Document Title	Explanation of Safety Overview
Document Category	Explanation (EXP)
Document UID	895
Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R26-11

Document Change History

Date	Release	Changed by	Description
2026-11-26	R26-11	AUTOSAR Release Management	• Fixed formating issues • Added chapter 3.5.6
2025-11-27	R25-11	AUTOSAR Release Management	• Add CP contents
2024-11-27	R24-11	AUTOSAR Release Management	• Editorial changes • Update figure 3.6
2023-11-23	R23-11	AUTOSAR Release Management	• Editorial changes • Update figure 1.2 and 1.3 • add chapter 3.4 Security • update functional cluster in 6.2.x • update functional cluster list
2022-11-24	R22-11	AUTOSAR Release Management	• Changed AUTOSAR Standard to: Foundation • Glossary updated • Update chapter 3.2.5.1 • rephrase safety goal to safety needs
2021-11-25	R21-11	AUTOSAR Release Management	• updates in chapters 4 and 5, tables 4.1, 4.2, 5.2, 5.1 • chapter 4.2: remove malfunction, keep failure only • rename abbreviation AP-HA to TLF (Top Level Failure) • update table 5.2: Feature -> Use-Case • update table 5.2: Malfunction -> Failure • remove hazard from table title • remove references to ara:rest

△

2020-11-30	R20-11	AUTOSAR Release Management	• moved functional and technical safety requirements to `RS_Safety` • editorial changes • correction of typos and layout • updated abbreviation table • update of identified safety artifacts in AUTOSAR Adaptive Platform
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • editorial changes • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• No content changes • minor layout changes
2018-10-31	18-10	AUTOSAR Release Management	• restructuring of document inspired by ISO 26262 • rework chapters 1-5 • add functional safety requirements table
2018-03-29	18-03	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction 11
1.1 Purpose . . . 11
1.2 Scope . . . 11
1.3 Intended audience . . . 14
2 Assumption of Use and Objectives 15
2.1 Assumption of Use . . . 15
2.2 Design Objectives . . . 15
2.3 Scenarios . . . 17
2.3.1 Example of Classic Platform . . . 17
2.3.2 Example Scenario: HAD . . . 17
2.3.3 Example Scenario: Instrument Cluster . . . 17
2.4 Top Level Feature Requests or Use Cases . . . 17
3 System Description 19
3.1 Element Under Investigation . . . 19
3.2 Assumed System Context . . . 19
3.2.1 Vehicle Context . . . 19
3.2.2 ECU Context . . . 20
3.2.3 Microprocessor Context . . . 21
3.2.4 Hardware Accelerator . . . 22
3.2.5 Software Context . . . 23
3.2.5.1 Dynamic Memory Allocation . . . 23
3.3 General Hardware and Software Fault Considerations . . . 23
3.3.1 Potential Hardware Faults and Safety Measures . . . 23
3.3.2 Potential Software Faults and Safety Measures . . . 25
3.4 Security . . . 26
3.5 AUTOSAR Platform Architecture Overview . . . 26
3.5.1 AUTOSAR Classic Platform Features . . . 26
3.5.2 AUTOSAR Adaptive Platform Features . . . 26
3.5.3 AUTOSAR Classic Platform Achitecture and Modules . . . 27
3.5.4 AUTOSAR Adaptive Platform Architecture . . . 27
3.5.5 AUTOSAR Adaptive Platform Functional Cluster . . . 28
3.5.6 Need for Runtime Supervision in AUTOSAR Adaptive Platform . . . 29
3.6 Configuring ASILs in AUTOSAR . . . 31
4 Hazard Analysis 33
4.1 Introduction . . . 33
4.2 Top Level Failures . . . 33
5 Safety Needs 35
5.1 Top Level Safety Requirements . . . 35
5.2 Potential Product Safety Quality Attributes . . . 35

6 Functional Safety Concept 37
6.1 Derived AUTOSAR Platform Functional Safety Requirements 37
6.1.1 Safe Execution 37
6.1.2 Safe Communication 38
6.1.3 Safe Storage 38
6.1.4 Safe Configuration and Update 38
6.2 Safety Artifacts of the AUTOSAR Platform 39
6.2.1 Ensure correct computation, execution and execution order of multiple applications with mixed criticality 39
6.2.1.1 Classic Platform Autosar 40
6.2.1.2 Adaptive Platform Autosar 40
6.2.2 AUTOSAR shall ensure correct configuration during the entire life cycle of the platform 40
6.2.2.1 Classic Platform Autosar 40
6.2.2.2 Adaptive Platform Autosar 40
6.2.3 AUTOSAR shall ensure correct update and upgrade of multiple platform and non-platform applications with mixed criticality 41
6.2.3.1 Classic Platform Autosar 41
6.2.3.2 Adaptive Platform Autosar 41
6.2.4 AUTOSAR shall ensure correct exchange (transmission and reception) of information 41
6.2.4.1 Classic Platform Autosar 41
6.2.4.2 Adaptive Platform Autosar 41
6.2.5 AUTOSAR shall detect faults and failures while processing data, communicating with other systems or system elements 42
6.2.5.1 Classic Platform Autosar 42
6.2.5.2 Adaptive Platform Autosar 42
6.2.6 AUTOSAR shall provide mechanisms to support safe storage for applications 42
6.2.6.1 Classic Platform Autosar 42
6.2.6.2 Adaptive Platform Autosar 42
6.2.7 AUTOSAR shall monitor, detect and provide means to react on detectable failures 43
6.2.7.1 Classic Platform Autosar 43
6.2.7.2 Adaptive Platform Autosar 43
A Abbreviations 44
B Glossary 45

References

[1] ISO 26262:2018 Road vehicles -– Functional Safety
https://www.iso.org

[2] Utilization of Crypto Services
AUTOSAR_CP_EXP_UtilizationOfCryptoServices

[3] Glossary
AUTOSAR_FO_TR_Glossary

[4] Virtual Functional Bus
AUTOSAR_CP_TR_VFB

[5] Layered Software Architecture
AUTOSAR_CP_EXP_LayeredSoftwareArchitecture

[6] AUTOSAR Introduction
https://www.autosar.org/fileadmin/ABOUT/AUTOSAR_Introduction.pdf

[7] Explanation of Adaptive Platform Design
AUTOSAR_AP_EXP_PlatformDesign

[8] Overview of Functional Safety Measures in AUTOSAR
AUTOSAR_CP_EXP_FunctionalSafetyMeasures

[9] Specification of Operating System Interface
AUTOSAR_AP_SWS_OperatingSystemInterface

[10] Specification of Execution Management
AUTOSAR_AP_SWS_ExecutionManagement

[11] Design guidelines for using parallel processing technologies on Adaptive Platform
AUTOSAR_AP_EXP_ParallelProcessingGuidelines

[12] Explanation of Safe API for hardware accelerators
AUTOSAR_AP_EXP_SafeHardwareAccelerationAPI

[13] General Specification of Adaptive Platform
AUTOSAR_AP_SWS_General

[14] Basic Software Module Description Template
AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate

[15] Mapping mixed-criticality applications on multi-core architectures
https://doi.org/10.7873/DATE.2014.111

[16] ISO 21434:2021 – Road vehicles – Cybersecurity engineering
https://www.iso.org

[17] IEEE Standard for Information Technology- Standardized Application Environment Profile (AEP)-POSIX Realtime and Embedded Application Support
https://standards.ieee.org/findstds/standard/1003.13-2003.html

[18] API standards for Open Systems
http://www.opengroup.org/austin/papers/wp-apis.txt

[19] Standardized M1 Models used for the Definition of AUTOSAR
AUTOSAR_FO_MOD_GeneralDefinitions

[20] Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform
AUTOSAR_FO_RS_Safety

[21] Data Integrity Pattern
http://en.wikipedia.org/wiki/Data_integrity

Known Limitations

This explanatory document may contain assumptions, exemplary items, like reference models, use-cases, scenarios, and/or references to exemplary technical solutions, devices, processes or software. Any such assumptions or exemplary items contained in this document are for illustration purposes only. These assumptions are not part of the AUTOSAR standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance products actually implementing such exemplary items, imply that intellectual property rights covering such items or assumptions are licensed under the same rules as applicable to the AUTOSAR standard.

No ASIL Ratings

The AUTOSAR consortium is only providing an architecture definition, descriptions of the functional blocks and a *proof of concept* implementation, it is not possible to add ASIL ratings to each architectural item in this scope. It is only possible to give the reader 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.

SEooC according to ISO26262 part 10

If the AUTOSAR definition itself can be considered being a SEooC according to ISO 26262 part 10 is still unresolved and not verified yet. According to the definition of an item, element or architecture from the ISO 26262 part 1, an architecture - in this case the software architecture - is a representation of the structure of the *item* or *element* and an *element* could be a *system*, a *software component* or a *software unit*, which eventually might also be an SEooC. Either way, following the ISO 26262 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 a common starting point. Still, the AUTomotive Open System ARrchitecture will eventually be the basis for a software component, which could be considered as an *element* and SEooC according to ISO 26262 part 10. The goal of the AUTOSAR architecture is to enable and support systems up to ASIL D.

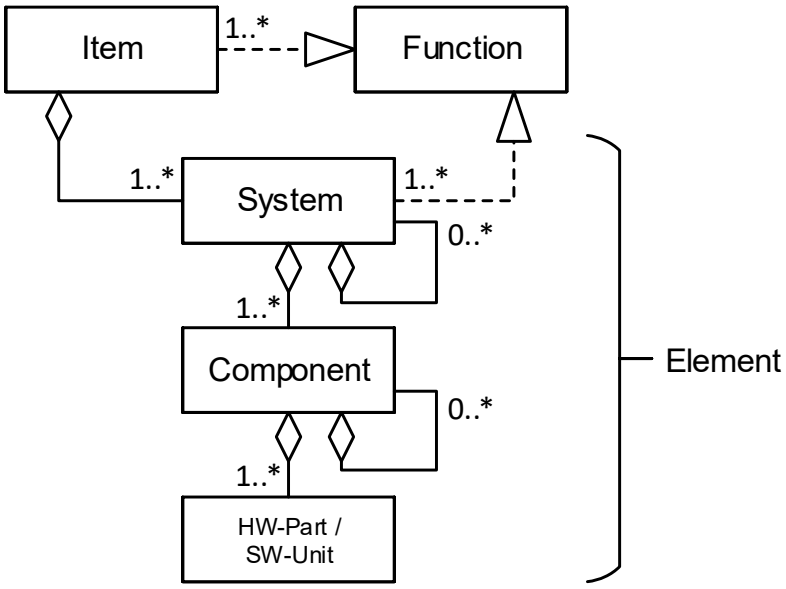

Figure 1: Relationship of item, system, component, hardware part and software unit, Figure 3 - ISO 26262-10 [1]

Cybersecurity

For autonomous driving, cybersecurity is expected to have a greater impact than in the past. Not only that communication channels and communication partners need to be authenticated and verified, they also need to be safe. The security concept and capabilities of the AUTOSAR can be found in the explanatory documentation [2]. This explanatory document, the AUTOSAR_EXP_SafetyOverview, contains only safety topics. It is the responsibility of the corresponding project-team, to decide if their specific safety goals can be realized with state-of-the art cybersecurity measures. Some security related safety features could be:

- Secure boot
- Authentication of communication partners within the vehicle network as well as with the off-board world
- Secure key exchange
- Secure key storage
- ...

The security specific algorithms like encryption, decryption and signing are not directly considered safety related, they still need to be developed and integrated in compli-

ance to ISO 26262 and with respect to cybersecurity guidelines and standards e.g. ISO 21434.

Completeness

This document might not cover all possible scenarios in which AUTOSAR Platforms 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 AUTOSAR Working Groups, contributors and reviewers.

1 Introduction

1.1 Purpose

Functional safety is a system characteristic which is taken into account from the beginning of the development of the AUTOSAR Platforms as it may influence system and software architectural design decisions. Therefore, the AUTOSAR Platforms 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 industry.

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 ease the achievement of functional safety. The AUTOSAR Platforms support the development of safety-related systems by offering safety measures and mechanisms.

However, the AUTOSAR Platforms are not a complete safe solution. The objective of this safety overview is to derive safety requirements from the top level safety requirements and assumed use-cases or scenarios and allocate them to the architectural elements of the item, or to any external measure. The use of the AUTOSAR Adaptive Platform does not imply ISO 26262-10 compliance. It is still possible to build unsafe systems using the AUTOSAR Platforms safety measures and mechanisms. The architecture of the AUTOSAR Platforms can, in the best case, only be considered to be an SEooC.

Information about the AUTOSAR Platforms 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 Platforms. The content of this document is structured into separate chapters as follows:

- AUTOSAR Platform objectives, use-cases and scenarios
- System definition, system context and assumptions
- Hazard analysis
- Safety Needs

- Functional safety concept

which could be mapped to the following chapters within the ISO 26262, figure 1.1:

- **[3-5]** Item definition
- **[3-6]** Hazard analysis and risk assessment
- **[3-7]** Functional safety concept

as visualized in figure 1.2. Safety requirements are hierarchically structured and assigned or referenced from hazard to safety goal to functional requirement and artifact, according to ISO 26262[1], as illustrated in figure 1.3. The development process and organizational topics are not part of this overview, a risk assessment is not done (see chapter Known Limitations) every system description, scenario or use-case in this document are just explanatory and *for reference only*. The system design is out of scope!

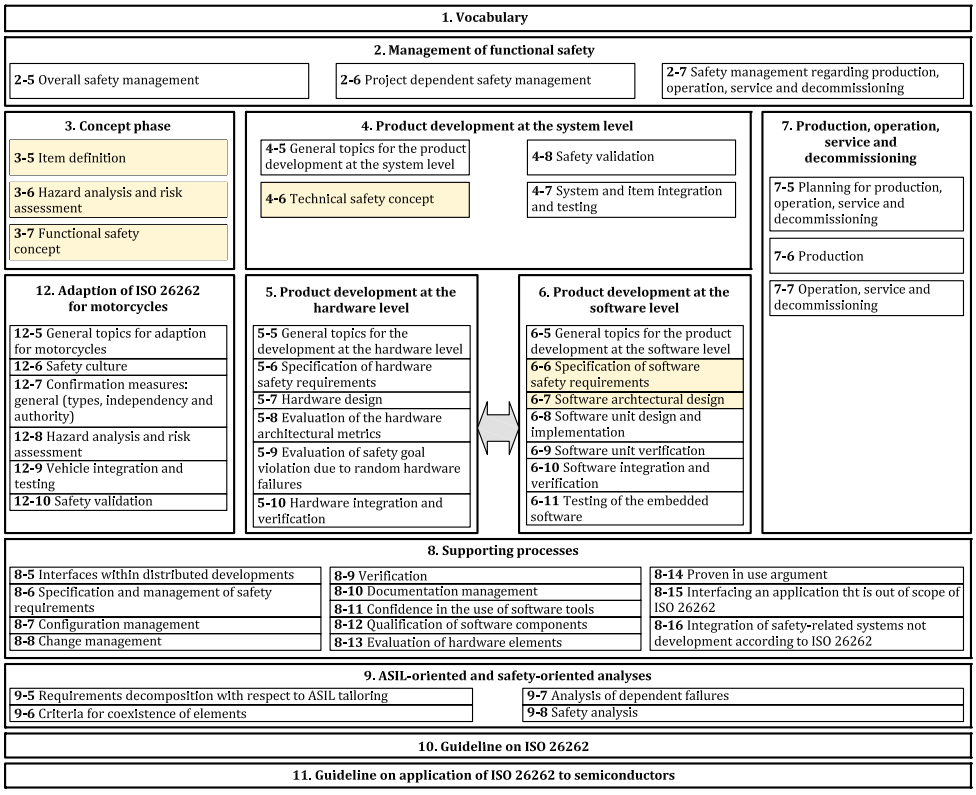

Figure 1.1: Considered chapters of ISO 26262, Overview of the ISO 26262 series of standards, Figure 1 - ISO 26262-1 [1]

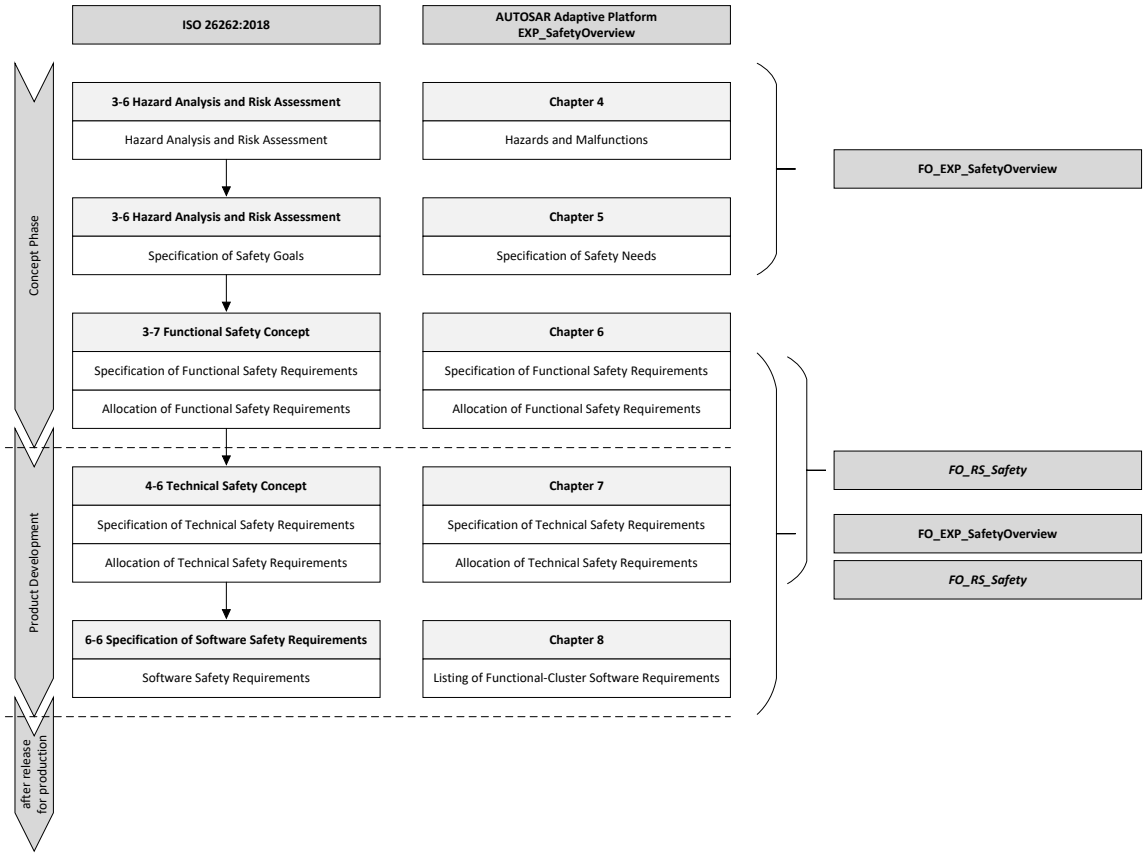

Figure 1.2: Structure of safety requirements and mapping to this Document, based on ISO 26262 [1]

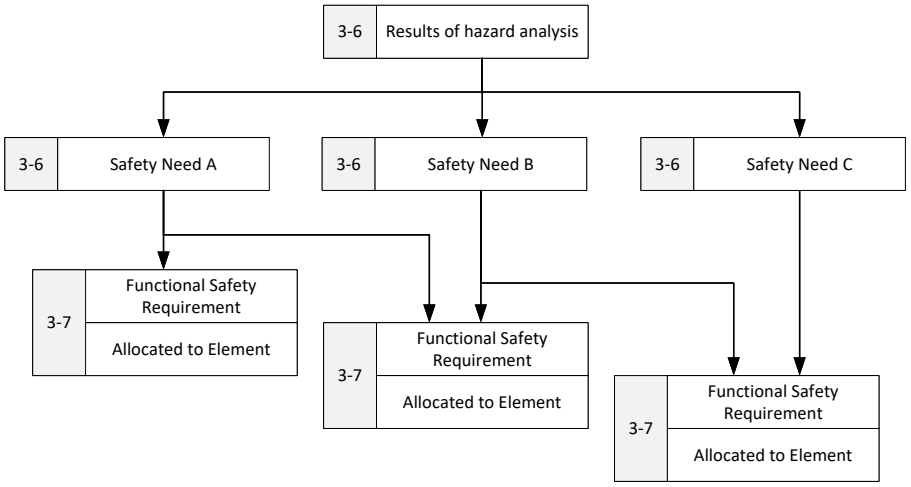

Figure 1.3: Hierarchy of safety goals and functional safety requirements

1.3 Intended audience

This document shall provide an overview of the functional safety measures and mechanisms of the AUTOSAR Platforms 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 Platforms, including people involved in safety analysis. AUTOSAR specific and functional safety related glossary terms are covered by the AUTOSAR Glossary [3] or the ISO 26262 [1] itself, and are not copied if no additional information or interpretation hint related to this document is necessary.

2 Assumption of Use and Objectives

2.1 Assumption of Use

Assumptions of use for the AUTOSAR Platforms are in particular, but not limited to, automotive grade electronic control units from the following domains:

- Autonomous Driving: from driver assistance to fully automated driving, including the ecosystem of AD, ADAS and/or Sensor-ECUs where applicable,
- Gateways,
- Body-Domain Controller,
- Infotainment-systems, etc.

To solve the requirements for more processing power, e.g. for sensor-data processing (images, radar), 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, often realized through specialized and proprietary hardware components and software interfaces.

2.2 Design Objectives

The overall design objectives of the AUTOSAR Adaptive Platform are similar to those of the well known and established AUTOSAR Classic Platform, and therefore describes layers of abstraction, interfaces and some common behavior of an automotive software for an electronic control unit. The AUTOSAR Adaptive Platform is still providing an abstraction layer for the software developers e.g. AUTOSAR Runtime for Adaptive Applications (ARA), so that AUTOSAR Adaptive Platform applications could be exchanged between ECUs or being ported easily. From a systematic viewpoint this is similar to the AUTOSAR Classic Platform BSW and VFB layer - as described in AUTOSAR Classic Platform architecture documentation [4] [5], and shown for comparison in figures 2.1 and 2.2.

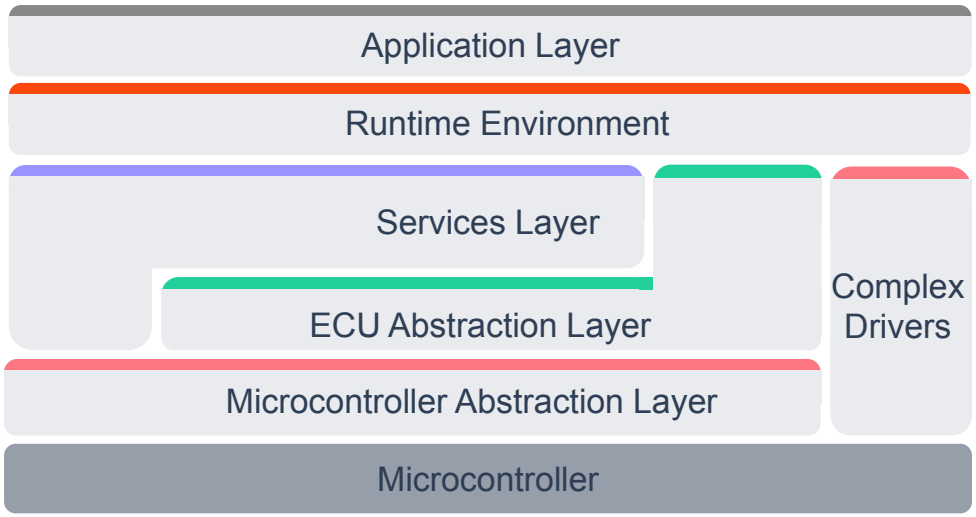

Figure 2.1: AUTOSAR Classic Platform layered architecture [6]

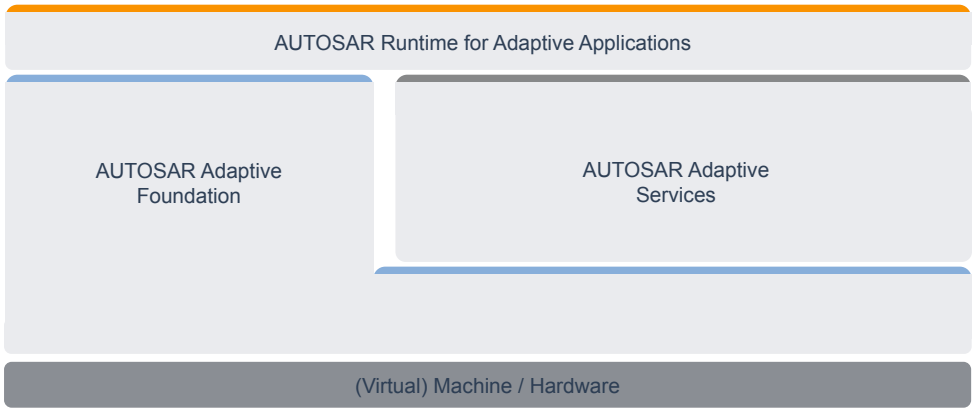

Figure 2.2: AUTOSAR Adaptive Platform layered architecture [6]

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

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 ISO 26262 compliant at all (or QM at max), 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 [6] and the explanatory AUTOSAR Adaptive Platform Design document [7].

2.3 Scenarios

2.3.1 Example of Classic Platform

For scenarios of Autosar Classic Platofrm please see [8].

2.3.2 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 ISO 26262 [1]. The system design on vehicle level is assumed to contain several sensors, being directly connected to sensors or Sensor-ECUs (e.g. radar, lidar, vision, INS, GNSS). The vehicle is expected to have at least one ADAS-ECU for the autonomous driving functionality where AUTOSAR Adaptive Platform could be integrated, not only on that ADAS-ECU, but also on the Sensor-ECUs or any other before mentioned domain controller.

2.3.3 Example Scenario: Instrument Cluster

Another example which is not as safety critical as HAD, but can be rated with an ASIL, is an instrument cluster. While the instrument cluster is not as safety critical as HAD, it is also not as trivial as an infotainment system.

Let's consider the use case where the speedometer gives a wrong speed and the driver drives well above the speed limit, risking himself as well as the rest of the traffic. Another critical scenario may occur when a failure indication is not turned on e.g. brake failure, airbag failure or an engine failure.

As the state of the art in the automotive industry advances, the instrument cluster would require high performance. Integrating an instrument cluster on AUTOSAR Adaptive Platform would naturally make sense to cater the high performance requirements. In turn, AUTOSAR Adaptive Platform should ensure functional safety requirements.

2.4 Top Level Feature Requests or Use Cases

Based on the initial stakeholder analysis and AUTOSAR consortium partner requirements the following feature requests according to the intended use and scope of the AUTOSAR Platforms have been identified:

[SUC_01]	Provide flexible execution time and resources for multiple, mixed criticality applications.
[SUC_02]	Provide dynamically configurable, updateable and upgradable runtime for multiple, mixed criticality applications.
[SUC_03]	Provide information exchange between multiple, mixed criticality applications.
[SUC_04]	Provide information exchange between mixed criticality application and other external components such as sensors, actors or ECUs inside the vehicle.
[SUC_05]	Provide information exchange between mixed criticality application and other external components outside the vehicle.
[SUC_06]	Maintain correct configuration and monitor correct operation during the driving cycle.

Table 2.1: Top Level Safety Use-Cases or feature requests

3 System Description

3.1 Element Under Investigation

The Element under investigation in this explanatory document is the AUTOSAR Platform architecture running in a system-context roughly described in chapter 3. The AUTOSAR Platform architecture will eventually be the basis for a software component, which could be considered as an *element* and SEooC according to ISO 26262-1 and ISO 26262-10.

The AUTOSAR Platforms are intended to be solution independent, except for the fact that it is developed for the automotive industry and according to objectives described in chapter 2. Still, the platform it will be executed on needs to be investigated too, in order to derive some hazards and safety requirements. Some of which will eventually be satisfied by software features as described and defined in the AUTOSAR Platform architecture, others by the OEM or their suppliers respectively. 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 ISO26262, if the embedded software consists of software components with different ASIL ratings, then 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 or equal ASIL rating, even or especially if decomposed from the functionality of a higher ASIL, e.g. 2×ASIL B(D).

3.2 Assumed System Context

The following system-context descriptions are just educated guesses and assumptions, necessary for derivation and explanation of the safety requirements.

3.2.1 Vehicle Context

At the time of the initial definition of the AUTOSAR Adaptive Platform high performance processing units developed as SEooC 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.

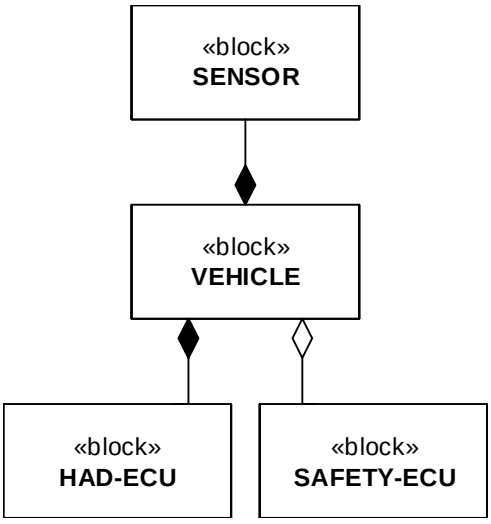

Figure 3.1: Exemplary simplified vehicle system

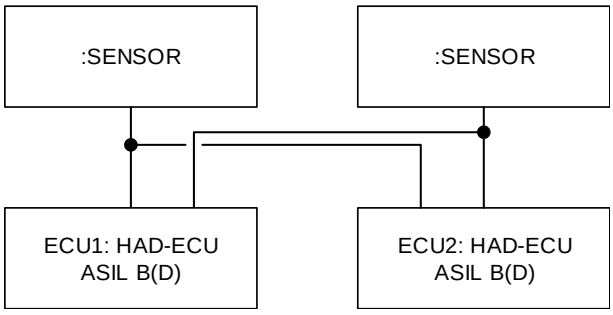

Figure 3.2: Systematic redundancy

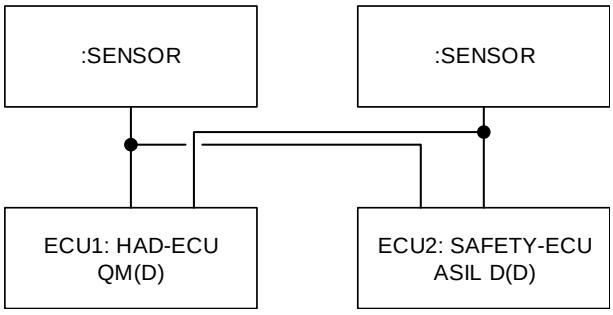

Figure 3.3: Decomposition with safety checker

The vehicle system design is **not** part of the AUTOSAR Platform specification, still either option (3.2 and 3.3) could be a valid system setup. It is up to the final product developer and safety engineer to choose a proper system design and decomposition strategy to achieve the specific safety goals and fulfill the specific safety requirements.

3.2.2 ECU Context

In a typical safety compliant ECU it can be assumed that, besides a 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.

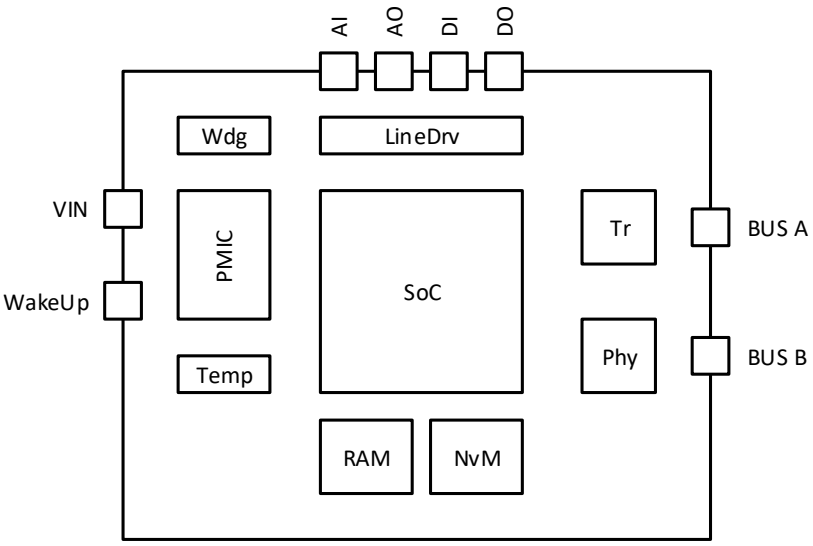

Figure 3.4: Exemplary draft of a simple ECU design

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 been triggered, the line driver and the transceivers might be disabled, to achieve the `Fail-Silent` behavior without software interaction.

3.2.3 Microprocessor Context

A Microprocessor or SoC design could look like the one shown in figure 3.5

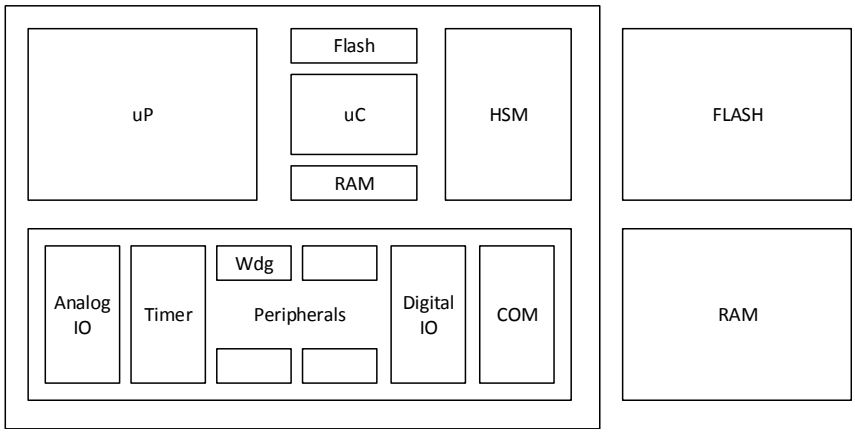

Figure 3.5: Exemplary draft of a simple MCU design

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. These can typically be based on AUTOSAR Classic Platform. 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 at the same level as it is in the AUTOSAR Classic Platform. The only hardware requirements from the AUTOSAR Adaptive Platform are indirectly defined through the OS, which shall provide multi-process support for isolation of applications and therefore requires a Memory Management Unit (MMU) according to Specification of Operating System Interface [9] and Specification of Execution Management [10]. 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 hardware designs (as of 2018).

3.2.4 Hardware Accelerator

Hardware accelerators and parallel processing are respected within the AUTOSAR architecture. For more information regarding this topic please read the "Design guidelines for using parallel processing technologies on Adaptive Platform [11]" and "Explanation of Safe API for hardware accelerators [12]". The software development process and the required software mechanisms for a hardware accelerator are basically the same as for the typical Microprocessor. There shall be mechanisms to check if software routines are scheduled correctly, the computations are correct and the control flow shall be monitorable.

3.2.5 Software Context

3.2.5.1 Dynamic Memory Allocation

Using dynamic objects or variables in safety relevant code is not recommended according to ISO 26262-6 Table 6 [1], or else online/run-time tests are required during their creation.

Dynamic memory allocation is inferred by some of the Adaptive platform APIs. Provided that Adaptive Platform vendors and Adaptive application ensure proper error handling and cleanup in case of allocation failure, and that when running safety relevant code the memory allocation and deallocation functions (e.g. malloc and free, new and delete) have deterministic performance, meaning that either their worst execution / blocking time is a known value, or a dedicated safety mechanism such as a watchdog is applied to handle timing violations, then they are allowed to use dynamic memory allocation in safety relevant code.

Please consider [AP_SWS_GENERAL_00023] General Specification of Adaptive Platform [13], which states that the implementation of the AUTOSAR Adaptive Platform treats a failed default allocation as non-recoverable. For Classic Platform please consider subchapters [ResourceStaticMemory] and [sec:DynamicMemoryNeeds] of [14].

3.3 General Hardware and Software Fault Considerations

The hardware is not part of the AUTOSAR Platform architecture, it is still necessary to respect the hardware to define the source of higher safety requirements eventually. This section is to be considered as general a priori knowledge and collects and describes typical hardware and software faults along with the safety measures which might directly affect the AUTOSAR Platforms. 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 Platforms according to the relevant industry standards.

3.3.1 Potential 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 (Mapping mixed-criticality applications on multi-core architectures [15]).

Hardware partitioning based on safety criticality of AUTOSAR 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 sporadically, 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 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 writeability)
- 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)

3.3.2 Potential 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 AUTOSAR Platform. 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.

3.4 Security

Security and Functional Safety are often considered two different concepts: functional safety *primarily* deals with unintended faults and errors and security *primarily* deals with intended threats leading to a harm or loss. Nevertheless, in a connected system, security is very important to create safe systems and achieve a high product quality. Security mechanisms needs to be developed according the provided standard, e.g. ISO 21434 Road vehicles - Cybersecurity engineering [16], and during runtime, monitoring of the security mechanisms is highly recommended to check that the security systems and mechanisms are working and running in the appropriate state.

3.5 AUTOSAR Platform Architecture Overview

3.5.1 AUTOSAR Classic Platform Features

Classic Autosar offers following features - independent of applications:

- Configuration if ECU and application data
- Execution of applications
- Scheduling of applications
- Application state management
- Runtime behavior monitoring: processing time and sequence, bus load, memory consumption, HW, etc.
- Safe and secure communication through vehicle networks: e.g. CAN, CAN-FD, FlexRay, Ethernet
- Diagnosis (UDS,OBD)
- Persistent data storage

3.5.2 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 2.2 (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, Ethernet

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

3.5.3 AUTOSAR Classic Platform Achitecture and Modules

For layered Classic platform Architecture and Modules see [5].

3.5.4 AUTOSAR Adaptive Platform Architecture

The architecture of the AUTOSAR Adaptive Platform is shown in figure 3.6 and is divided into

1. AUTOSAR Adaptive Platform Foundation Functional Clusters (FCs)
2. AUTOSAR Adaptive Platform Services FCs
3. AUTOSAR Adaptive Platform Standardized Application/Interface FCs
4. AUTOSAR Adaptive Platform Vehicle Service FCs
5. Operating System Interface

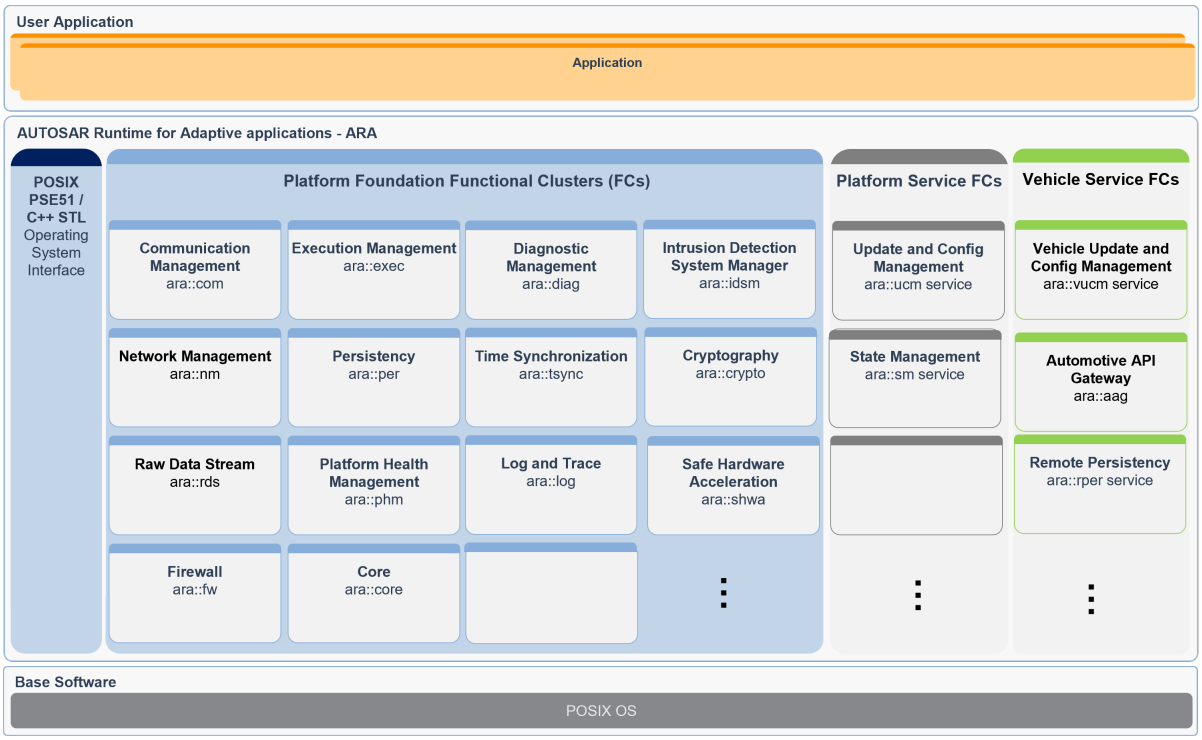

Figure 3.6: AUTOSAR Adaptive Platform Logical Architecture - Safety View [7]

The operating system (OS) itself is not directly part of the architecture, but the AUTOSAR Adaptive Platform has several requirements regarding the OS [10], like being a POSIX PSE51 compliant OS [17][18].

3.5.5 AUTOSAR Adaptive Platform Functional Cluster

The AUTOSAR Adaptive Platform functional clusters [19] are

Adaptive Platform Functional Cluster	Short Name
Log and Trace	log
Intrusion Detection System Manager	idsm
Communication Management	com
Network Management	nm
Operating System Interface	osi
Cryptography	crypto
Execution Management	exec
Raw Data Stream	rds
Firewall	fw
Platform Health Management	phm
Vehicle Update and Configuration Management	vucm
Persistency	per
Core	core
Time Synchronization	tsync
State Management	sm
Update and Configuration Management	ucm
Automotive API Gateway	aag
Diagnostics	diag

Table 3.1: AUTOSAR Adaptive Platform Functional Cluster

The detailed description for the AUTOSAR Adaptive Platform functional clusters can be found in the respective specialized documents. A summary is also part of the "Explanation of Adaptive Platform Design [7]."

3.5.6 Need for Runtime Supervision in AUTOSAR Adaptive Platform

Runtime supervision is used as measure of functional safety according to ISO 26262-6 (Product development at the software level), to detect loss of execution, timing faults and incorrect execution sequence.

In safety relevant systems (brake, servo steering, damping) the software computes and uses safety relevant values. It is essential that the computation fulfills safety aspects (safe coding, monitoring) and that the computed values are not obsolete (too old).

Example: A servo steering system computes the power the electrical engine has to provide to support the movement of the frontwheels. The intended power depends on the movement of the steering wheel imposed by the driver, the speed of movement and many environmental values of the vehicle. One of these environmental data is the vehicle speed. The movement power depends on the vehicle speed and it makes a large difference if the car is moved at pedestrian speed (parking) or on a highway. As the power of the electrical engine is about 1500 Watt the driver has little chance to act against it if a wrong power value is computed. For this reason the vehicle speed as an input parameter is safety relevant. It must be computed safely and be available in time. If the computation uses a speed value which is two seconds old we have a safety problem because the speed (and driving direction) might have changed in the last two seconds. So the system has to supervise if the computation runs correctly

and provides results in time. This is provided by runtime supervision and realized by platform health management in AUTOSAR.

Platform health management offers three types of supervision:

Alive supervision:

This addresses the supervision of cyclic processes. Platform Health Management checks cyclically if the process reports that it is still running. If this cyclic reporting of "running" is available to Platform Health Management then it assumes that it provides the expected safety relevant results. If this reporting of "running" is missed then Platform Health Management expects that the process is stuck or terminated and can no longer provide these results. In this case Platform Health Management triggers a safety reaction.

Deadline supervision:

Deadline supervision addresses the topic, whether the execution time of steps (of a supervised entity) is within its two limits of minimum and maximum. Supervision of minimum time is useful if it is clear that process needs a certain time to compute a valid result. If it terminates earlier the results may be void. Supervision of maximum time checks if the process is running longer than defined. In this case the results may come too late. Both issues - too short and too long runtime - are safety issues and Platform Heath Management triggers a safety reaction.

Logical supervision:

Logical supervision addresses the sequence of executions. This is essential when one process (e.g., Process A) produces results that another process (e.g., Process B) depends on for its computation. Some systems use redundant computation of values. The main process computes a value and stores it in memory. The plausibility check process computes a value - maybe with a different algorithm i.e. the redundant computation. After this computation the plausibility check process compares the result to the value stored by the main process and triggers a safety reaction if the difference of both values is above a threshold value. In this case the sequence of the main and the plausibility check process is essential. If the plausibility check process runs first it may use the main process' result which was computed too long ago. If the processes set up for a logical supervision do not run in the expected order, Platform Health Management triggers a safety reaction.

Triggering a safety reaction:

If Platform Health Management detects a fault, it informs the State Management about this fault and leaves the reaction to this fault up to State Management. Now State Management has the chance to react e.g., by switching function group states and hereby triggering a restart of processes, that have caused the fault.

If State Management is not able to handle the fault it gives a negative response to Platform Health Management which then triggers a watchdog reset.

In case that State Management does not reply to Platform Health Management at all a timer of Platform Health Management expires and causes Platform Health Management to trigger a watchdog reset.

The case that Platform Health Management itself does not run properly anymore is handled by the mandatory cyclic call of the Watchdog Interface. If the Watchdog Interface does not receive the cyclic expected Alive Notification in time, it starts a watchdog reset.

The ISO 26262 defines the Fault Handling Time Interval as the sum of two time intervals. The time to detect a fault and the time to react to this fault. Reaction to a fault means to transfer the system into a state which cannot cause harm. The fault handling time interval must be considered for runtime supervision by Platform Health Management.

Conclusion:

Platform Health Monitoring is essential for monitoring safety relevant software. The software of Platform Health Monitoring must be developed with the same ASIL (or higher) as the safety relevant applications running on the platform.

3.6 Configuring ASILs in AUTOSAR

This specification is intended to support the Automotive Safety Integrity Level (ASIL) of ISO 26262 [1]. Other safety integrity levels will not be considered and are out of scope of this document. The ASIL is determined as part of the HARA in the concept phase as of ISO 26262-3 and assigned to each safety goal. A system element, software element or hardware element will inherit this ASIL as an attribute via the allocation of safety requirements to safety goals through the system hierarchy.

The ASIL is stored as `AdminData` that contains `Sdg` data element with the attribute `gid="SAFEX"`. The contents of this element shall contain a `Sd`-element with the attribute `gid="ASIL"`.

Valid values for this attribute are:

- QM
- A
- B
- C
- D
- QM(A)
- QM(B)
- QM(C)

- QM(D)
- A(A)
- A(B)
- A(C)
- A(D)
- B(B)
- B(C)
- B(D)
- C(C)
- C(D)
- D(D)

Note that the parentheses notation is used to express decomposed safety requirements. In this specification we will refer to the original ASIL (i.e. the value in parentheses) as the contextual ASIL before decomposition, since it belongs to the context of safety goal.

Listing 5.1: Example for the AUTOSAR XML representation of an ASIL attribute at an element

```
<!-- Example AUTOSAR element with ASIL -->
<APPLICATION-SW-COMPONENT-TYPE>
<SHORT-NAME>MyComponent</SHORT-NAME>
<ADMIN-DATA>
<SDGS>
<SDG GID="SAFEX">
<SD GID="ASIL">B</SD>
</SDGS>
</ADMIN-DATA>
<PORTS>
```

4 Hazard Analysis

4.1 Introduction

Any failure or malfunction which violates the safety goals is considered to be dangerous.
Most common safety related failures or malfunctions are

- Hardware errors in CPUs, RAM, Flash or Bus of the MCU and their peripherals
- Any systematic and safety-relevant error in the software (also of lower ASIL or QM if violating the freedom from interference)
- Electromagnetic interference on the communication lines
- Hardware errors in communication hardware components
- Software errors in communication drivers which cause corruption, delay, loss, repetition, re-sequencing, insertion, or masquerading of messages (taken from ISO 26262-6 clause D2.4).

Based on the initial hardware software fault considerations from chapter 3.3, the above mentioned failure sources and the safety goals, as well as the ISO 26262, which provides examples for faults which cause interference between software components, faults can be grouped as follows:

- Memory,
- Timing,
- Execution,
- Exchange of information,
- Authentication of applications and services,
- Rights management.

4.2 Top Level Failures

The top level safety related failures for the AUTOSAR Platforms considered to be

[TLF_01]	Unintended, untimely and/or incorrect execution of applications
[TLF_02]	Unintended, untimely and/or incorrect configuration, update and upgrade of applications
[TLF_03]	Unintended, untimely and/or incorrect exchange of information between applications
[TLF_04]	Unintended, untimely and/or incorrect exchange of information between applications and external components inside the vehicle
[TLF_05]	Unintended, untimely and/or incorrect exchange of information between applications and external components outside the vehicle
[TLF_06]	Corruption of configuration

Table 4.1: Top Level Failures

5 Safety Needs

5.1 Top Level Safety Requirements

The AUTOSAR Platforms are only a part of "larger" item definition, as explained in the chapters before, the architecture will eventually form the basis of a real software component, which might correspond to the element definition of an SEooC [1].

[RS_SAF_00001] Safe Execution:
AUTOSAR shall provide supporting mechanisms to monitor the control flow and manage the execution order of multiple applications with mixed safety criticality.

[RS_SAF_00002] Safe Configuration:
AUTOSAR shall provide mechanisms to support correct configuration during the entire driving cycle of the vehicle.

[RS_SAF_00003] Safe Update or Safe Upgrade:
AUTOSAR shall provide mechanisms to support correct update and upgrade of multiple platform and non-platform applications with mixed criticality.

[RS_SAF_00004] Safe Exchange of Information:
AUTOSAR shall provide mechanisms to support safe exchange (transmission and reception) of information between safety critical applications.

[RS_SAF_00005] Detection of Data Corruption:
AUTOSAR shall provide mechanisms to detect faults and failures while processing data, communicating with other systems or system elements.

[RS_SAF_00006] Safe Storage:
AUTOSAR shall provide mechanisms to support safe storage for applications.

[RS_SAF_00007] Recovery upon failure:
AUTOSAR shall monitor, detect, and provide means to react on detectable failures.

Table 5.1: Top Level Safety Requirements [20]

All Top Level Safety Requirements shall be achievable up to ASIL D. ASIL D Fail-operational qualities shall be achievable, even if one of the Top Level Safety Requirements is violated wherever applicable.

5.2 Potential Product Safety Quality Attributes

Use-Case	Failure	Safety Requirement	Dimension of required safety			
			Availability	Reliability	Maintainability	Integrity[1]
SUC_01	TLF_01	[RS_SAF_00001]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope
SUC_06	TLF_06	[RS_SAF_00002]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope
SUC_02	TLF_02	[RS_SAF_00003]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope

[1] AUTOSAR is not responsible for the safety integrity of the host application

SUC_04	TLF_04	[RS_SAF_00004]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope
SUC_03	TLF_03	[RS_SAF_00004]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope
SUC_05	TLF_05	[RS_SAF_00004]	Fail Operational	Fail Operational Fail Degradation	Not in scope	Not in scope

Table 5.2: Use-Cases, Failures and and derived safety requirements

6 Functional Safety Concept

6.1 Derived AUTOSAR Platform Functional Safety Requirements

From the architectural top level safety requirements (5.1) and potential hazards (4.2) from the previous chapters 4 and 5 and respecting the general Hardware and Software Fault Considerations (3.3) the following functional requirements can be derived by walking through the typical life cycle of an ECU and simple categories: *safe execution*, *safe communication*, *safe storage*, and *safe configuration and update*.

6.1.1 Safe Execution

Starting with the initialization procedure:

- Safe initialization needs to be taken into consideration [RS_SAF_10001]
- Check integrity of applications and services [RS_SAF_10002]

Information: The safe boot itself, is according to the Layered Architecture, below the AUTOSAR Platform Layer and therefore not part of the AUTOSAR Platform architectural design and the scope of this safety related investigation. The vigilant safety engineer shall still be aware that the integrity needs to be verified before starting the corresponding partition.

Depending on the architectural decision of the final product and its environment, the safety impact of the aforementioned tasks is difficult to rate. Considering dynamic deployment possibilities of AUTOSAR Adaptive Applications, these safety functions might be necessary to be executed during initialization in order maintain safety in environments supporting dynamic configurations of mixed criticality applications deployed on the same partition. If only pre-verified configurations are allowed to be uploaded to the system in a safe way, only integrity checks are required during startup to ensure that the applications have not been altered.

If all these start-up checks have been passed the following runtime capabilities needs to be provided:

- Safe resource management to achieve freedom from interference [RS_SAF_10008]
- Dependable scheduling for applications and services [RS_SAF_10028]
- Safe program execution [RS_SAF_10030]
- Defined program execution time [RS_SAF_10031]
- Separation of applications and services [RS_SAF_10008]
- Protection of applications and services [RS_SAF_10008]
- Safe shutdown of application and services [RS_SAF_10005]
- Safe transition of states in an application/service life cycle [RS_SAF_10006]

Information: If the underlying hardware has the same ASIL rating as the software, then safe computation seems to be expected and it only needs to be investigated if the ASIL level of the hardware is lower than required by the function. Several AUTOSAR Adaptive Platform mechanisms can be combined to achieve these goal, e.g. repeated or redundant execution in combination with some sort of self-test libraries and control-flow monitoring. The AUTOSAR Adaptive Platform might not directly support this feature with a specific interface or description, 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 the application.

6.1.2 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 different partitions, different controllers, ECU borders and even with the off-board world. And additionally, dynamic deployment requires authentication of communication partners and therefore:

- Provide an interface for an application or service to allow safe communication [RS_SAF_10014]

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

6.1.3 Safe Storage

It is also expected that applications and services require to load and store data persistently in a non-volatile memory unit, hence:

- Prevent unexpected alteration of data [RS_SAF_10037]
- Detect unexpected alteration of data [RS_SAF_10002]
- Prevent delay of data or storage access [RS_SAF_10008]

The AUTOSAR Adaptive Platform is hereby just providing an interface to the applications and services. The hardware 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.

6.1.4 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 during production. To prevent a 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
- Verify if an application is allowed to be deployed on the ECU
- Verify if an application is allowed to be deployed on the dedicated resource

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 checks 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. Therefore it is recommended that updates of safety critical applications are only performed in a safe state:

- Ensure that the safety relevant software is updated/upgraded in a state that cannot cause a hazardous situation [RS_SAF_10038]

If the application is just optional, the impact might not be big because the application might just not get scheduled. If the application is an update, then:

- Mitigate or prevent unintended or incorrect alteration of a valid configuration [RS_SAF_10002]
- Mitigate or prevent loss of a valid configuration [RS_SAF_10027]

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 to and can keep track of each configuration of each car in the field.

6.2 Safety Artifacts of the AUTOSAR Platform

Based on the Hazard Analysis, the Safety Needs and the Functional Safety Requirements, the following artefacts of the AUTOSAR Adaptive Platform have been identified:

6.2.1 Ensure correct computation, execution and execution order of multiple applications with mixed criticality

[RS_SAF_00001]

6.2.1.1 Classic Platform Autosar

- WdgM
- State Manager
- Basic Software Mode Manager
- OS

6.2.1.2 Adaptive Platform Autosar

- EM
- PHM
- SM

Information: The architectural elements EM, SM and PHM are highly safety relevant; safe execution and safe health management are fundamental to the safe operation of an Adaptive Application. The EM, PHM, SM elements are inter-dependent and coordinate their activities to ensure functional safety within the AUTOSAR Adaptive Platform.

6.2.2 AUTOSAR shall ensure correct configuration during the entire life cycle of the platform

[RS_SAF_00002]

6.2.2.1 Classic Platform Autosar

- EcuC
- OS
- BndM
- NvM

6.2.2.2 Adaptive Platform Autosar

- EM
- PHM
- UCM
- PER

6.2.3 AUTOSAR shall ensure correct update and upgrade of multiple platform and non-platform applications with mixed criticality

[RS_SAF_00003]

6.2.3.1 Classic Platform Autosar

- FOTA

6.2.3.2 Adaptive Platform Autosar

- UCM
- PHM
- CM[E2E]
- PER
- SM

6.2.4 AUTOSAR shall ensure correct exchange (transmission and reception) of information

[RS_SAF_00004]

6.2.4.1 Classic Platform Autosar

- E2E Library
- E2EXf
- ComXf
- SomeIpXf
- RTE

6.2.4.2 Adaptive Platform Autosar

- CM

6.2.5 AUTOSAR shall detect faults and failures while processing data, communicating with other systems or system elements

[RS_SAF_00005]

6.2.5.1 Classic Platform Autosar

- E2E Library
- E2EXf
- ComXf
- SomeIpXf
- RTE
- OS
- WdgM

6.2.5.2 Adaptive Platform Autosar

- CM[E2E]
- PER

6.2.6 AUTOSAR shall provide mechanisms to support safe storage for applications

[RS_SAF_00006]

6.2.6.1 Classic Platform Autosar

- NvM
- CRC
- OS

6.2.6.2 Adaptive Platform Autosar

- PER

6.2.7 AUTOSAR shall monitor, detect and provide means to react on detectable failures

[RS_SAF_00007]

6.2.7.1 Classic Platform Autosar

- WdgM
- OS

6.2.7.2 Adaptive Platform Autosar

- PHM
- UCM
- PER
- SM
- EM

A Abbreviations

Abbreviation	Description
2oo2	two out of two
2oo2D	two out of two with diagnostics
2oo3	two out of three
AD	Automated Driving
ADS	Automated Driving Systems
ADAS	Advanced Driver Assistance System
ASIL	Automotive Safety Integrity Level
BIOS	Basic Input Output System
CCA	Common Cause Failure Analysis
DFA	Dependent Failure Analysis
DMR	Dual Modular Redundancy
ECC	Error Correction Code
EMC	Electro Magnetic Compatibility
FSR	Functional Safety Requirement
HAD	Highly Automated Driving
HARA	Hazard And Risk Assessment
HSM	Hardware Security Module
NvM	Non-volatile Memory
PMIC	Power Management Integrated Circuit
QM	Quality Management
SG	Safety Goal
SoC	System on a Chip
SOP	Start of Production
TMR	Triple Modular Redundancy
TSC	Technical Safety Concept
TSR	Technical Safety Requirement
Wdg	Watchdog

Table A.1: List of Abbreviations

B Glossary

All technical terms used throughout this document - except the ones listed here - can be found in the official AUTOSAR Glossary [3] or ISO 26262 [1].

Term	Description
ASIL capability	Capability of an item or an element to meet assumed safety requirements assigned with a given ASIL
Checksum	A value used to verify the integrity of a data stored or transmitted
Context Switching Time	The time consumed by the CPU in switching from one process or thread to another
Cybersecurity	A set of techniques used to protect the integrity of networks, programs and data from attack, damage or unauthorized access. → *Security*
Data integrity	Data integrity is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.[21]
Memory Management Unit	Hardware element that handles virtual memory, memory translation and caching operations
Mixed criticality	A system or partition contains, schedules and executes software components like AUTOSAR Adaptive Applications according to different ASIL Levels at the same time

Table B.1: Glossary

List of Figures

1 Relationship of item, system, component, hardware part and software unit, Figure 3 - ISO 26262-10 [1] . . . 9
1.1 Considered chapters of ISO 26262, Overview of the ISO 26262 series of standards, Figure 1 - ISO 26262-1 [1] . . . 12
1.2 Structure of safety requirements and mapping to this Document, based on ISO 26262 [1] . . . 13
1.3 Hierarchy of safety goals and functional safety requirements . . . 13
2.1 AUTOSAR Classic Platform layered architecture [6] . . . 16
2.2 AUTOSAR Adaptive Platform layered architecture [6] . . . 16
3.1 Exemplary simplified vehicle system . . . 20
3.2 Systematic redundancy . . . 20
3.3 Decomposition with safety checker . . . 20
3.4 Exemplary draft of a simple ECU design . . . 21
3.5 Exemplary draft of a simple MCU design . . . 22
3.6 AUTOSAR Adaptive Platform Logical Architecture - Safety View [7] . . . 28

List of Tables

2.1 Top Level Safety Use-Cases or feature requests . . . 18
3.1 AUTOSAR Adaptive Platform Functional Cluster . . . 29
4.1 Top Level Failures . . . 34
5.1 Top Level Safety Requirements [20] . . . 35
5.2 Use-Cases, Failures and and derived safety requirements . . . 36
A.1 List of Abbreviations . . . 44
B.1 Glossary . . . 45