECU design

Some simple on-board safety measures are:

- regulated and controlled power management
- power monitoring (voltage and current)
- temperature monitoring
- alive monitoring (Watchdog)
- input/output control

If the controller or the running software is not trustworthy anymore, e.g. if voltage levels are not stable or the watchdog has triggered, the line driver and the Transceivers might be disabled, to achieve the `Fail-Silent` behavior without software interaction.

3.1.2 Microprocessor

A Microprocessor design could look like shown in figure 3.2

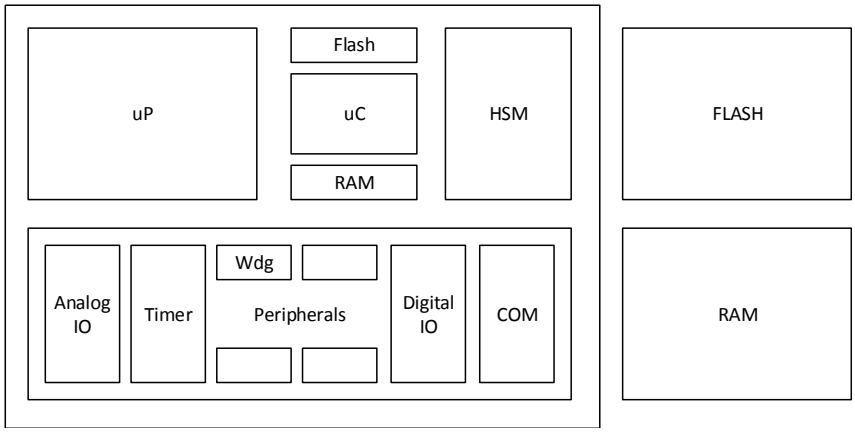


Figure 3.2: Exemplary draft of a common simple MCU design

A typical microprocessor suited for the AUTOSAR Adaptive Platform might contain several performance processing cores (uP) a Hardware Security Module (HSM) and in some cases also a peripheral micro-controller core (uC). The HSM and uC could be typical general purpose controller and be user-programmable or equipped with a firmware from the vendor. The main target for the AUTOSAR Adaptive Platform is the performance processor. The peripherals may or may not be accessible through the uP, peripheral access is not standardized in the AUTOSAR Adaptive Platform. The only HW requirements from the AUTOSAR Adaptive Platform are indirectly defined trough the OS, which shall provide multi-process support for isolation of applications and therefore requires a Memory Management Unit (MMU) according to [7]. If the ECU shall communicate with other ECUs support for Ethernet is intended with the SOME/IP protocol. External Flash and RAM is not directly required, but common practice in actual HW designs (as of 2018).

3.1.3 HW Accelerator

Hardware accelerators and parallel processing is respected within the of the AUTOSAR Adaptive Platform architecture. For more Information regarding this topic please read the "Design guidelines for using parallel processing technologies on Adaptive Platform [8]". The software development process and the required SW mechanisms for a hardware accelerator are basically the same as for the typical Microprocessor. There shall be mechanisms to check if SW routines are scheduled correctly, the computations are correct and the control flow shall be monitorable.

3.2 AUTOSAR Adaptive Platform Architecture Overview

3.2.1 AUTOSAR Adaptive Platform Architecture

The layered architecture of the AUTOSAR Adaptive Platform is shown in 3.3 and can be divided into three main parts as described in figure 2.1b

1. AUTOSAR Adaptive Platform Foundation Libraries
2. AUTOSAR Adaptive Platform Services
3. User Applications (Adaptive Applications and Non-Platform Services)

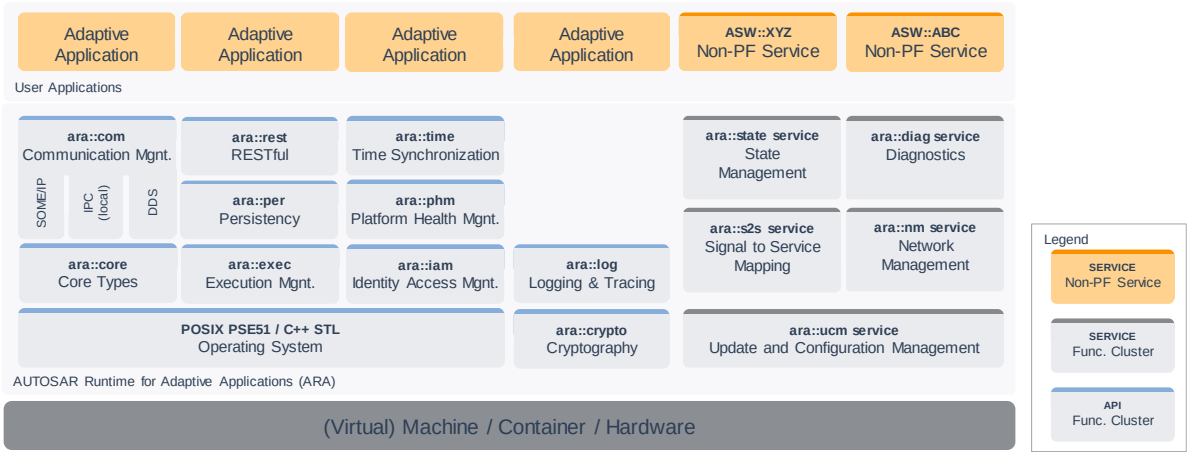


Figure 3.3: AUTOSAR Adaptive Platform functional blocks

The operating system (OS) itself is not directly part of the architecture, but the AUTOSAR Adaptive Platform has several requirements regarding the OS.

3.2.2 AUTOSAR Adaptive Platform Functional Cluster

The AUTOSAR Adaptive Platform functional cluster of the **Foundation Library** are

`ara::core`	Core Types [9]
`ara::exec`	Execution Management [10]
`ara::com`	Communication Management [11]
`ara::per`	Persistency [12]
`ara::phm`	Platform Health Management [13]
`ara::iam`	Identity Access Management [14]
`ara::rest`	RESTful communication [15]
`ara::time`	Time Synchronization [16]
`ara::log`	Logging & Tracing [17]
`ara::crypto`	Cryptography [18]

The functional cluster of the **Foundation Services** are

`ara::state`	State Management [19]
`ara::diag`	Diagnostics [20]
`ara::s2s`	Signal to Service mapping
`ara::nm`	Network Management [21]
`ara::ucm`	Update and Configuration Management [22]

The detailed description for the AUTOSAR Adaptive Platform modules can be found in the respective specialized documents. A summary is also part of the "Explanation of Adaptive Platform Design [4]"

3.3 Functional Safety Mechanisms

Furthermore, the ISO26262 standard provides examples for faults, which cause interference between software components. The faults are grouped as follows:

- Memory
- Timing
- Execution
- Exchange of Information
- Authentication of applications and services
- Rights Management

3.3.1 Memory Partitioning

A modular implementation of embedded systems that consists of both safety-related software components of different ASIL or of safety-related and non-safety-related software components is facilitated by AUTOSAR features that support freedom from interference between such software components. AUTOSAR Adaptive Platform applications or services, which are developed according to a low ASIL rating may interfere by wrongfully accessing memory regions of software components with a higher ASIL

rating. An execution of applications in separate memory regions or memory partitions supports the prevention of such memory access violations. The features described in this chapter are part of the OS and the `ExecutionManager` functionality, which are required to enable groups of applications or services to run in separate memory partitions, in order to provide freedom from interference to and from other applications or services.

3.3.1.1 Fault Models

According to ISO26262, the following memory-related effects of faults can be considered as a cause for interference between software components:

- Corruption of content.
- Read or write access to memory allocated to another software element.

The functional safety mechanism memory partitioning provides protection by means of restricting access to memory and memory-mapped hardware. Memory partitioning means that Applications reside in different memory areas (partitions) that are protected from each other. In particular, code executing in one partition cannot modify memory of a different partition. Moreover, memory partitioning enables to protect read-only memory segments (e.g. code execution), as well as to protect memory-mapped hardware. The memory partitioning and user/supervisor-modes related features address the following goal: Supporting freedom from interference between software components by means of memory partitioning (e.g. memory-related faults in applications do not propagate to other software modules and applications executed in user-mode have restricted access to CPU instructions like e.g. reconfiguration).

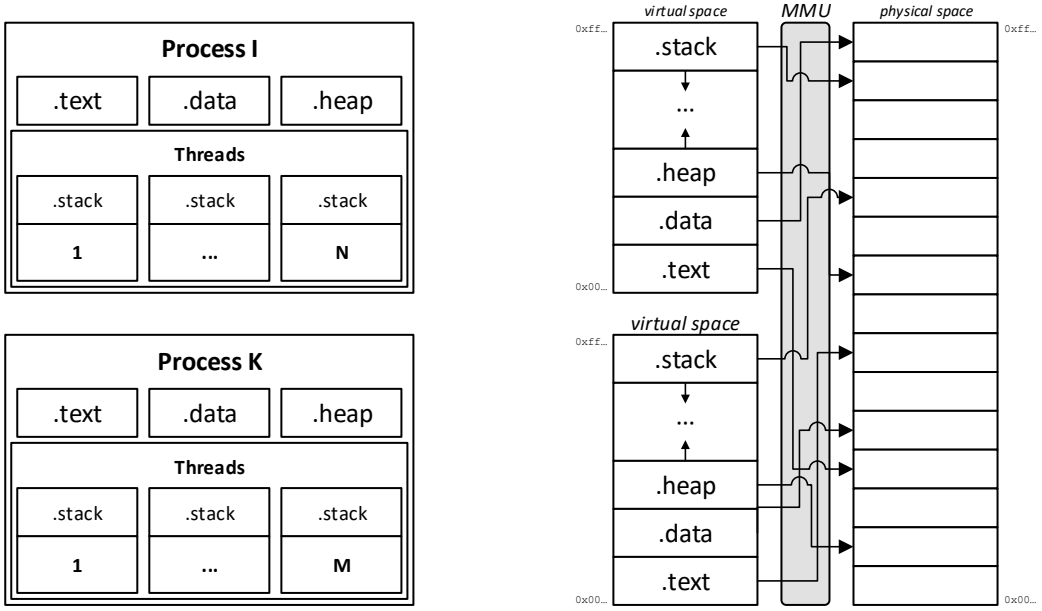
3.3.1.2 Description

Memory partitioning is provided by the VMM, the OS and HW through the MMU and VM-tags. The POSIX process and thread model will use the MMU to create an virtual address space.

During the course of this chapter, this extension will be described as the relationship of Processes, Threads and Applications in the context of the AUTOSAR Methodology [23].

3.3.1.3 Applications and Services

In the AUTOSAR Adaptive Platform architecture, Adaptive Applications make use of the AUTOSAR Adaptive Platform Foundation Libraries and Services Layer. By including the interfaces to the AUTOSAR Adaptive Platform Foundation Libraries and Ser-



(a) Process and Thread Model **(b) Virtual/Physical Address Space**

vices Adaptive Applications can access and use the functional clusters provided by the AUTOSAR Adaptive Platform Foundation Libraries and Services.

Adaptive Applications are intended to be hardware-independent, so that they can be integrated or deployed on any available ECU. To facilitate the inter- and intra-ECU information exchange, AUTOSAR Adaptive Platform application can communicate exclusively over the provided interfaces of `ara::com`

AUTOSAR Adaptive Platform application contain a number of functions and variables, which provide the internal functionality. The internal structure of an AUTOSAR Adaptive Platform application, its variables and function calls, is hidden from the public view. Inter process communication (IPC) can be realized through the `ara::com`, services or via RESTful communication.

AUTOSAR Adaptive Platform applications are executed cyclically or event-driven. Various scheduling mechanisms like Round Robin (RR), First-In-First-Out (FIFO) or Deadline could be used to achieve the intended runtime behavior.

3.3.1.4 Memory Partitioning within application software

Application Software in an AUTOSAR ECU can consist of safety-related and non-safety-related applications or services. Freedom from interference between applications, processes and threads with different ASIL ratings shall be ensured according to the requirements of ISO26262.

AUTOSAR Adaptive Platform applications can consist of multiple processes with different ASIL levels. However, threads with different ASIL ratings should not be assigned to the same process. Memory partitioning does not provide freedom from interference between threads which are assigned to the same virtual address space. The operating system only prevents other processes from performing improper accesses. A faulty software thread in the same process would not be prevented from modifying memory areas of other threads within the same virtual address space.

3.4 Hardware and Software Fault Considerations

Even if the Hardware is not part of the AUTOSAR Adaptive Platform architecture, it is necessary to respect the HW to define the source of a higher safety requirement eventually. This section is meant to collect and describe typical hardware and software faults along with the safety measures which directly affects the adaptive platform. Most likely, not all hardware and software faults will be described here and not all effects will be analyzed sufficiently enough. Therefore, it is mandatory to perform a full safety evaluation for each safety-critical application built on top of the AUTOSAR Adaptive Platform according to the relevant industry standards.

3.4.1 Hardware Faults and Safety Measures

Incorrect execution of multiple applications with mixed criticality may be due to systematic faults (e.g. bugs in processor design) or random hardware faults. Natural phenomena, such as ionized radiation (e.g. high energy particle impacts), electromagnetic compliance, vibrations, aging effects or external environmental conditions, can lead to such malfunctions. Integrating applications with different criticalities on a single platform can be very tricky. Partitioning mechanisms on hardware level can be applied in order to isolate these applications [24]. Hardware partitioning based on safety criticality of AUTOSAR Adaptive Platform applications, ensures a lesser impact of single points of failure compared to software or logical partitioning as errors in one hardware partition do not have effect on other partitions. However, hardware partitioning techniques may compromise performance when two applications on different hardware partition need to communicate.

We may categorize hardware faults into three different classes; transient, intermittent and permanent. Transient fault may occur once and is not reproducible (e.g. Single Event Upset). An intermittent fault on the other hand occurs every now and then, but usually at irregular intervals (e.g. A fault occurring due to environmental conditions such as temperature or humidity). As the name suggests, a permanent fault is reproducible every time and will persist unless the faulty component is not replaced (e.g. Single Event Latch-up).

Following is a list of typical measures that can be taken in order to detect/avoid the above mentioned hardware faults:

- Cyclic Configuration Test
- Cyclic hardware part test (using known test vectors)
- Shutdown Path Test ("Can the safe state be reached?")
- Memory Walk-through Tests (e.g. test for writability)
- Clock Monitoring, Power Monitoring, Timing Monitoring (timing predictions may be very inaccurate in high-performance microprocessors due to the inherent complexity of such systems)
- Plausibility Checks (but only applicable if checks are significantly easier to calculate than the functions to be monitored)
- External watchdog
- End-to-End Protection
- Hardware Lockstep CPU Cores (although this may not always be present in high-performance microprocessors)
- ECC Memory, Error detection for data and address links
- Redundant Execution (2oo2, 2oo2D, 2oo3)
- Proper Hardware Design (the choices in high-performance microprocessors may be very limited due to the complexity of hardware architecture and may result in common cause failures)
- Proper communication BUS
- Proper shielding
- Proper Electromagnetic Compatibility (EMC)

3.4.2 Software Faults and Safety Measures

Hardware faults may impact software directly or indirectly. Examples of direct impact may include an arithmetic miscalculation (although the control flow of a program may be correct) or a wrong control flow may cause a jump in address which could result in undefined behavior, infinite loop or premature end of execution. Examples of indirect impact may include; affecting other CPU Cores (overload on OS, caches, memory, peripherals or cross-core interrupt flooding or an intense heating of one core may cause shutdown), memory corruption via software and misconfiguration of OS, platform services or peripherals (corruption of OS scheduling table or unintended execution of 'Disable Interrupts' instruction or misconfiguration of real-time clock).

Following is a list of typical measures that can be taken in order to detect/avoid the above mentioned software faults:

- Redundant Execution (2oo2, 2oo2D, 2oo3)

- Program Flow Control ("Does the software pass-by known points in the right order?")
- Checksums
- Arbitration
- Collision Detection
- Signatures
- Software Lockstep
- Parallel Execution
- Safety Checker

One of the robust safety measures would be to detect and prevent failure propagation via software in an Adaptive AUTOSAR system. Failure propagation can be detected by software monitors performing plausibility checks. With dual modular redundancy (DMR) a failure can be detected. Moreover, with a triple modular redundancy (TMR) in place and a voting mechanism, a failure can even be corrected. Thus, redundant execution is helpful in detecting if not correcting a failure propagation. Enforcement of security policies can help detect access violations e.g. a user process accesses a resource it has no access rights to.

In order to avoid failure propagation, access rights need to be restricted. The privileges should be reduced in user-mode. If a user process executes privileged operations, the OS should run plausibility checks before granting this. However, OS and drivers may be running in privileged mode and become a common cause of failure. Platform configurations (such as BIOS settings and special registers) should be read-only at runtime and read-write only before booting the OS. Only a reasonable bandwidth should be allocated for CPU computational power, memory and peripherals at runtime to avoid affecting the whole system due to a faulty module/component. Another measure to prevent failure propagation is to enforce mutual exclusion, through hardware or OS, for specific resources e.g flash, peripherals etc.

4 Hazard Analysis and Risk Assessment & Safety Goals

4.1 Top Level Safety Feature Requests

AP-FEAT-01	Provide flexible execution time and resources for multiple, mixed criticality applications.
AP-FEAT-02	Provide dyn. configurable, updatable and upgrade runtime for multiple, mixed criticality applications.
AP-FEAT-03	Provide information exchange between multiple, mixed criticality applications.
AP-FEAT-04	Provide information exchange between mixed criticality application and other external components as sensors, actors or ECUs inside the vehicle.
AP-FEAT-05	Provide information exchange between mixed criticality application and other external components outside the vehicle.
AP-FEAT-06	Maintain correct configuration during the driving cycle

4.2 Top Level Safety Goals

The AUTOSAR Adaptive Platform is only a part of "larger" item definition, as displayed above, there are no direct safety goals stemming from a hazard and risk analysis. Therefore the following Top Level Safety Goals are considered as they define the major failure modes of the AUTOSAR Adaptive Platform.

AP-SG-01	Ensure correct computation and execution of multiple applications with mixed criticality
AP-SG-02	Ensure correct configuration maintenance during the entire driving cycle
AP-SG-03	Ensure correct update and upgrade of multiple applications with mixed criticality (& platform)
AP-SG-04	Ensure correct exchange (transmission and reception) of information

All Top Level Safety Goals shall be achieved with ASIL-D. ASIL-D Fail-operational qualities shall be achieved, if one of the Top Level Safety Goals is violated 4.1.

4.3 Top Level Hazards and Malfunctions

AP-HA-01	unintended, untimely, incorrect execution of applications
AP-HA-02	unintended, untimely, incorrect configuration, update and upgrade of applications
AP-HA-03	unintended, untimely, incorrect exchange of information between applications
AP-HA-04	unintended, untimely, incorrect exchange of information between applications and external components inside the vehicle
AP-HA-05	unintended, untimely, incorrect exchange of information between applications and external components outside the vehicle
AP-HA-06	Corruption of configuration

4.4 AUTOSAR Adaptive Platform Supported Failure Modes Targets

Fail Silent	Fail-Silent is a property of a system in which no output is produced in the presence of the fault *Service of safety relevant function is discontinued to avoid harms due to hazard*
Fail Operational	The system continuous to fully operate in the presence of errors or faults, without any degradation *Service of safety relevant function if fully operational and provided*
Fail Degraded	The system continuous to operate in the presence of errors, accepting a partial degradation of functionality or performance during recovery or repair *Service of safety relevant function is limited operational but provided - except for the erroneous functionalities*

4.5 Potential product safety rating or metrics

Availability	readiness for correct service
Reliability	continuing for correct service
Maintainability	ability to undergo modifications and repairs
Integrity	absence of unreasonable hazardous functionality

Feature	Malfunction	Safety Goal	Dimension of required Safety			
			Availability	Reliability	Maintainability	Integrity
AP-FEAT-01	AP-HA-01	AP-SG-01	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope
AP-FEAT-06	AP-HA-06	AP-SG-02	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope
AP-FEAT-02	AP-HA-02	AP-SG-03	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope
AP-FEAT-04	AP-HA-04	AP-SG-04	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope
AP-FEAT-03	AP-HA-03	AP-SG-04	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope
AP-FEAT-05	AP-HA-05	AP-SG-04	Fail Operational	Fail Operational or Fail Degradation	Not in scope	Not in scope

Table 4.1: AUTOSAR Adaptive Platform - Hazards and derived Safety Goals

4.6 Dangerous Failures

Any failure which violates one of the Top Level Safety Goals above is considered to be dangerous.
Most common are - with respect to **AP-SG-01**, **AP-SG-02** & **AP-SG-03**:

- Hardware errors in CPUs, RAM, Flash or Bus of the MCU and
- any systematic and safety-relevant error in the SW (also of lower ASIL or QM if violating the freedom from interference)

With respect to **AP-SG-04**

- Electromagnetic interference on the communication lines,
- Hardware errors in communication hardware or
- Software errors in communication drivers which cause corruption, delay, loss, repetition, re-sequencing, insertion, or masquerading of messages (taken from ISO 26262-6 table D2.4).

4.7 Safe States

In general, the fail-operational requirement for the ADP requires mitigation measures for all failure modes. That means for each detected failure mode, a possible degraded or alternative similar functionality shall be available.

The safe state **"fail-silent"** shall only be used when no alternative functionality is reasonably possible.

The following states of the AUTOSAR Adaptive Platform shall only be considered safe states on the abstraction level of an ECU

AP-SAFE-1	Discontinuation of service (for fail-silent systems)
AP-SAFE-2	Continuation of service with failure indication (for fail-operational systems)

Note that on a higher abstraction level further safe states may exist (e.g. invalid qualifiers sent for specific data affected by an error) but those are not available on this abstraction level.

4.8 Fault-tolerant time interval

AP_FTTI_1	The fault-tolerant time interval (FTTI) It starts at the first transmission of a message influenced by a dangerous error. At its end the failure either shall be resolved by either the ECU operating correctly again and transmitting correct messages or the ECU adopting in one of the three safe states.
AP_FTTI_2	The time interval to detect latent faults Hence it is expected that at least every driving cycle the ECU on which ADP is hosted will be power cycled (i.e. shut down) and thus be enabled to execute start-up or shut-down self-tests. Any safety relevant failure detected due to these tests shall lead to a safe states.

4.9 Failure metrics

AP_FM_01	The standard ASIL D values for SPFM (99%) and LFM (90%) shall be supported by the ECU which hosts an ADP for ASIL D applications.
AP_FM_02	Alternatively, the standard ASIL-C values shall be supported by the ECU which hosts an ADP for ASIL C applications.
AP_FM_03	Alternatively, the standard ASIL-B values shall be supported by the ECU which hosts an ADP for ASIL B applications.
AP_FM_04	Alternatively, the standard ASIL-A values shall be supported by the ECU which hosts an ADP for ASIL A applications.
AP_FM_05	Alternatively, for ECU which hosts an ADP for QM applications no failure rates have to be considered.

5 Functional Safety Concept

5.1 Derived AUTOSAR Adaptive Platform top level functional safety requirements

From the architectural safety goals (4.2) and potential hazards (4.3) from chapter 4 and respecting the general Hardware and Software Fault Considerations (3.4) the following functional requirements can be derived easily.

5.1.1 Correct Execution (AP-SG-01)

By walking through the typical lifecycle of an ECU by starting with the initialization procedure the AUTOSAR Adaptive Platform shall provide:

- Safe and secure initialization [RS_SAF_0001]
- Authentication of applications and services [RS_SAF_0002]
- Validation of application prerequisites [RS_SAF_0003]
- Validation of application dependencies [RS_SAF_0004]

The safe and secure boot itself, is below the AUTOSAR Adaptive Platform Layer in the overall controller SW architecture and therefore not a part of the AUTOSAR Adaptive Platform architectural design and this investigation. Depending on the architectural decision of the final product developer the safety impact of the aforementioned tasks is difficult to rate, but considering dynamic deployment this might be necessary to maintain safety in environments supporting mixed criticality application deployment on the same partition.

After the prerequisites are met and dependencies are resolved the global context needs to be verified according to the information provided in the manifests and the following tasks evaluated:

- Calculate if all applications and services are schedulable [RS_SAF_0005]
- Calculate if all described resources are available [RS_SAF_0006]
- Calculate if all timing criteria are met [RS_SAF_0007]

with respect to the defined ASIL level and partitioning.

If all these start-up checks have been passed the AUTOSAR Adaptive Platform need to provide the following runtime capabilities:

- AUTOSAR Adaptive Platform shall provide safe scheduling [RS_SAF_0008]
- AUTOSAR Adaptive Platform shall provide separation and protection of mixed criticality Adaptive Applications to fullfil freedom from interference [RS_SAF_0009]

- AUTOSAR Adaptive Platform shall provide safe and reliable runtime monitoring [RS_SAF_0010]
- AUTOSAR Adaptive Platform shall provide control flow monitoring [RS_SAF_0019]
- AUTOSAR Adaptive Platform shall monitor if timing constraints are met and do not exceed the defined limits [RS_SAF_0011]
- AUTOSAR Adaptive Platform shall monitor if the memory usage does not exceed the defined limits [RS_SAF_0012]
- AUTOSAR Adaptive Platform shall monitor if the network usage does not exceeds the defined limits [RS_SAF_0013]
- AUTOSAR Adaptive Platform shall provide a mechanism to detect application behavior inconsistencies in a local and global scale [RS_SAF_0020]

If the underlying HW has the same ASIL rating as the SW, then safe computation seems to be expected and needs only to be investigated if the ASIL level of the HW is lower than required by the function. Several AUTOSAR Adaptive Platform mechanisms can be combined to achieve this goal, a short exemplary workflow is shown in 2.3.4.3. The AUTOSAR Adaptive Platform might not directly support this feature, but if this is known from the start, the customer specific implementation could respect this behavior in an easy fashion, in some cases maybe even transparent to application.

5.1.2 Safe Communication

During the runtime it could be expected that applications and services need to communicate with each other, not only on the same partition, but also through partition, controller, ECU borders and even with the off-board world. And additionally, dynamic deployment requires authentication of communication partners and therefore

- AUTOSAR Adaptive Platform shall provide an interface for an application or service to allow safe and secure communication [RS_SAF_0014]

with all the cybersecurity requirements connected to this task. The security specific sub-requirements, like secure key exchange, key storage etc. and even encryption is not directly considered safety related if they are correct developed and integrated in compliance to ISO26262 and with respect to common cybersecurity guidelines and standards.

If the communication is allowed by the security subsystem AUTOSAR Classic Platform already provides the End-2-End-Protection (E2E) concept for safe communication. Therefore

- AUTOSAR Adaptive Platform shall provide an interface for an application or service to generate E2E protected messages [RS_SAF_0015]

- AUTOSAR Adaptive Platform shall provide an interface for an application or service to send and receive E2E protected messages [RS_SAF_0016]
- AUTOSAR Adaptive Platform shall provide an interface for an application to service to validate E2E protected messages [RS_SAF_0017]

More information about E2E protection can be found in [25][26][27]

The application and service authentication could be initiated during initialization of the vehicle ECUs and the vehicle network. The required communication partners should be mention in the manifest of the ECU or the application.

- The Manifest shall contain information about communication partners (e.g. application dependencies, required services) [RS_SAF_0018]

If dependencies are not met, that application is not fully operationally, and based on the overall safety strategies, the full ECU is eventually not considered to be fully operational.

5.1.3 Safe Storage

It is also expected that applications and services require to load and store data persistently in a non-volatile memory unit. The AUTOSAR Adaptive Platform is also a means for isolation of the application from the hardware and peripheral interfaces (hardware abstraction), therefore

- AUTOSAR Adaptive Platform shall provide an interface for the application to load and store data persistently [RS_SAF_0021]
- AUTOSAR Adaptive Platform shall provide an interface for an external tester or programming device to load and store data persistently [RS_SAF_0022]
- AUTOSAR Adaptive Platform shall prevent or mitigate alteration of, loss in, delayed of data access or storage [RS_SAF_0023]

The AUTOSAR Adaptive Platform is hereby just providing an interface to the applications and services. The HW specific mechanisms are part of the platform specific implementation, e.g. if the NvM is an eMMC NAND Flash with wear-leveling, an EEPROM, NAND-, NOR-flash or FRAM, etc.

5.1.4 Safe Configuration and Update

The possibility for an external tester to modify the NvM without interacting with the application itself is just one part of safe configuration and update. The goal of the AUTOSAR Adaptive Platform is to provide means that applications can be deployed in the field and not only in workshops or even during production. To prevent an wrong application from being deployed in the first place the following tasks are necessary to maintain correct configuration.

- verify if an application is allowed to be deployed on the vehicle [RS_SAF_0024]
- verify if an application is allowed to be deployed on the ECU [RS_SAF_0025]
- verify if an application is allowed to be deployed on the dedicated resource [RS_SAF_0026]

Part of this verification is indeed to check if the local and global dependencies are met, the ASIL rating of the machine/partition has the proper classification etc. Finally all the check to ensure safe initialization and execution needs to be run before deployment, otherwise after the initialization, the system might end up in a failure mode. If the application is just optional, the impact might not be big because the application might just not get scheduled. If the application shall be an update, then the

- AUTOSAR Adaptive Platform shall mitigate or prevent unintended or incorrect alteration to, loss of a valid configuration [RS_SAF_0027]

The dynamic deployment feature has a big impact on every foundation module or service helping to fulfill the above mentioned roughly described safety requirements. Every foundation application or service needs either the possibility to get the configuration data from the manifests, and interpret this dynamically during initialization, activation of the new application or the vendor needs to update the machine configuration as an attachment to the updated application and impacted applications and services from the foundation. This is considered to be a customer specific behavior, and therefore implementation specific. This depends on how open the integration platform might be designed and if the vendor wants and can keep track of each configuration of each car in the field.

List of Figures

List of Tables