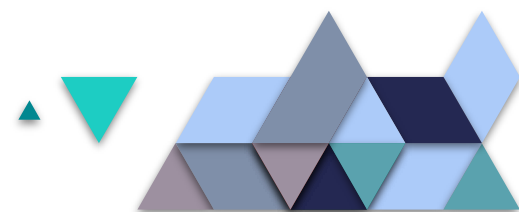




*Trustable architectures with acceptable residual risk for the electric,
connected and automated cars*

Deliverable	<i>Report on requirements and targets of reliability and safety at system level</i>		
Deliverable File	<i>D 1.4</i>		
Project	ArchitectECA2030	Grant Agreement Number	877539
Lead Beneficiary	IFAT	Dissemination Level	Public
Involved SCs	SC4	Related Task/s	Task 1.4
Due Date	m12 (30 June 2021)	Actual Submission Date	m12
Status	Final	Version	1.0
Contact Person	Dr. Horst Lewitschnig	Organisation	IFAT
Phone	+ 43 51777 3564	E-Mail	horst.lewitschnig@infineon.com



Document history			
V	Date	Author	Description
0.1	10. 03. 2021	Dr. Horst Lewitschnig	Initial Version
0.2	18. 03. 2021	Dr. Horst Lewitschnig	Update for the monitoring device
0.3	16. 04. 2021	Dr. Horst Lewitschnig	Partner inputs and first draft
0.4	10. 05. 2021	Dr. Horst Lewitschnig	Consolidation of partner input
0.5	17. 05. 2021	Dr. Horst Lewitschnig	Update with inputs from SC2, update of residual risks. Submitted for review.
0.6	26. 05. 2021	Petr Blaha	First round review
0.7	16. 06. 2021	Dr. Horst Lewitschnig	Update the input from the reviewers, structural update
0.8	24. 06. 2021	Dr. Horst Lewitschnig	Update with further inputs from SC4-partners
0.9	28. 06. 2021	Petr Blaha	Second round review
0.91	29. 06. 2021	Dr. Andreas Soppa	Second round review
0.92	30.06.2021	Dr. Cristina De Luca	Final review before upload
1.0	30. 06. 2021	Dr. Horst Lewitschnig	Final and reviewed version

Disclaimer

The opinion stated in this document reflects the opinion of the authors and not the opinion of the European Commission. The Agency is not responsible for any use that may be made of the information contained (Art. 29.5 of the GA).

1 Table of Contents

2	Executive/ Publishable Summary	6
3	Non-publishable Information	6
4	Introduction & Scope	6
4.1	Purpose and Target Group.....	6
4.2	Contributions of Partners	6
4.3	Relation to other Activities in the Project.....	7
4.4	Main Objectives and Key Targets Overview	7
5	System Level View	9
6	SC4 Overview, Structure, and Demonstrators	9
7	Requirements on Monitoring Devices.....	9
7.1	General Requirements	11
7.2	Design Phase	13
7.3	Validation and Verification	14
7.4	Monitoring in Operating Cars	15
8	Residual Risks	17
8.1	Current Risk Level	17
8.2	Risk Reduction with Monitoring Devices	18
8.3	Risk Calculations at Simulations.....	19
9	Hardening Automatism for Power or Motor Control Design – IFAG – Demonstrator 4.1	20
9.1	Target Goals and Achievements	20
9.2	Demonstrator Structure.....	21
9.3	Demonstrator Description	21
9.4	Residual Risks.....	22
9.5	Demonstrator Relations to the Main Objectives and Key Targets	22
9.6	Homologation Framework Mapping.....	22
9.7	Non-functional Requirements, KPIs, and Measures	23
9.8	Functional Requirements, KPIs, and Measures	24
9.9	Mapping to Existing Standards	25
9.10	Verification and Validation	26
9.11	Demonstrator Milestones.....	26
10	Lifetime Drift Model for Discrete Parameters – IFAT – Demonstrator 4.2	27
10.1	Target Goals and Achievements	27

10.2	Demonstrator Structure.....	27
10.3	Demonstrator Description	27
10.4	Residual Risks.....	28
10.5	Demonstrator Relations to the Main Objectives and Key Targets	28
10.6	Homologation Framework Mapping.....	29
10.7	Non-functional Requirements, KPIs, and Measures	29
10.8	Functional Requirements, KPIs, and Measures	30
10.9	Mapping to Existing Standards	32
10.10	Verification and Validation	32
10.11	Demonstrator Milestones.....	32
11	Virtual Verification & Validation Framework (V3 Framework) – AVL, TUG, SBA – Demonstrator 4.3	34
11.1	Target Goals and Achievements	34
11.2	Demonstrator Structure.....	35
11.3	Demonstrator Description	35
11.4	Residual Risks.....	35
11.5	Demonstrator Relations to the Main Objectives and Key Targets	35
11.6	Homologation Framework Mapping.....	37
11.7	Non-functional Requirements, KPIs, and Measures	37
11.8	Functional Requirements, KPIs, and Measures	39
11.9	Mapping to Existing Standards	40
11.10	Verification and Validation	42
11.11	Demonstrator Milestones.....	42
12	Automated Driving (AD) Demonstrator – VIF, DATA – Demonstrator 4.4	43
12.1	Target Goals and Achievements	43
12.2	Demonstrator Structure.....	44
12.3	Demonstrator Description	44
12.4	Residual Risks.....	46
12.5	Demonstrator Relations to the Main Objectives and Key Targets	47
12.6	Homologation Framework Mapping.....	48
12.7	Non-functional Requirements, KPIs, and Measures	48
12.8	Functional Requirements, KPIs, and Measures	49
12.9	Mapping to Existing Standards	51
12.10	Verification and Validation	52

12.11	Demonstrator Milestones.....	53
13	SC4 Demonstrators Summarized.....	53
14	Conclusion	54
14.1	Contribution to Overall Picture.....	54
14.2	Relation to the State-of-the-Art and Progress beyond It	54
14.3	Impact to other WPs, Tasks, and SCs	54
14.4	Contribution to Demonstration	55
14.5	Other Conclusions and Lessons Learned	55
15	References.....	56
16	List of Figures.....	57
17	List of Tables.....	57
18	Internal Review.....	58

2 Executive/ Publishable Summary

The concept of a monitoring device is to establish functionalities that can monitor performance, health status, and **remaining useful life (RUL)** of a system. Monitoring devices can be realized in various forms and at diverse system levels.

Furthermore, monitoring devices contribute to the reduction of residual risks of automated driving cars.

In this report, general requirements for monitoring devices are described. Based on these, specific requirements for the demonstrators in Supply Chain (SC) 4 are derived.

3 Non-publishable Information

Not applicable.

4 Introduction & Scope

4.1 Purpose and Target Group

The purpose of this work is to define requirements and targets for monitoring devices.

We see monitoring devices in a wider scope: monitoring devices can be hardware, software, and methods. Monitoring methods are then implemented in the hardware and software components. Various monitoring devices in a system also contribute to reduce the residual risks of automated driving vehicles.

The target group of the presented results are the SC4 partners, partners from SC1 – 3 and SC5, as well as Tier1s and OEMs.

4.2 Contributions of Partners

Partners of the task 1.4 as well as TU Graz and VIF contributed to this report.

TABLE 4-1: PARTNER CONTRIBUTION OVERVIEW

Chapter	Partner	Contribution
9	IFAG	Demonstrator: Hardening automatism for power or motor control design
10	IFAT	Lead of task 1.4, demonstrator: lifetime drift model for discrete parameters
11	AVL	Demonstrator: virtual validation and verification
11	SBA	Demonstrator: virtual validation and verification
11	TU Graz	Demonstrator: virtual validation and verification
12	DATA	Automated driving demonstrator
12	VIF	Automated driving demonstrator

4.3 Relation to other Activities in the Project

The activities of WP1 in general and task 1.4 specifically refer to the starting point of the development, i.e., the requirements and specifications, followed by the concept phase. The work is aligned within WP1 and SC4.

There are links to SC1 – SC3 and tasks 1.1 – 1.3, where monitoring functionalities in various forms are specified. E.g., in task 1.2 the partner Brno University of Technology (BUT) works on condition monitoring and predictive maintenance of inverter power components. An exchange of information has been established.

This report is the input to task 2.3.

4.4 Main Objectives and Key Targets Overview

O1 - Continuous robust design optimization for each part in the ECS value chain (Technical): Related to ensuring secure connected, cooperative, and automated mobility and transportation; and managing critical, autonomous, cooperating and evolvable systems. The outcome is a robust mission validated design. The objective's KPI addresses coverage components.

O2 - Framework for safety validation of ECS value chain (Technical): Related to managing critical, autonomous, cooperating, evolvable systems; managing complexity; and increasing compactness and capabilities by functional and physical systems integration. The outcome is accepted residual risk in ACS for HAD to enable type approval. The objective's KPI addresses coverage validation.

O3 - Identification and management of residual risks over the entire ECS value chain (Technical): Related to ensuring secure connected, cooperative, and automated mobility and transportation; and reliability and functional safety. The outcome is accepted monitoring device methodologies. The objective's KPI addresses coverage qualification.

O4 - End-user acceptance by trustworthy ECS value chain (Value): Related to ensuring secure connected, cooperative, and automated mobility and transportation; and secure, safe, and trustable connectivity and infrastructure. The outcome is usability. The objective's KPI addresses coverage test.

O5 - Zero emissions, crashes, and congestions by ECA2030 vehicle (Value): Related to ensuring secure connected, cooperative, and automated mobility and transportation; and increasing compactness and capabilities by functional and physical systems integration. The outcome is sustainability. The objective's KPI addresses energy efficiency and shorter validation time.

KT1 - Architectures, components, sub-systems enabling virtual development and validation (monitoring device, failure risk): Currently no agreed development framework exists to certify SAE L3+ automated driving functions in unstructured environments and adverse weather conditions using virtual validation methods. The reduction of the validation process compared to a mileage accumulation driven approach will be shown. Additionally, the project will illustrate that the proposed framework is appropriate for certification of SAE L3+ automated driving functions on case studies within the project.

KT2 - Methods and tools to validate the models used in virtual validation (lifetime monitoring, residual risk, methods, and tools): The incorporation of real-world test data into the virtual V&V process in combination with test data derived from a knowledge base is most likely the key to develop,

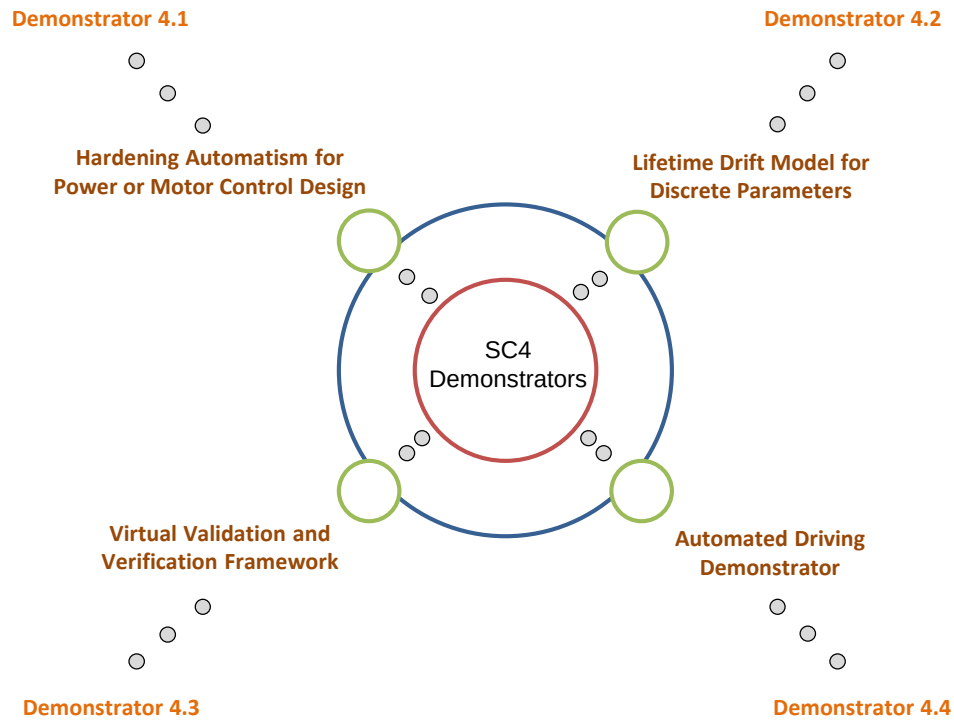
certify, and re-certify automated vehicles with reasonable costs and efforts. As an assessment, the project will provide a knowledge base that integrates pertinent data for safety validating and testing CAVs ECA vehicles and illustrate the alimentation of a corresponding data base with pilot test data generated by the monitoring device.

KT3 - Metrics for quality assurance for Electronic Components and Systems (ECS) (mission-oriented qualification, residual risk): Standardized safety, security and privacy metrics are of high relevance for type approval, vehicle operation residual risks (e.g., insurance) and homologation procedures. As an assessment, the project will illustrate the introduced coverage metrics by application of the homologation framework on case studies within the project. The project will show on case studies in a lab demonstrator within the project, proposed techniques keeping the residual risk less than $10E-9/h$.

KT4 - Definition and understanding of test coverage (residual risk, design feedback, lifetime monitoring, aggregated risk): The system understanding is the precondition to elaborate the homologation of automated vehicles. The functional safe design, system availability, and safe operation are mandatory to bring the vehicles into real traffic operation. As an assessment, the project will illustrate that the proposed processes and requirements enable to meet the expected increase of availability on case studies within the project. Reduction of homologation effort by virtual validation. Reduction of fatalities which might happen during homologation only occur in virtual environment and therefore issues are not perceived as severe as in real world driving. This increases customer acceptance.

KT5 - Methods for shorter validation in respect to acceptable residual risk (methods): The project's goal is to provide methods for shorter validation in respect to an acceptable residual risk. The project will bring together representative stakeholders from public authorities and reduce the heterogeneity of regulations to allow autonomous mode by introducing the developed homologation framework. As an assessment, the project will provide the list of propositions emanating from the project that found their way into international standards.

5 System Level View



6 SC4 Overview, Structure, and Demonstrators

	SC4 Demonstrator Number	Title
Method	4.1	Hardening automatism for power or motor control design
Model	4.2	Lifetime Drift Model for Discrete Parameters
Simulation	4.3	Virtual Validation & Verification (V3) Framework
Vehicle	4.4	Automated Driving Demonstrator

7 Requirements on Monitoring Devices

In this chapter, we describe overall requirements that are related with monitoring devices. In Chapters 9 – 12 we apply these overall requirements to the demonstrators of SC4.

The idea of the monitoring device is to give an early indication about an upcoming failure of an assigned component.

Such monitoring devices exist already in various forms in a wide range of applications, e.g., in mining, construction works, gas flow leakage control, and the like. In this project, we bring this idea to electronic applications. The innovation in this project is to develop monitoring devices for their usage in automated driving cars.

The concept of the monitoring device is a contribution to assess the health status of components and systems. It is part of a prognostics and health management system (PHM), see, e.g., (Economic Analysis of Canary-Based Prognostics and Health Management, 2011), (A Patent Analysis of Prognostics and Health Management (PHM) Innovations for Electrical Systems, 2018), and (A Framework of Discrete-Event Simulation Modeling for Prognostics and Health Management (PHM) in Airline Industry, 2017).

At automated driving cars, we see the need for various monitoring devices in different domains.

The challenge for PHM in electronic systems is to find the right signals and information that indicate the health status. An optimum has to be found between costs, reliability (additional devices bear the risk to fail as well), and integration. Therefore, we are looking for smart solutions that minimize the number of additional devices and maximize the information we can gain.

In this project, we identified in SC4, four demonstrators for monitoring devices. We see these as concrete starting points for their realization in automated driving cars.

The idea of the monitoring device is essentially a matter to have a reference at which the degradation or loss of functionality of a component or a subsystem can be observed during operation. This reference can be realized in various ways, e.g.:

- **A weaker structure is used parallel to a component.** If the weaker structure degrades or fails, this is an indication about the **health status of the component**. An example from semiconductors is:
 - **Risk:** intrinsic dielectric breakdown.
 - **Measure:** Design a capacitor where the dielectric material is exposed to a slightly higher electrical field than the dielectric material in the rest of the device. A failure of this capacitor indicates the upcoming end of life of the device.
- **Comparison to a target state.** An example from a **perception system** is:
 - A virtual picture could be submitted as an input to the perception system and then it can be **checked if the perception system still operates** as intended.
 - Any **deviation in the perception over time** can be tracked and extrapolated to calculate the **remaining useful life (RUL)**.
- **Counter:** An example is an error counter:
 - The number of errors when transferring data can give an indication about the **degradation** of the car's internal communication system.

With a monitoring device, we can trace the status of components, modules, subsystems, and systems directly at their operations. This gives an excellent opportunity to react on their change of behaviour and properties directly.

The kind of monitoring device depends on the specific component or subsystem and needs to be embedded in the overall monitoring concept.

Generally, monitoring comprises several functionalities, like

- fault detection,

- diagnosis capabilities,
- report of health status.

Any action based on monitoring data is done beyond the monitoring functionality by a “decision system”.

In the context of automated driving cars, there is an increasing demand to recognize failures before they occur. Such failures are typically degrading failures which “announce” themselves. This can be, e.g., an increased leakage current, unusual vibration signals, higher error counts, and the like. In this context, also a loss of minor functionality can announce a more severe damage.

In this project, we are focussing on the monitoring functionalities that identify degradation effects.

If degradation can be detected and a decision system acts on it, it turns a potential failure into a maintenance action. In this way, monitoring is a measure to reduce the residual risk, failure rates, and built-in redundancy.

Monitoring should be supported at all levels of an automated driving car, i.e.,

- component level,
- module level,
- subsystem level, and
- system level.

Monitoring devices have to be incorporated into the design of the corresponding systems. A defined standardized communication between the various system levels is essential.

Generally, there is an increasing trend to take advantage of monitoring functionalities in order to improve a product’s performance. This is also reflected in several ISO standards, like (ISO, 2013), (ISO, 2015), and (ISO, 2018).

For automotive applications, substantial work about monitoring functionalities is done in the SAE standard JA6268™ (SAE, 2018). We refer to this work in this section. Additionally, we are introducing concepts and needs for monitoring devices for automated driving cars.

The remainder of this section is structured as follows: In the remainder of Section 7, we state general requirements on a monitoring device. In Section 8, we address the topic of residual risks. In Sections 9 - 12, we describe the demonstrators and derive specific requirements for the monitoring devices.

7.1 General Requirements

The monitoring capabilities of a component depend on the complexity and functional integration of the device. Monitoring data are an input to a higher-level system, e.g., a microcontroller or a decision system.

For highly integrated devices, a monitoring device can be implemented directly in the device, therefore a deviation at the device itself can be detected and reported.

We see general requirements for such monitoring devices and functionalities as:

- Degradation of relevant features and parameters should be detected.

- The monitoring device should be able to evaluate the health status of a component at any point in time.
- The monitoring data and methods should support to retrieve an estimation of the remaining useful life (RUL).

7.1.1 Monitoring Data

The kind of data for monitoring can range from raw data to complex and aggregated information. E.g., (see also (SAE, 2018))

raw data	health status
processed data	degradation analysis
build-in self tests	fault analysis
self-diagnosis	prediction of remaining useful life

7.1.1 Defined Interfaces

For any kind of monitoring data exchange, there should be a defined interface on how monitoring information is transferred. It is highly recommended to use existing default interfaces for this purpose wherever it is applicable.

It is important to note that an interface definition goes beyond the pure physical connection and includes protocols and software stacks as well.

7.1.2 Intellectual Property

Each supplier can have his own solution how monitoring devices and functionalities are designed into his product. The requirements for defined interfaces and information about degradation should not impact the supplier's available room for design and the protection of his intellectual property (IP). Innovative solutions to provide monitoring data should still be encouraged in order to spur competition. Nevertheless, this also underlines the need for defined interfaces for monitoring data.

7.1.1 Calculation and Storage Resources

For any kind of monitoring, additional calculation and memory resources are required. Depending on the application and the criticality of the monitoring information, this calculation resources can be located

- in an IC,
- in a module/subsystem,
- on system level, or
- in the cloud.

Monitoring functionalities should be designed in a way that they consume a minimum of memory and calculation resources. Nevertheless, this kind of calculation and storage resources have to be considered in the system architecture.

Furthermore, multiple use of monitor components, e.g., for error correction and error monitoring shall be encouraged to contribute to overhead reduction as well.

7.2 Design Phase

In general, monitoring functionalities have to be designed into a system. Monitoring functionalities are typically very specific for individual components and subsystems.

7.2.1 Reliability Targets

In order to design a monitoring device, we start with reliability targets. Such targets are, e.g.,

- lifetime, e.g., 15 years or 200 000 km;
- FIT-rate, percentage of failures;
- uptime, downtime, unplanned downtime;
- warranty costs;
- functional safety targets (e.g., ASIL D).

Reliability targets can differ for safety critical functions, operating functions, and comfort functions. Especially for safety critical functions, we see the need for higher reliability targets compared to today's state-of-the-art.

Monitoring functionalities help to reduce the number of degradation failures. This can also be formulated as a concrete target, e.g., a monitoring device should contribute to reduce the residual risk for a certain function by 90 %.

7.2.2 Failure Model

A failure model for the system is needed (SAE, 2018). In particular, we need failure models for system components together with their (faulty) behaviour.

In order to identify failure models, established tools should be used, like FMEA, FMEDA, FTA, and the like. From the failure models, failure mechanisms are identified. Monitoring devices can then be designed to cover specific failure mechanisms.

Nevertheless, one needs to conceive that a monitoring device can fail as well. This should be considered in the design of the monitoring device and the architectural setup.

- In general, single monitoring devices should be kept small and robust.
- For an architectural setup, e.g., there could be two devices that check each other. Such an approach could also reduce the risk of a failing monitoring device.

7.2.3 Processing Model

The processing model (SAE, 2018) is a functional description of the operating device. It includes the input from the failure model and its monitoring functionality as well.

The Open Software Abstraction Model (OSAM) from the partner DATA should be implemented into the processing model. This supports flexibility in used components and software and contributes to achieve functional safety and reliability targets.

7.2.4 Design Strategy

New design automation methods shall be encouraged to reduce the design implementation and integration effort of monitoring devices.

An example for this is demonstrator 4.1 from the partner IFAG (see Chapter 9).

This document and the information contained may not be copied, used or disclosed, entirely or partially, outside of the ArchitectECA2030 consortium without prior permission of the partners in written form.

7.2.5 Monitoring Functionality

Functional monitoring requirements need to be specified. That covers, e.g., (SAE, 2018)

fault detection capabilities	data processing
fault detection accuracy	reporting to higher-level systems
fault isolation accuracy	machine-readable communication standards
fault prediction accuracy	compliance with ISO26262
data recording	support of security functionalities

7.2.6 Monitoring Functionality within the System Architecture

For the data provided by the monitoring functionality, we propose a structure like the following:

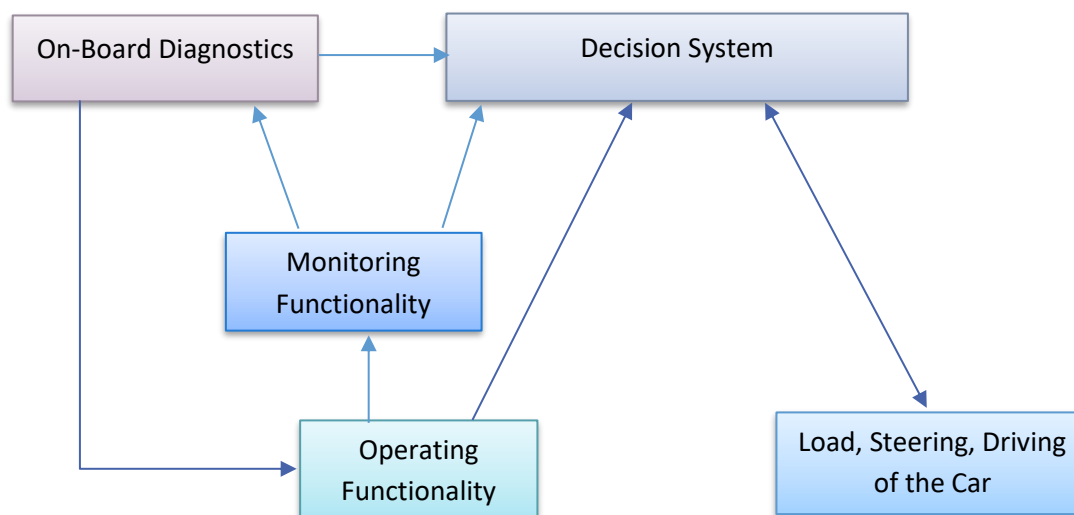


FIGURE 1: MONITORING FUNCTIONS IN THE SYSTEM ARCHITECTURE

This is a general proposal. The basic idea is that monitoring functionality should not interfere with the operating functionalities, but it should be an add-on. Monitoring functionalities should report to higher level systems and to the on-board diagnostics system. Such a concept should be embedded in the system architecture of a car.

We recommend that monitoring functionalities should be placed as close as possible to the location of the observed data. E.g., the monitoring information of a sensor should be provided by the sensor itself, or a microcontroller should inform itself about its deterioration. We also understand that, depending on the concrete technical situation, there might be good technical reasons to put monitoring functionalities to other appropriate locations.

7.3 Validation and Verification

For the validation and verification of monitoring devices, activities in standardization bodies are ongoing. According to our analysis in the project, such standards are not yet finalized (e.g., SAE: Verification & Validation of Integrated Vehicle Health Management Systems and Software (SAE, 2014), which is still ongoing).

Validation and verification require test cases. In our case, we need additional test cases for the use of the monitoring devices.

We see as a **minimum set of requirements for validation and verification** the following aspects:

Functional suitability	The test cases should be complete and should provide sound data.
Performance efficiency	The number of test cases should be as small as possible in order to test all in a feasible time.
Compatibility	The test cases should be defined using well known or open standards.
	The input model for discrete modelling should be a discrete model, i.e., a model having discrete parameters and discrete parameter values.
Usability	The test report should give easy to grasp feedback about the test status.
	Residual risk assessment should be visible in the test report.
Security	Access to the test data should only be under control of the testing party.
Safety	No harm to the system under test or the environment should be done.
Maintainability	It should easily be possible to translate the tests to other test formats.
Portability	The method to generate test cases should not be specific to a certain vehicle-type, decision system, monitoring functionality, or environment.
Reliability	The generated test cases should be trustworthy.
	The generated data should be traceable.
	The generated test cases and the result of the execution of these test cases should be validated using a suitable metric. Such suitable reliability metrics could be, e.g., the proportion of correct behaviour, the proportion of behaviour not according to specification, or the proportion of false alarms.
Implementation	A special task of verification is to make sure that the insertion of monitoring devices does not change the original behaviour. Also here, the use of automated methods shall be encouraged.

7.4 Monitoring in Operating Cars

Monitoring devices are embedded in the various functionalities of a car. During operation, several aspects need to be considered.

7.4.1 Monitoring Method

The monitoring method determines how monitoring data are transmitted (SAE, 2018).

Push or pull of monitoring data	Is the monitoring information pushed by a device or subsystem, or is it requested by a higher-level decision-system?
--	---

Monitoring frequency	The rate of the transmission of monitoring data should be defined. It depends on the criticality of the supported function. This can be, e.g., - online monitoring, - monitoring data are provided at defined time intervals or events, e.g., start up or shut down of a car; - monitoring data provided on request, e.g., by a decision system, by a motor vehicle workshop, or by the car user. Monitoring data can also be requested remotely.
Monitoring at on- or off-state	It should be specified what monitoring data have to be available during the car's operation and what data should be processed in the car's off-state.

7.4.2 Monitoring Data Logging and Reporting

Reported data should be logged. It should be specified what data are reported in addition to the monitoring data (SAE, 2018):

- **data type,**
- **time stamp,**
- **component ID,**
- **component status.**

We see also the need for logging further data, like

- **operating conditions of the monitored device,**
- **usage data of the car (usage profile),**
- **monitoring history.**

7.4.3 Monitoring Runtime Data Exchange

It should be specified what kind of monitoring and health status data are exchanged during the operation of the car (SAE, 2018).

Transfer of monitoring data within the car	This should be part of the operational model where monitoring functionalities are embedded.
Transfer of monitoring data to external systems	E.g., to motor vehicle workshops, to the car user via an app, or to an external decision system which can also be in the cloud.
Definition of what kind of information is exchanged during operation.	Data protection issues have to be considered. For the case of an accident, it should be defined what kind of additional monitoring data are submitted via the automatic emergency call system (eCall system). The eCall system makes an automated emergency call in case of an accident. It was introduced by the European regulation 2015/758.

8 Residual Risks

Driving a car is related with risks. Assisted and automated driving cars have the potential to substantially reduce this risk level. We see that monitoring devices support this effort.

The term “risk” is defined in, e.g., ISO 26262-1:2018 (ISO, 2018-1) Chapter 3.128:

- **risk**
combination of the probability of occurrence of harm and the severity of that harm.

Similar definitions can be found in ISO/IEC 25010:2011 (ISO, 2011) Chapter 4.3.12, and UL 4600 (UL, 2020) Chapter 4.2.35.

The term “residual risks” is defined in ISO26262-1:2018 (ISO, 2018-1) Chapter 3.126:

- **residual risk**
risk remaining after the deployment of safety measures.

The term “residual risk” is further used in the ISO-Standard “Road vehicles – Safety of the intended functionality” (ISO, 2019) Chapter 4 which states

“Furthermore, this document also seeks to assess the **remaining residual risk** from:

- i) a system not able to process a given scenario in a safe manner, and
- ii) the involved persons (driver, other vehicle occupants, or bystanders) are not capable of mitigating the hazardous event, is acceptable.”

Moreover, Chapter 11.2 and Table 9 in the ISO-Standard “Road vehicles – Safety of the intended functionality” (ISO, 2019) gives **methods how to evaluate residual risks**.

ISO 26262-3:2018 (ISO, 2018-3) Chapter 6 describes **hazard analysis and risk assessment in the concept phase**. Table 4 in (ISO, 2018-3) determines ASIL-levels for various combinations of severity, exposure, and controllability classes.

8.1 Current Risk Level

A further point of view is to take a look at the EU road accident statistics (European Road Safety Observatory, 2021):

- In 2019, the road fatality level within the European Union (EU) was 51 ppm.
- The range over the member states was from 22 ppm to 96 ppm.

Let us further take a look at the number of injuries related to road traffic in 2019:

- In the European Union, 1.2 Mio. people were reported to get injured in road traffic in the year 2019 (European Road Safety Observatory, 2021).
- For the population, let us take data from 1st Jan. 2020 as an approximation of the population in the EU in 2019. The EU population was estimated 447.7 million (Eurostat, 2020).
- This results in ~ 0.27 % probability to get involved in a crash resulting in injuries or death.

Let us take a closer look about those accidents where people were injured or killed. These data are from Austria (Statistik Austria, 2020) from 2019. They serve as an example, because European data are not readily accessible:

Assumed root cause	Percentage
Misbehaviour of involved people - Speed not adopted to the traffic - Distraction - Failure to yield to traffic	90 %
Vehicles - Wrong securing of freight - Technical defects	1.1 %
Impairments - Alcohol, drugs, medication - Tiredness - Health impairment	8.9 %

These numbers reflect today's risks at the current traffic level and can serve as a starting point for further discussions.

One target of automated driving is to reduce the fatality level and the risk of injuries in the road traffic. Monitoring devices contribute to reduce this residual risk. The current risk levels should act as a reference to assess the effectiveness of monitoring devices and automated driving cars.

8.2 Risk Reduction with Monitoring Devices

At monitoring devices, we can see the degradation of a device/ module/ subsystem/ system directly. We can project the degradation path; based on this, we can calculate the remaining useful life (RUL).

First simulations have shown that the uncertainty related with this projection is higher than the uncertainty related with classical end-of-life reliability testing. On the other hand, following degradation paths of devices is an established method for highly reliable devices where failures in accelerated lifetime tests are hardly seen; see, e.g., (Accelerated Degradation Tests: Modeling and Analysis, 1998), (Degradation Modeling Using Stochastic Processes With Random Initial Degradation, 2019), and (A General Stochastic Degradation Modeling Approach for Prognostics of Degrading Systems With Surviving and Uncertain Measurements, 2019). Furthermore, information about mission profiles and usage conditions are directly reflected in the used devices.

Monitoring devices cover certain failure mechanisms of the assigned device/ module/ subsystem/ system. The risk reduction that can be achieved by monitoring devices has to be assessed by each domain separately. This assessment will be further developed in the course of this project.

Generally, for automated driving vehicles we see the need to increase the reliability of the involved components. For this purpose, we propose to pursue the idea of an additional ASIL-level in ISO 26262, like ASIL-level "E", to account for failure rates of 10^{-9} /h and smaller.

Furthermore, we propose to establish the framework of a reliability growth model, see, e.g., (Reliability Growth Modeling, 2014) and (Reliability Growth Testing, 2014). The purpose is that with improvements on each domain level the overall performance of automated driving cars will increase as well. Reliability growth models are suitable approaches to reflect progress of the overall system performance.

8.3 Risk Calculations at Simulations

During the development of monitoring devices, simulations are essential tools. Traffic simulations can substitute real situations. Digital twins and relevant test cases need to be developed. Based on these, the behaviour of the automated vehicle is simulated. In this way, simulations are crucial in the assessment of the residual risks.

For the simulation, the setup of a digital twin of an automated driving car is needed. A practical compromise has to be found at what granularity level the digital twin is established. Different domains can be pictured with individual levels of details. Nevertheless, the monitoring devices should be reflected accordingly within the digital twins.

On the other hand, the development of test cases is needed for simulations. We see this as a dynamic process. Test cases are updated and new test cases are developed continuously, based on real road experiences and scenarios from automated driving cars.

The residual risk that remains after testing can be estimated with different measures of test coverage. Besides structural coverage, we also apply measures of input space coverage such as combinatorial coverage. This is, we have an estimate of how many combinations of the input space parameters are covered by the test cases. Existing combinatorial coverage measurement tools support this task.

Furthermore, at simulations monitoring devices can be virtually switched on and off. In this way, we can get to a direct measure of how efficient monitoring devices are and can quantify their contribution in reducing the overall residual risk of a system.

9 Hardening Automatism for Power or Motor Control Design – IFAG – Demonstrator 4.1

The semiconductor industry employs accurate models, methodologies, and tools in the course of the virtual development process in order to anticipate, analyse, and address physical effects in later real-life applications.

For the purpose of assessing the impact on functional safety, technology-specific fault mechanisms and failure rates are considered. Those failure mechanisms and failure rates are assumed to be sufficiently stable during the expected product lifetime as typically depicted by a so-called ‘bathtub-curve’. Figure 2 gives an example with reference to ISO 26262-11:2018 (ISO, 2018) Chapter 4.6.1.5.

With the exponential increase of the failure rate on the right-hand side a product starts to operate out of its specification. This increase is attributed to, e.g., continuous thermal or physical stress causing a permanent change of semiconductor characteristics as the performance drift of transistor or the migration of material.

The beginning of this ‘wear-out’ phase depends also on the mission profiles in applications which are not always completely specified during the development phase.

This demonstrator implements an approach to observe fault rates during operation.

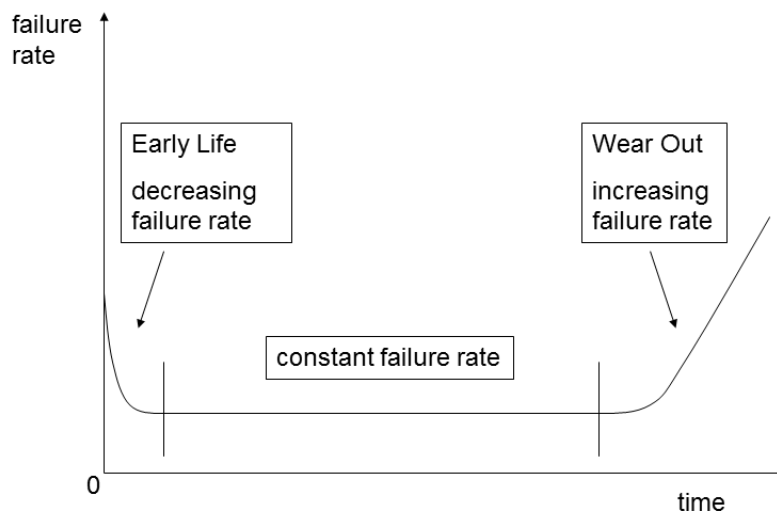


FIGURE 2: BATHTUB CURVE - EVOLUTION OF FAILURE RATE OVER TIME

9.1 Target Goals and Achievements

The goal of the demonstrator is to show the industry strength on the implemented methods and to act as an edge device in ArchitectECA2030 addressed domains.

The Infineon Architect2030 demonstrator will be implemented in a power control (e.g., voltage regulator, battery charging) or a motor control design. The demonstrator shall prove that the developed methods and tools for optimized automated insertion of error detection/correction hardware (bit-flip checker/corrector (BFC) as well as a safety controller which accumulates the errors) work properly and how they contribute to standards like ISO26262. The monitoring of errors is essential to several ASIL levels. A trend analysis of the error monitoring may also give hints on degradation and stress.

9.2 Demonstrator Structure

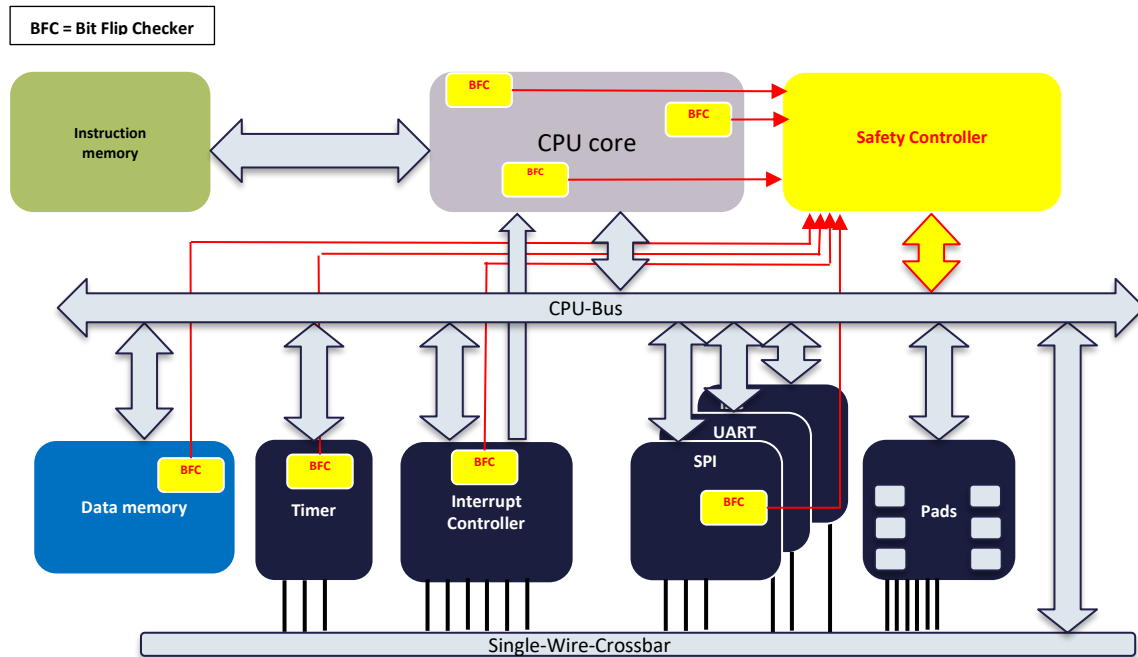


FIGURE 3: BFC EMBEDDED IN AN IC

9.3 Demonstrator Description

The quantitative analysis of the residual risk in the course of the semiconductor development is based on design considerations and technology's fault mechanisms which in turn depend on the real application profiles. This demonstrator monitors for one exemplary fault class detected resp. corrected events (transient bit flips of register storage elements). These fault rates can be reported to a higher application level for a further assessment whether the electronic components are operated in an expected context or the assumptions for the quantified risk are valid.

The power control demonstrator is based on classical System on Chip (SoC) with CPU (RISC-V in our case), memory, interrupt controller, timer, and communication peripherals (see Figure 3). The demonstrator is accessible via an external Serial Peripheral Interface (SPI). Depending on the implementation, like SPI-Interface with or without Direct Memory Access (DMA), i.e., with or without access to the status registers of the safety controller, data transmission, e.g., general data, programs snippets, and reads status information as for instance the number of observed faults, is done directly or via firmware running on the RISC-V.

The demonstrator will be available as FPGA hardware prototype.

The design will be hardened with the automation developed in ArchitectECA2030. This hardening includes both error correction and error detection hardware (BFC), as well as a „safety controller“, which accumulates the detected errors. The safety controller is accessible from the on-chip CPU and

via an SPI-interface. Generated properties will verify the correctness of the hardening, while simulation will validate fault propagation methods. Both will use ArchitectECA2030 developed tools and methods.

The further assessment of the counted errors is done outside of this demonstrator.

For validation of the entire system, the design loaded to the FPGA will be modified so that it mimics one or more faults. The fault handling from occurrence, via detection and accumulation in a safety controller, to the final handling by a supervisor device connected to the SPI, shall be validated.

9.4 Residual Risks

The demonstrator adopts an understanding of residual risk as defined by the ISO 26262-1:2018 (ISO, 2018-1), Chapter 3.126,

residual risk

risk remaining after the deployment of safety measures.

The understanding and quantification methods of a residual risk is in the scope of the ruling ISO 26262 editions; an adjustment in the scope of this demonstrator is not required.

Today the quantified residual risk is derived during the development cycle based on partially assumed mission profiles, base failure rates, and product lifetime.

The observed fault events of this demonstrator can be employed to validate the development assumptions in an application context.

9.5 Demonstrator Relations to the Main Objectives and Key Targets

9.5.1 Objectives

O1 - Continuous robust design optimization for each part in the ECS value chain (Technical):

The automatically inserted safety mechanisms harden edge devices in a robust and scalable way.

O2 - Framework for safety validation of ECS value chain (Technical):

The analysis methods developed in parallel enable safety validation of the edge chip.

9.5.2 Key Targets

KT1 - Architectures, components, sub-systems enabling virtual development and validation (monitoring device, failure risk):

An FPGA prototype will be built to enable early validation.

KT2 - Methods and tools to validate the models used in virtual validation (lifetime monitoring, residual risk, methods, and tools):

Generated properties verify the correct insertion of the safety mechanisms and their correct behavior.

KT5 - Methods for shorter validation in respect to acceptable residual risk (methods):

HW assisted as well as massively parallel simulation will shorten the validation time.

9.6 Homologation Framework Mapping

The BFC is used at a semiconductor device (subcomponent element in Figure 6).

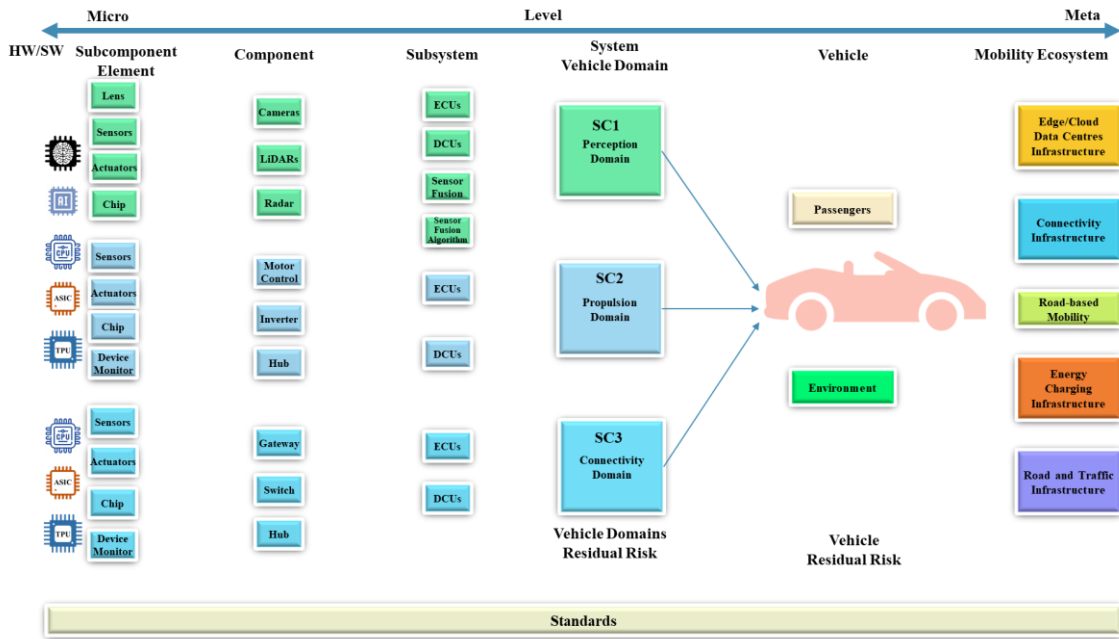


FIGURE 4: SYSTEM STRUCTURE OF AUTOMATED VEHICLES

Accordingly, when the BFC is used in an IC, it is part of the component approval as well as the functional safety assessment according to ISO 26262.

9.7 Non-functional Requirements, KPIs, and Measures

In Table 9-1 and Table 9-2 we present the three most relevant NFRs, KPIs, and measures for the demonstrator 4.1.

TABLE 9-1: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.1 – RELIABILITY

NFR	Reliability
NFR definition	The safety controller shall have an own fit rate.
KPI name	Availability of safety controller.
Description	The correct functionality and availability of the safety controller is required to enable a decision unit on an application layer on top of demonstrator to initiate safety related actions. Hence the safety controller has to be considered a safety function.
Measure	Safety Concept, Safety Analysis
Type of measure	Concept/Design
Method of collection and measurement	n.a.
Demonstrator target	The monitoring of fault events while operation shall be demonstrated.
KPI for Verification and validation	The HDL simulation and FPGA have to demonstrate the correct handling of faults injected into the safety controller.

TABLE 9-2: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.1 - RELIABILITY

NFR	Reliability
NFR definition	The BFC – to – safety interface controller shall have an upper limit for false positives and false negatives.
KPI name	Trade-off between robustness (fault scenarios) and design constraints (area-performance).
Description	The safety controller monitoring fault events have to account for the possibility of independent, parallel fault events, by ensuring a robust operation while still providing meaningful counter results.
Measure	Safety Concept, Safety Analysis.
Type of measure	Concept/Design
Method of collection and measurement	n.a.
Demonstrator target	Demonstration of handling of parallel, independent faults.
KPI for Verification and validation	The HDL simulation and FPGA have to demonstrate the correct handling of faults injected into the safety controller.

9.8 Functional Requirements, KPIs, and Measures

In Table 9-3, Table 9-4, and Table 9-5 we present the three most relevant FRs, KPIs, and measures for the demonstrator 4.1.

TABLE 9-3: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.1 - PERFORMANCE

FR	Performance
FR definition	The Safety Controller shall have own scalable BICs.
KPI name	Availability of safety controller.
Description	Failures may also show-up in the safety controller causing the controller to report faults that do not impact the design.
Measure	Independent implementation of BICs in Safety Controller.
Type of measure	Concept/design
Method of collection and measurement	n.a.: Part of Collection and Measurement.
Demonstrator target	Error Handling in Safety Controller.
KPI for Verification and validation	HDL Simulation and FPGA Prototype will be used to validate concept. Design will be formally verified.

TABLE 9-4: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.1 – FAILURE HANDLING

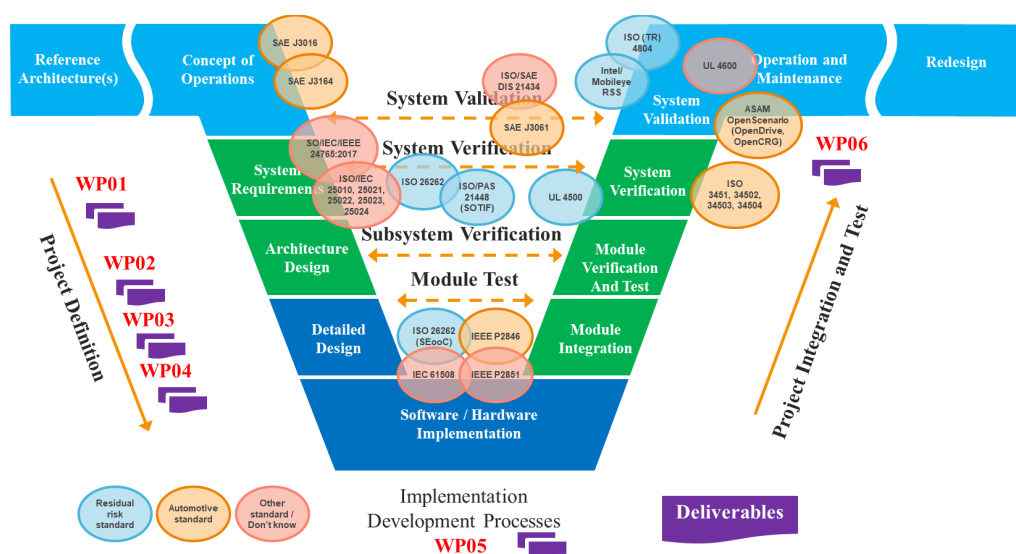
FR	Failure Handling
FR definition	The Safety Controller shall handle its own failures.
KPI name	Availability of safety controller.

Description	Faults detected in Safety Controller shall be distinguishable from faults in the functional design.
Measure	Separate reporting of faults detected in Safety Controller. More powerful error correction in Safety Controller.
Type of measure	Concept/design
Method of collection and measurement	n.a.: Part of Collection and Measurement.
Demonstrator target	Independent analysis of faults in Safety Controller
KPI for Verification and validation	HDL Simulation and FPGA Prototype will be used to validate concept. Design will be formally verified.

TABLE 9-5: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.1 - SCALABILITY

FR	Scalability
FR definition	The Safety Controller shall have a configurable number of so-called safety channels.
KPI name	Availability of safety controller.
Description	This is a way to allow a safety controller to serve different sized applications.
Measure	Generic implementation of Safety Controller.
Type of measure	Concept/design.
Method of collection and measurement	N.A.: Part of Collection and Measurement.
Demonstrator target	Code review.
KPI for Verification and validation	HDL simulation and FPGA prototype will be used to validate the concept. Design will be formally verified.

9.9 Mapping to Existing Standards



Standard code	Standard title	Why relevant
AEC Q100	Failure Mechanism Based Stress Test Qualification for Integrated Circuits	This standard is most relevant for the qualification of integrated semiconductor devices.
ISO 26262	Road vehicles - Functional safety	Address safety-related systems that include one or more E/E systems and that are installed in series production passenger cars (with a maximum gross vehicle mass up to 3 500 kg). The standard defines FIT-rate targets for several ASIL-levels. The installation of a monitoring device can turn a failure into a maintenance action. Furthermore, it can be seen as a safety mechanism. This has a direct impact to the FIT-rates of safety critical functions.

9.10 Verification and Validation

The safety controller is part of the mechanisms that are required to ensure ISO26262 required error detection levels. There is only a small extension to do error counting over time. The safety manager, the BFCs and the infra-structure will be formally verified and fulfill the highest quality requirements since the most modern and powerful verification techniques will be used. The widely use of an infrastructure required to serve ISO26262 goals reduces the additional effort to a minimum.

9.11 Demonstrator Milestones

At the end of each project year, the following milestones are planned to be achieved:

- 1st project year:
 - Concept.
- 2nd project year:
 - FPGA prototype ready and working stand alone.
- 3rd project year:
 - FPGA prototype proven and working with higher level of analysis tools.

10 Lifetime Drift Model for Discrete Parameters – IFAT – Demonstrator 4.2

Electrical parameters of semiconductor devices, like transistors or diodes, change their values over time. This phenomenon is called lifetime drift. Semiconductor products that can contain many devices have to be designed in a way that they cope with this drift and fulfil their specifications over their lifetimes.

A typical method to control lifetime drift is the introduction of guard bands (JCGM, 2012). Guard bands are the difference between specification and test limits. Thus, the product is tested with tighter limits in order to account for any lifetime drift of its electrical parameters.

Drift models are available for continuous parameters like voltage, current, time, and the like; see, e.g., (Determination of tolerance limits for the reliability of semiconductor devices using longitudinal data, 2017) and (Spline-based drift analysis for the reliability of semiconductor devices, 2021). To our best knowledge, drift models for discrete parameters are not yet available. Examples for discrete parameters at semiconductor devices are, e.g., counters, numerical values of bytes, number of cycles, etc.

In this project, we will develop a lifetime drift model for discrete parameters.

10.1 Target Goals and Achievements

The goals of this demonstrator are the development and publication of a drift model for discrete parameters.

The model should be embedded in existing theories that describe changes of discrete values, like probability distributions which are defined on discrete supports, Markov kernels, stochastic processes where the random variable can only take discrete values and the index set is continuous, etc.

10.2 Demonstrator Structure

The structure of the drift model shall consist of two parts:

- Model for discrete random variables. This is, the basic set of the random variable contains discrete values.
- Model for discretized random variables. This is, the basic set of the random variable contains continuous values and is mapped onto a set of discrete values. This is the case when, e.g., the tester resolution is low. The underlying parameter is continuous, e.g., voltage, but the resolution of the tester is low and the number of possible results forms a countable set.

10.3 Demonstrator Description

The lifetime drift model will be data based. This is, data of reliability stress tests are used to calculate the parameters of the model. These parameters are then applied to products in the field, i.e., in the cars at the customers.

The lifetime drift of discrete parameters can be predicted. This enables the estimation of the RUL.

Furthermore, guard bands can be calculated in order to limit the risk that discrete parameters violate the specified limits during the products' lifetimes.

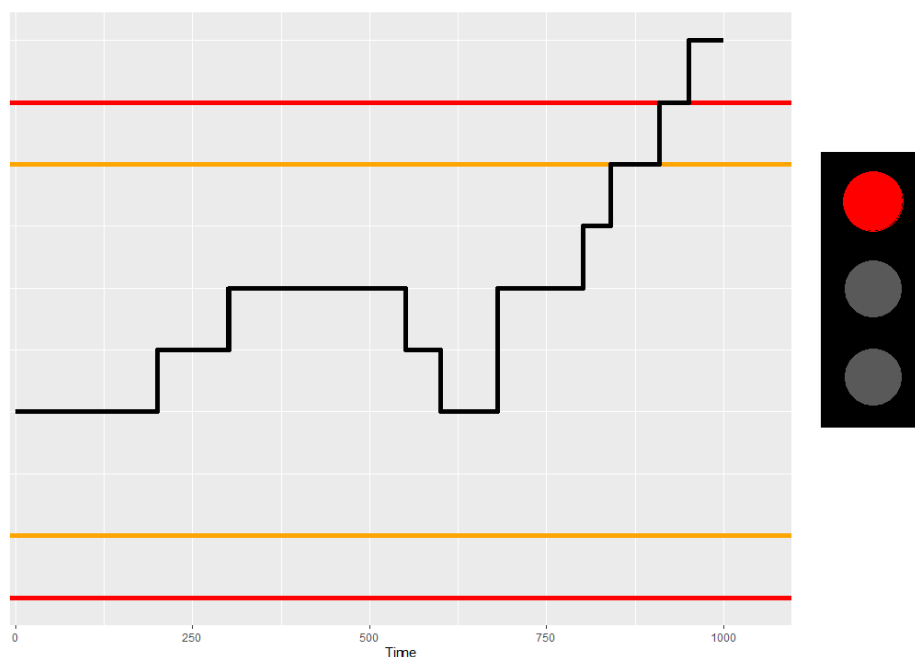


FIGURE 5: DRIFT MODEL FOR DISCRETE PARAMETERS

10.4 Residual Risks

With this model, the probability that discrete parameters drift outside the specified limits can be calculated.

This calculation enables the setup of guard bands in order to achieve the quality targets and to control the risk to drift outside the specified limits.

Furthermore, the RUL can be calculated. The advantage of this model is that the usage of a device and its current health status directly enter into the RUL calculation. With this, we see the possibility to decrease the prediction error of the RUL and to make its estimation more precise.

10.5 Demonstrator Relations to the Main Objectives and Key Targets

10.5.1 Objectives

O2 - Framework for safety validation of ECS value chain (Technical)

Detection of parameter drift on semiconductor level.

O3 - Identification and management of residual risks over the entire ECS value chain (Technical)

Identification of critical parameters to be monitored to reduce residual risk.

O4 - End-user acceptance by trustworthy ECS value chain (Value)

End-User acceptance by trustworthy ECS value chain.

10.5.2 Key Targets

KT3 - Metrics for quality assurance for ECS (mission-oriented qualification, residual risk):

Calculation of remaining useful life of devices.

10.6 Homologation Framework Mapping

The drift model can be used at any level where discrete data change over time, i.e., show lifetime drift. Nevertheless, the focus is on semiconductor devices (subcomponent element in Figure 6).

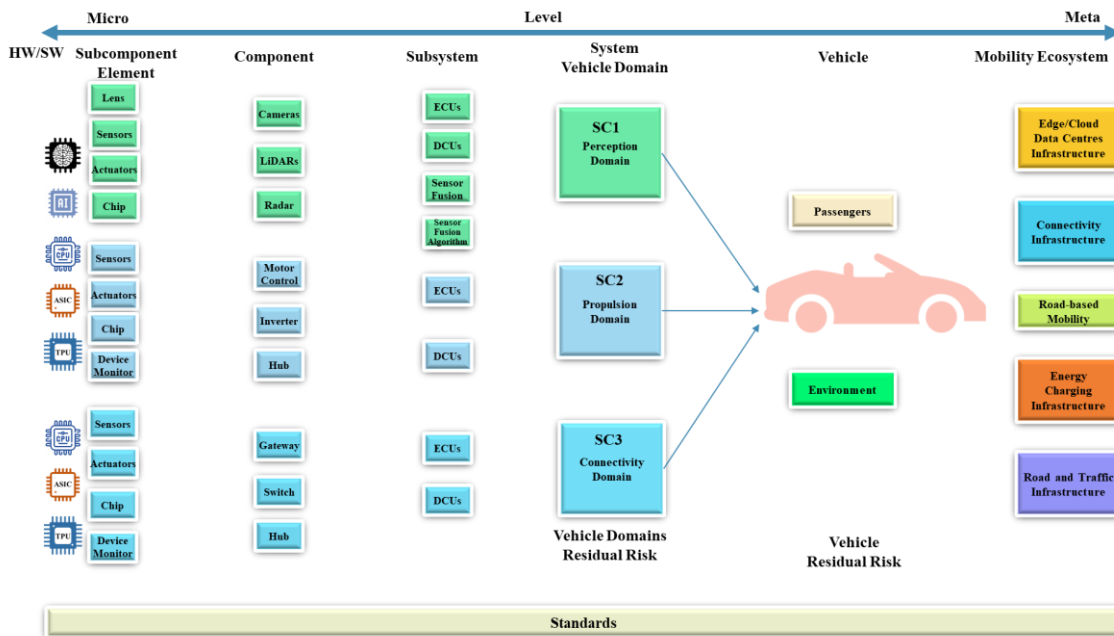


FIGURE 6: SYSTEM STRUCTURE OF AUTOMATED VEHICLES

Accordingly, when the drift model for discrete parameters is used in an IC, it is part of the component approval as well as the functional safety assessment according to ISO 26262.

10.7 Non-functional Requirements, KPIs, and Measures

In Table 10-1 and Table 10-2 we present the most relevant NFRs, KPIs, and measures for demonstrator 4.2.

TABLE 10-1: NFRs, KPIs, AND MEASURES FOR DEMONSTRATOR 4.2 – PERFORMANCE EFFICIENCY

NFR	Performance Efficiency
NFR definition	Fast RUL calculation for usage in final applications.
KPI name	Calculation time.
Description	Lifetime drift analysis can be done in many applications in the automated driving car. It depends on the application in what frequency lifetime drift calculations are needed. This can range from long time intervals until real time calculation. Therefore, the drift model should enable fast calculations to support real time requests of, e.g., RUL.
Measure	Linear models, approximations.
Type of measure	Statistical optimization.
Method of collection and measurement	Standard statistical software, like R.
Demonstrator target	Optimize the runtime while keeping the prediction error for RUL at a minimum level.

KPI for Verification and validation	Runtime of calculation.
-------------------------------------	-------------------------

TABLE 10-2: NFRs, KPIs, AND MEASURES FOR DEMONSTRATOR 4.2 – PERFORMANCE EFFICIENCY

NFR	Performance Efficiency
NFR definition	Minimize calculation resources.
KPI name	Processor usage.
Description	Lifetime drift analysis can be done anywhere in the automated driving car or in the cloud. Nevertheless, frequently it is advisable to run the drift calculation close to the location where the data are generated. Therefore, the model should be made in a way that calculation resources are used at a minimum necessary level in order to perform calculations at edge devices.
Measure	Linear models, approximations.
Type of measure	Statistical optimization.
Method of collection and measurement	Standard statistical software, like R; Aurix.
Demonstrator target	Optimize the needed calculation resources while keeping the prediction error for RUL at a minimum level.
KPI for Verification and validation	Processor usage at calculation.

10.8 Functional Requirements, KPIs, and Measures

IN TABLE 10-3, TABLE 10-4, AND

Table 10-5 we present the three most relevant FRs, KPIs, and measures for demonstrator 4.2.

TABLE 10-3: FRs, KPIs, AND MEASURES FOR DEMONSTRATOR 4.2 – CONTINUITY IN TIME

FR	Continuity in Time
FR definition	The model should calculate drift at any point in time.
KPI name	Continuity in time.
Description	The model parameters are calculated from stress test data. Such data are highly censored. The model must be able to describe discrete parameters outside of the censoring scheme as well – at any point in time.
Measure	Model approach to use continuous time.
Type of measure	Statistical model.
Method of collection and measurement	Simulations of processes of discrete random variables with continuous index sets can be used as data input for the drift model.
Demonstrator target	The drift model has to describe the drift of discrete electrical parameters at any point in time.
KPI for Verification and validation	- Check if the drift model can calculate lifetime drift at any point in time. - Difference between simulated data and calculated data based on the drift model.

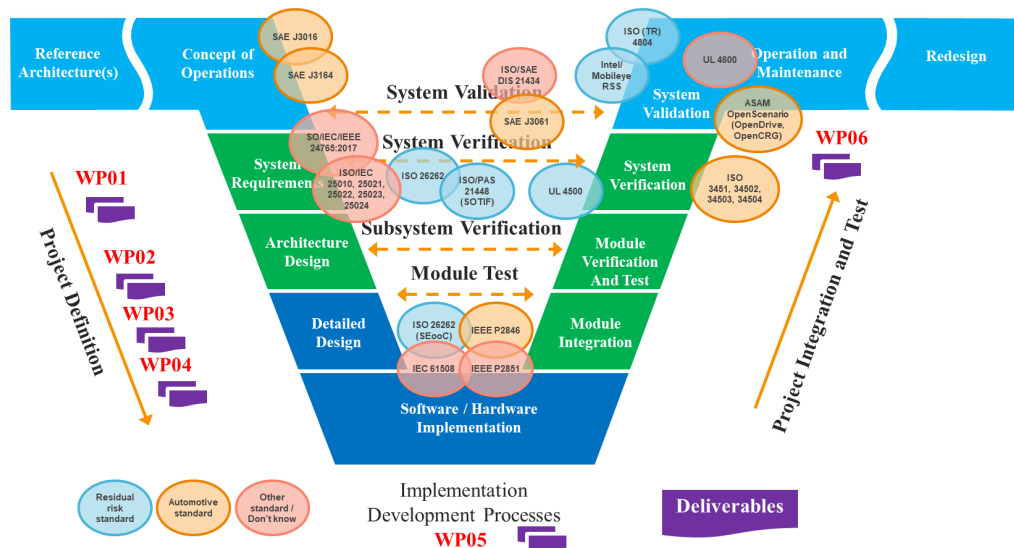
TABLE 10-4: FRs, KPIs, AND MEASURES FOR DEMONSTRATOR 4.2 – ELIMINATE MEASUREMENT ERRORS

FR	Eliminate Measurement Errors
FR definition	The model should take measurement errors into account and remove them from the observed drift, thus calculate to the real drift of discrete electrical parameters.
KPI name	Measurement errors eliminated.
Description	The model should take measurement errors into account and remove them from the observed drift, thus calculate to the real drift of discrete electrical parameters.
Measure	Variance analytical model.
Type of measure	Statistical model.
Method of collection and measurement	Measurement errors at electrical measurements are two-fold: - Tester offsets: These are corrected by testing unstressed devices, so-called reference devices. - Gauge Repeatability and Reproducibility (GR&R): This is the measurement error which is determined in a GR&R-study.
Demonstrator target	The real drift of devices without the measurement errors should be calculated.
KPI for Verification and validation	Difference between simulated data and calculated data based on the drift model.

TABLE 10-5: FRs, KPIs, AND MEASURES FOR DEMONSTRATOR 4.2 – REMAINING USEFUL LIFE (RUL)

FR	Remaining Useful Life (RUL)
FR definition	Estimation of RUL.
KPI name	Efficiency of estimator.
Description	The RUL should be estimated based on the drift behavior until a certain time t . The closer the parameter drifts to its specified limit, i.e., the shorter the RUL, the more accurate the estimation of RUL is.
Measure	Evaluate efficiency of estimator by simulation.
Type of measure	Statistical model, simulation.
Method of collection and measurement	Simulations of processes of discrete random variables with continuous index sets can be used to verify the drift model.
Demonstrator target	Unbiased estimation of RUL with minimized prediction error.
KPI for Verification and validation	Ratio of model prediction error and RUL simulations.

10.9 Mapping to Existing Standards



Standard code	Standard title	Why relevant
AEC Q100	Failure Mechanism Based Stress Test Qualification for Integrated Circuits	This standard is most relevant for the qualification of integrated semiconductor devices.
ISO 26262	Road vehicles - Functional safety	Address safety-related systems that include one or more E/E systems and that are installed in series production passenger cars (with a maximum gross vehicle mass up to 3 500 kg). The standard defines FIT-rate targets for several ASIL-levels. The installation of a monitoring device can turn a failure into a maintenance action. Furthermore, it can be seen as a safety mechanism. This has a direct impact to the FIT-rates of safety critical functions.

10.10 Verification and Validation

The verification and validation of the drift model for discrete parameters will be twofold:

- Simulations: We will run simulations to generate data. The underlying behavior that we simulate is our ground truth. The test cases will cover predictable as well as unforeseen scenarios. We will use the model on such simulated data. This facilitates a firm assessment of the drift model.
- Real data from stress tests: We will use the model at real stress test data. This enables us to refine the model to its practical use and proof it to be used in a wide range of applications.

10.11 Demonstrator Milestones

At the end of each project year, the following milestones are planned to be achieved:

This document and the information contained may not be copied, used or disclosed, entirely or partially, outside of the ArchitectECA2030 consortium without prior permission of the partners in written form.

- 1st project year:
 - Concept of the drift model for discrete parameters.
- 2nd project year:
 - Drift model for discrete parameters refined.
 - First simulations started.
- 3rd project year:
 - Drift model for discrete parameters finalized.
 - Simulations and verification finalized.
 - Publication applied.

11 Virtual Verification & Validation Framework (V3 Framework) – AVL, TUG, SBA – Demonstrator 4.3

The demonstrator is a virtual framework in which verification and validation methods for monitoring devices and advanced driver assistant systems / automated driving (ADAS/AD) controllers can be tested without risks to the environment or system.

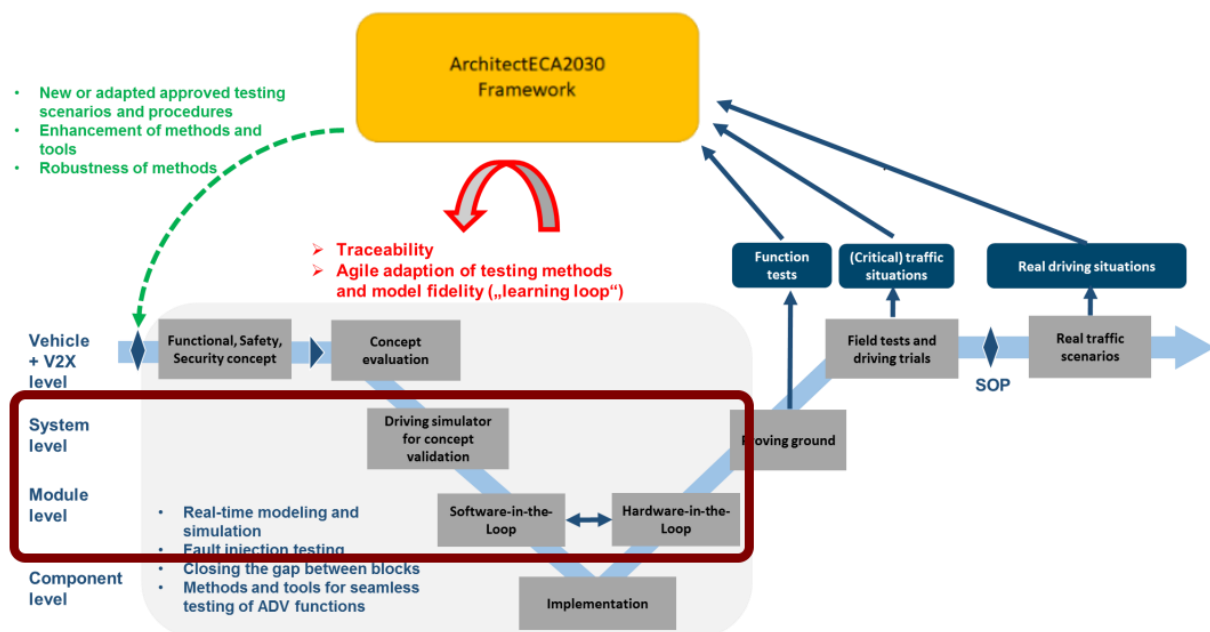
In this testing framework, different approaches for verification and validation can be used and qualitatively compared to each other.

Furthermore, the framework should be reusable for different monitoring devices and ADAS/AD controllers.

11.1 Target Goals and Achievements

The virtual verification and validation framework addresses several topics on system and module level:

- new or adapted approved testing scenarios and procedures,
- enhancement of methods and tools, and
- robustness of methods.



11.2 Demonstrator Structure

The structure of the demonstrator is given in Figure 7.

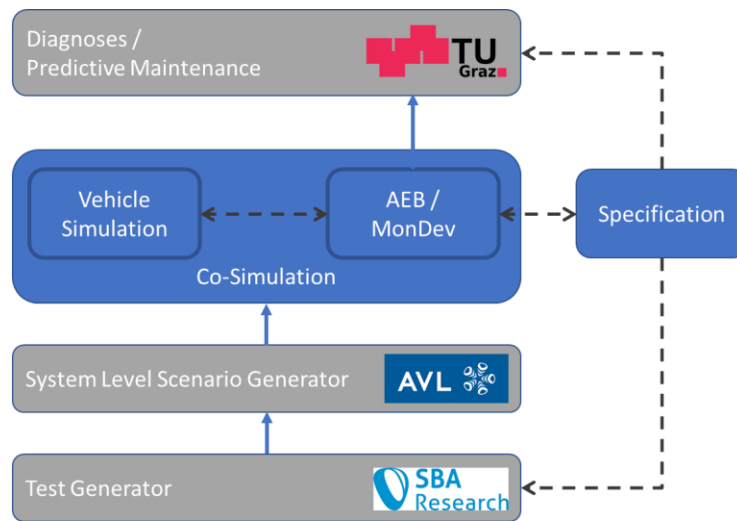


FIGURE 7: STRUCTURE OF V3-FRAMEWORK

11.3 Demonstrator Description

The demonstrator will consist of four main components:

- A co-simulation platform (already existing): On this platform a vehicle simulation as well as a monitoring device or ADAS/AD controller will be simulated. The input for the co-simulation will be a complex scenario.
- Diagnoses/Predictive Maintenance: TUG will contribute algorithms and methods to perform predictive maintenance, state estimation, and reliability and uncertainty measurements based on the output of the monitoring device or the ADAS/AD controller.
- System Level Scenario Generator: AVL will contribute a system level scenario generator that can create complex scenarios (i.e., including behavior) from simple input parameters.
- Test Generator: SBA will contribute a test generator that will generate inputs for the former mentioned system level scenario generator.

11.4 Residual Risks

This demonstrator will reduce the residual risk in the following ways:

- Verification and validation are important to estimate the residual risk. Our demonstrator aims to make this process faster and more reliable, therefore, decreasing the residual risk by increasing the test coverage of monitoring devices and ADAS functions.
- Diagnosis and predictive maintenance help to reduce the residual risk by preventing faults to even happen. For example, we can reduce the residual risk of a component by improving the estimation of its RUL. Hence, we have a more accurate estimation of the residual risk.

11.5 Demonstrator Relations to the Main Objectives and Key Targets

11.5.1 Objectives

O1 - Continuous robust design optimization for each part in the ECS value chain (Technical):

This document and the information contained may not be copied, used or disclosed, entirely or partially, outside of the ArchitectECA2030 consortium without prior permission of the partners in written form.

Testing the monitoring device during development leads to a robust design of the developed component.

O2 - Framework for safety validation of ECS value chain (Technical):

The monitoring device can be tested like in a real environment, without risks to the environment or system. Normal operation as well as critical or “corner case”-operation can be simulated.

O3 - Identification and management of residual risks over the entire ECS value chain (Technical):

Quantitative comparison of different verification and validation approaches enables us to calculate the residual risk.

O4 - End-user acceptance by trustworthy ECS value chain (Value):

Proving the safety of a monitoring device in a virtual environment is a necessary step before even beginning with on-the-road tests.

O5 - Zero emissions, crashes, and congestions by ECA2030 vehicle (Value):

Testing monitoring devices before they are released mitigates errors during operation and, therefore, helps with the vision of zero crashes on the road.

11.5.2 Key Targets

KT1 - Architectures, components, sub-systems enabling virtual development and validation (monitoring device, failure risk):

This demonstrator contributes to the architecture enabling virtual development and validation.

KT2 - Methods and tools to validate the models used in virtual validation (lifetime monitoring, residual risk, methods, and tools):

Methods used in this demonstrator focus on virtual validation of monitoring devices or SAE L3+ autonomous functions.

KT3 - Metrics for quality assurance for ECS (mission-oriented qualification, residual risk):

In this demonstrator we will tackle residual risks in terms of coverage and predictive maintenance/diagnoses.

KT4 - Definition and understanding of test coverage (residual risk, design feedback, lifetime monitoring, aggregated risk):

The demonstrator will adapt well known coverage metrics to be also applicable to the task of validation and verification of SAE L3+ autonomous functions.

KT5 - Methods for shorter validation in respect to acceptable residual risk (methods):

This demonstrator will feature an end-to-end example of a virtual verification and validation framework that significantly decreases testing wall-clock time compared to mileage accumulation.

11.6 Homologation Framework Mapping

The V3-Framework addresses the subsystem and system vehicle domain, see Figure 8.

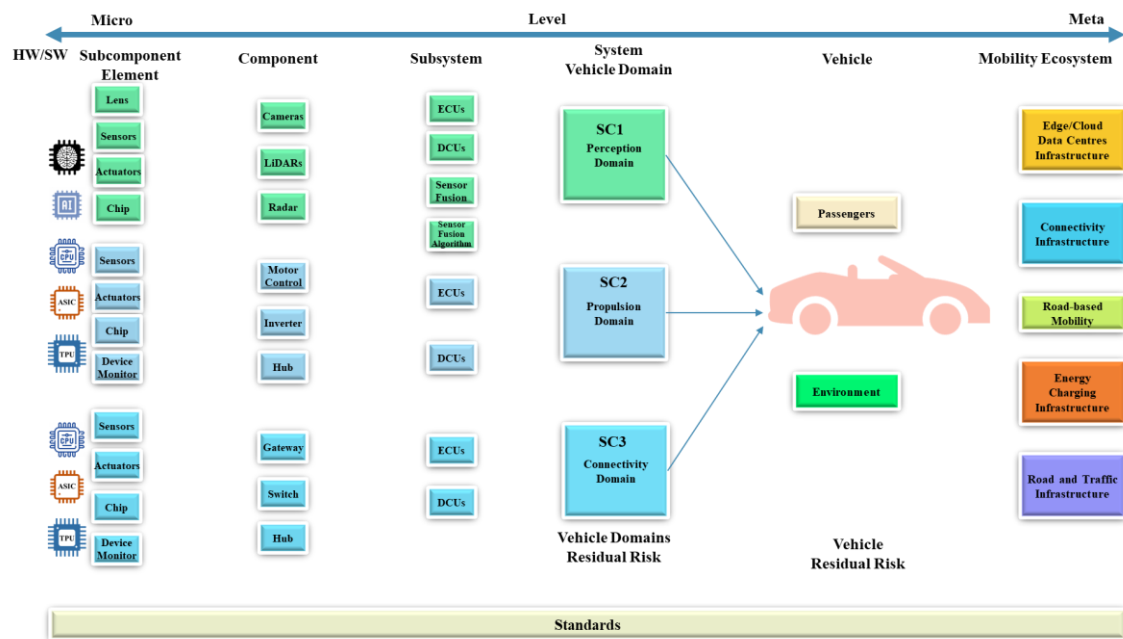


FIGURE 8: SYSTEM STRUCTURE OF AUTOMATED VEHICLES

11.7 Non-functional Requirements, KPIs, and Measures

In Table 11-1, Table 11-2, and Table 11-3 we present the three most relevant NFRs, KPIs, and measures for the demonstrator 4.3.

TABLE 11-1: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 – ROBUSTNESS

NFR	Robustness
NFR definition	Diagnosis robustness.
KPI name	Robustness.
Description	Performance, robustness, and reliability must be specified for the diagnosis functionality. The implemented diagnosis functionality must meet real-time requirements as well as robustness requirements. Moreover, the output should be correct and complete within given boundaries that may be application-specific.
Measure	Carrying out runtime tests for obtaining performance measures for diagnosis. Coming up with a fault injection framework for testing reliability of the diagnosis.
Type of measure	Qualitative.
Method of collection and measurement	For runtime testing as well as for reliability, different application scenarios have to be collected. The scenarios will be tested using the diagnosis implementation. The obtained results will be checked against the pre-defined robustness and performance requirements.
Demonstrator target	All requirements for diagnosis are fulfilled.

KPI for Verification and validation	Fulfilled/not fulfilled.
-------------------------------------	--------------------------

TABLE 11-2: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 - MAINTAINABILITY

NFR	Maintainability
NFR definition	Diagnosis Maintainability.
KPI name	Maintainability.
Description	Maintainability must be achieved for both the verification & validation framework as well as for the underlying system used in the demonstrator (which includes diagnosis functionality). The verification & validation framework as well as the diagnosis part should be easily adaptable in case of different application scenarios. The effort for maintenance shall be minimized.
Measure	Analysis of the proposed methods regarding maintainability. This includes discussing required changes necessary to adapt the verification & validation framework and the diagnosis system for different application scenarios.
Type of measure	Quantitative.
Method of collection and measurement	List all required changes for adaptation and estimate their effort.
Demonstrator target	The estimated effort shall be within pre-specified boundaries. Alternatively, the list of required changes may not include unwanted items.
KPI for Verification and validation	Fulfilled/not fulfilled.

TABLE 11-3: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 - SCALABILITY

NFR	Scalability
NFR definition	Diagnosis scalability.
KPI name	Diagnosis scalability achieved.
Description	Diagnosis functionality shall be configurable and adaptable to fit the particular purpose of the monitoring device. Because of different applications, scalability of diagnosis is important to achieve. The performance of diagnosis and predictive maintenance shall be scalable even when considering larger systems to be diagnosed.
Measure	Performance measure, i.e., calculation time of diagnosis, considering a wide variation of systems of different sizes.
Type of measure	Quantitative.
Method of collection and measurement	Different systems of various sizes ranging from a small to a larger number of components have to be provided. The systems should also make use of different data types. For each system, various diagnosis scenarios have to be defined. For each system and diagnosis scenario, the time required for diagnosis is measured.
Demonstrator target	The diagnosis system should be able to diagnose the given systems within a pre-defined time.

KPI for Verification and validation	For all systems and diagnosis scenarios the diagnosis system is able to provide a result within a pre-defined time limit.
-------------------------------------	---

11.8 Functional Requirements, KPIs, and Measures

In Table 11-4, Table 11-5, and Table 11-6 we present the three most relevant FRs, KPIs, and measures for the demonstrator 4.3.

TABLE 11-4: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 – INPUT/OUTPUT

FR	Input/Output of Monitoring Devices
FR definition	Input/Output requirements.
KPI name	I/O requirements.
Description	The monitoring functionality must have inputs which can be manipulated as well as outputs which can be observed.
Measure	List of predefined inputs/outputs must match the inputs/outputs of the monitoring functionality.
Type of measure	Cross-reference with the list and the inputs/outputs of the monitoring device.
Method of collection and measurement	The predefined list has to be created based on the developed method for verification and validation (specification of the framework). The list of inputs/outputs is created from the monitoring functionality (specification of the monitoring functionality).
Demonstrator target	Full coverage of the list.
KPI for Verification and validation	Fulfilled/not fulfilled.

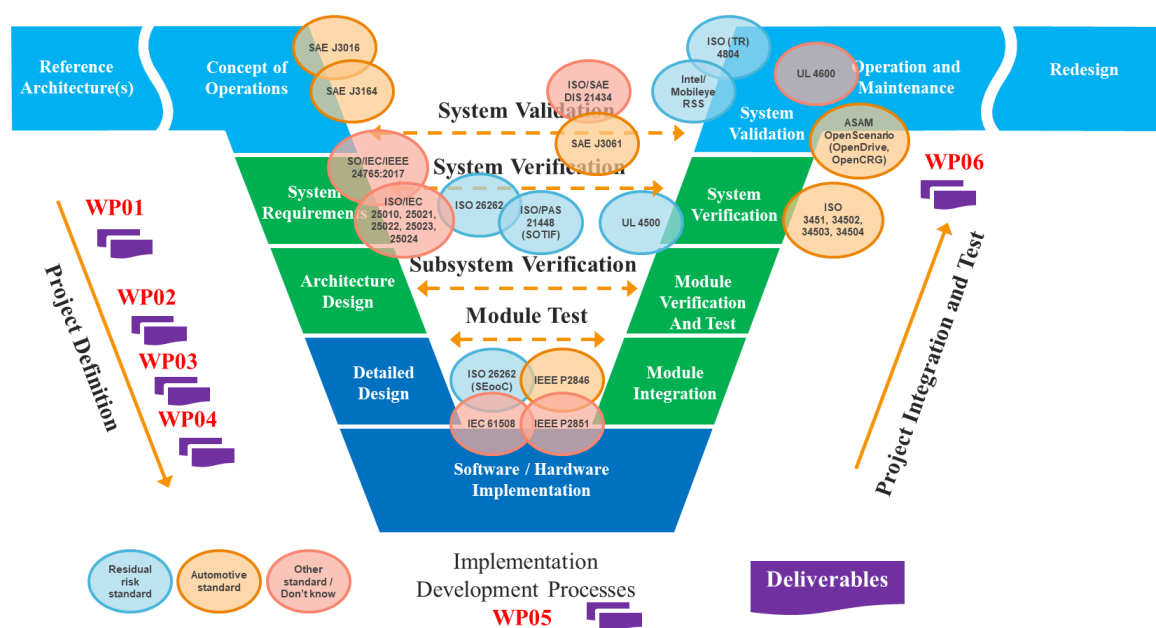
TABLE 11-5: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 - COMPATIBILITY

FR	Compatibility of Monitoring Devices
FR definition	Monitoring functionality must work independent of the environment.
KPI name	Work independent of the environment.
Description	The monitoring functionality must work independent of the environment in which the monitoring functionality is embedded. For example, the monitoring functionality should not try to detect if it is running in a Model in the Loop (MiL), Software in the Loop (SiL), Processor in the Loop (PiL), or Hardware in the Loop (HiL) environment.
Measure	Fulfilled/Not fulfilled.
Type of measure	The specification of the monitoring device must mention such dependencies.
Method of collection and measurement	Assessing the specification of the monitoring device.
Demonstrator target	The monitoring device does not have to be dependent on the environment in which it is running.
KPI for Verification and validation	Fulfilled/not fulfilled.

TABLE 11-6: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.3 - COMPATIBILITY

FR	Compatibility of the Framework
FR definition	The input model must be a discrete model.
KPI name	Discrete model.
Description	For the framework to function correctly, it must be possible to create a discrete model from the specification of the monitoring functionality. This is, only discrete parameter values can be used.
Measure	Parameter/value abstraction to discrete model.
Type of measure	Creating a discrete model out of the specification provides that this requirement is fulfilled.
Method of collection and measurement	Creating a model out of the specification and checking if it is successful or not.
Demonstrator target	Successfully creating a discrete model.
KPI for Verification and validation	Fulfilled/not fulfilled.

11.9 Mapping to Existing Standards



Standard code	Standard title	Why relevant
ISO 26262	Road vehicles – Functional safety	ISO 26262 covers a wide range of topics related to development of automotive products.
SOTIF (ISO PAS 21448)	Safety of the intended functionality	SOTIF describes how functions should be tested. For example, the usage of combinatorial testing for test case derivation is given as an example.
ISO TR 4804	Road vehicles — Safety and cybersecurity for automated driving systems — Design, verification and validation	ISO TR 4804 focuses especially on the development validation of automated driving systems for SAE L3 and L4.
IEEE P2846	Assumptions for Models in Safety-Related Automated Vehicle Behavior	Describes assumptions for safety-related models in ADAS/AD systems. These assumptions include rules of the road and their regional dependencies.
ISO/IEC TR 24028	Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence	Deals with trustworthiness of AI systems. Especially threats and risk and the mitigation of those.
ISO / IEC 25010	Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models	ISO / IEC 25010 deals with software quality. However only a subset of these quality metrics is applicable to SAE L3+ cars.
UL 4600	Standard for Safety for the Evaluation of Autonomous Products	This standard includes safety considerations and risk analysis for autonomous systems (including self-driving cars).
ASAM OpenDrive®	Open Dynamic Road Information for Vehicle Environment	The OpenDrive standard specifies how road networks and static road elements can be described. This enables us to take such a file and use it as an input for a simulation.
ASAM OpenScenario®	ASAM OpenScenario® for dynamic road elements (e.g., vehicles, pedestrians, traffic lights, etc.) and their behavior.	Similar to OpenDrive, OpenScenario specifies dynamic road elements (e.g. vehicles, pedestrians, traffic lights, etc.) and their behavior. Together with an OpenDrive file we can use an OpenScenario file to create a complex simulation.

11.10 Verification and Validation

We will assess the generated test cases and results by hand to be sure that we are creating sensible test cases. Furthermore, we will compare our result with already existing test methods in terms of quality of tests, quantity of tests, and quality of results. To test the diagnosis component, we will run test cases with and without the diagnosis component and compare them to see if the diagnosis was successful.

11.11 Demonstrator Milestones

At the end of each project year, the following milestones are planned to be achieved:

- 1st project year:
 - Initial framework for validation and verification methods for monitoring devices/ADAS controllers is setup. This includes the basic Co-simulation framework.
- 2nd project year:
 - A single diagnosis method and test generation method is implemented. Furthermore, one monitoring device/ADAS controller is used.
- 3rd project year:
 - It is possible to dynamically switch between diagnosis and test generation methods. It is easy to integrate new monitoring devices/ADAS functions into the demonstrator.

12 Automated Driving (AD) Demonstrator – VIF, DATA – Demonstrator 4.4

Robust and fault-tolerant environmental perception in the vicinity of an automated vehicle is a fundamental requirement for the realization of highly automated driving (AD). The decision-making algorithms used in AD vehicles and accompanying advanced driver assistance systems (ADAS) depend solely on the correct and accurate behavior of perception algorithms using the on-board sensors. As a result, modern vehicles are equipped with an increasing number of different sensors such as radar, camera, lidar and ultrasonic sensors that continuously monitor the environment in redundant configurations. Each environmental perception sensor comes with its respective strengths and limitations, which motivates the utilization of redundant sensor configurations. Redundancy is also a technique to increase the ASIL Level (in the sense of ISO26262) of automotive systems, which also applies to ADAS/AD systems to combine the advantages of each sensor.

Environment perception using redundant sensors also require the utilization of sensor fusion techniques. Different environment conditions such as adverse weather situations, changing light, and reflection properties of different materials as well natural or unnatural sensor degradation can be named among several reasons which necessitate reasons for employing redundancy and sensor fusion in ADAS/AD systems. While sensor fusion using Lidar/Camera or Radar/Camera is not a new topic, its full exploitation in 360° surround perception for ADAS/AD systems is not fully studied or exploited. Also, some new combined sensor products include multiple sensor modalities combined in a single component to achieve more robust scene understanding, but they also come with their limitation such as very limited field of view.

This demonstrator represents the overall automated system-level implementation of the project developments to show related use cases for robust perception and monitoring of the risks relevant to environmental perception and automated driving. For this purpose, two typical driving functions, namely ACC (Adaptive Cruise Control) and LKA (Lane Keeping Assistance) were chosen to demonstrate the operation of the respective ADAS functions using at least two sensor modalities. Using the demonstrator, fault initiation and propagation as well as monitoring of the associated risks will be achieved.

12.1 Target Goals and Achievements

Target goal of this demonstrator is the monitoring of the ADAS function behavior subjected to sensor anomalies and sensor signal degradation and the demonstration of robustness of the corresponding ADAS functions under compromised conditions. In this demonstrator, the following achievements are expected:

- Development and demonstration of ADAS functions capable of operating with at least two sensor sources and fuse the corresponding sensor information according to quality of the respective sensor data.
- Demonstrate the functioning of monitoring devices on the overall automated driving system.

12.2 Demonstrator Structure

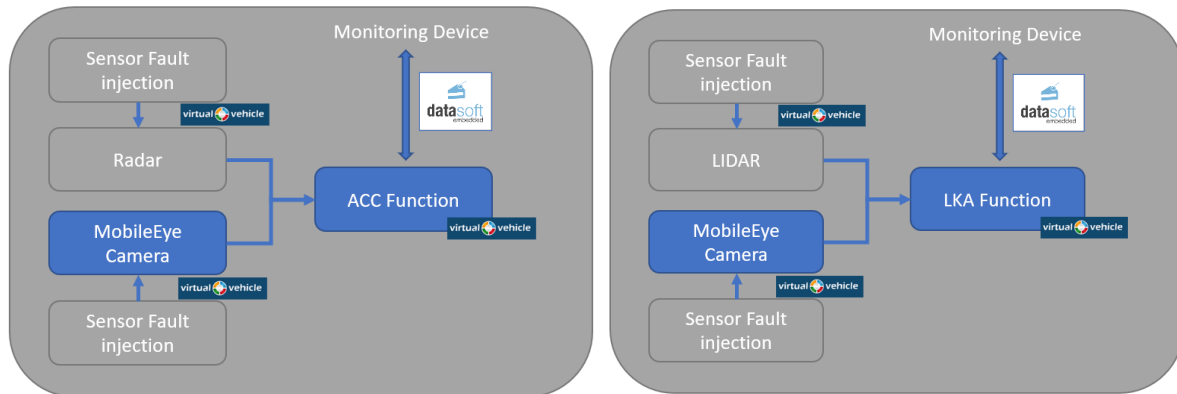


FIGURE 9: DEMONSTRATOR 4.4 STRUCTURE AND PARTNER CONTRIBUTIONS

In this demonstrator, two driving functions, namely Adaptive Cruise Control (ACC) and Lane Keeping Assistance (LKA), will be demonstrated as separate use cases and utilizing different sensor modalities. The demonstrator structure and respective partner contributions for this demonstrator are reflected in Figure 7. Grey color indicates items/components that need to be developed and/or integrated, while blue color respectively indicates readily available items/components. According to the structure, the ACC function shall be based on the sensor fusion utilizing radar and an intelligent camera. In the case of the LKA function, a 360° Lidar shall be combined with an intelligent camera. For both demonstrators, sensor fault injection models will be developed to investigate the fault propagation from component to overall system levels to evaluate robustness and performance of sensor-fusion based driving functions as opposed to those using only single sensor modalities. Real-time risk evaluation and monitoring device functionalities will also be developed and demonstrated for each respective use case.

12.3 Demonstrator Description

At this demonstrator, the monitoring device's functionality is to be demonstrated on a full-scale automated vehicle platform. This demonstrator also shows the robustness of ADAS/AD system to systemic and component failures using two specific ADAS function examples. In this way, fault propagation stemming from component failures (e.g., leading to sensor switch-over) and/or degradation of sensor signals can be emulated and the corresponding effects on the automated driving function performance can be demonstrated using use-case specific KPIs. At this demonstrator, the monitoring device's basic functionality, as well as its performance in predicting failures (as well as initiating certain corrective measures), shall be assessed and demonstrated for automated driving functionalities.

This demonstrator aims to demonstrate the following specific features:

- (1) Fault propagation in the AD system based on a complete automated vehicle demonstrator platform with emulated sensor failures during automated driving operation as given in Figure 10. Figure 10

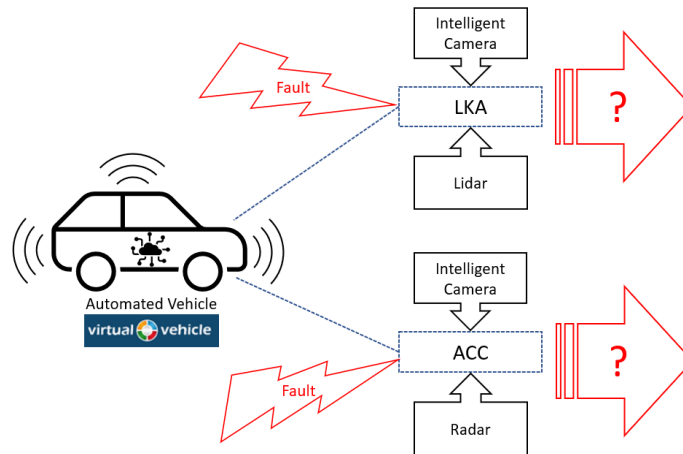


FIGURE 10: DEMONSTRATOR 4.4 CONCEPT DESCRIPTION

- (2) Mapping of software functions (algorithms) enabling monitoring of the device functionality and performance analysis with emulated sensor failures by applying the cross-platform safety & security software platform Open Software Abstraction Model (OSAM) on software component level (related to the algorithms).

Realization of this demonstrator will be based on an automated driving demonstrator (ADD) vehicle from VIF with the following specific capabilities:

- ACC function operational with intelligent camera.
- ACC function operational with 360° Lidar.
- LKA function operational with intelligent camera.
- LKA function operational with long range radar.
- Software interface for running the automated driving functions (“Algo” in Figure 11) and implemented as blackbox models within the OSAM virtualization security middleware from DATA.

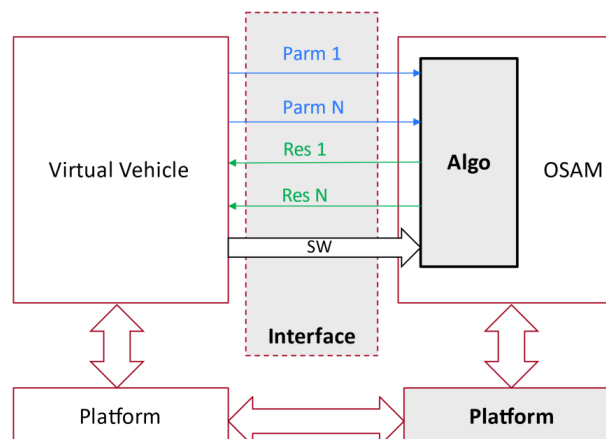


FIGURE 11: DEMONSTRATOR 4.4 INTERFACE BETWEEN OSAM MIDDLEWARE AND ALGORITHM (ADAS FUNCTIONS)

12.4 Residual Risks

Residual risk concept was first introduced in the SOTIF standard (ISO, 2019) and continued in UL4600 (UL, 2020). However, there are no well-defined calculation regulations for it. The definitions therein are rather given as guidelines and recommendations. The white paper “Safety First for AD” (Aptiv Services US, LLC; AUDI AG; Bayrische Motoren Werke AG; Beijing Baidu Netcom Science Technology Co., Ltd; Continental Teves AG & Co oHG; Daimler AG; FCA US LLC; HERE Global B.V.; Infineon Technologies AG; Intel; Volkswagen AG, 2019) also provide good "high level" considerations for it. The definition we utilize is inspired by the risk definition of ISO26262 functional safety standard as well as the FMEA methods, which suggest a function a factor of severity, probability (occurrence), and detectability.

We define the Risk (R) as the combination of the severity (S) of a hazard, the probability or frequency (f) of occurrence of that particular hazard, and the controllability (C),

$$R = F(S, f, C).$$

Residual risk shall be computed as the remaining risk after taking measures against R . The high-level considerations for the residual risk calculation are listed below:

What is state of the art?

- *LKA & ACC typically work only with one sensor.*
- *Some new combined sensor products that include available multiple sensor sources to achieve more robust object detection - > but this is in component level and not in algorithmic level (AD functions).*

What is the residual risk?

- *Issues with single sensor usage (component level): limited FOV, sensitivity to environmental conditions, no redundancy, not suitable for higher levels of autonomy.*

How can the demonstrator contribute to reduce the residual risk?

- *Show the benefit of component redundancy and analyze effects of possible switching in sensors.*

Furthermore, residual risks can be seen as risks which are not addressed by the ISO26262 today. As the today's ISO26262 defines risk related to the ASIL level D ($\Rightarrow 10^{-8}$ to 10^{-7}), residual will get covered by extending this risk levels to ASIL E (e.g., $\geq 10^{-9}$ to 10^{-8}) related to autonomous driving.

The residual risk correlates with the safety and security of the software, i.e., the probability of a malfunction. However, since residual risk occurs on different disciplines in the development process, technical concerns as well as organizational and process related topics needs to get taken into consideration (and provided to SC5 for standardization).

12.5 Demonstrator Relations to the Main Objectives and Key Targets

12.5.1 Objectives

O2 - Framework for safety validation of ECS value chain (Technical):

- Implementation of the monitoring device on a complete vehicle AD demonstrator.
- Fault injection and sensor failure emulation.
- Real-life tests with the AD demonstrator vehicle during automated driving operation.

O3 - Identification and management of residual risks over the entire ECS value chain (Technical):

- Evaluation and performance analysis of the monitoring device on emulated failures for automated driving.
- Demonstration of sensor switch-over to continue automated driving.
- Monitoring device's KPI analysis for detecting the systemic faults & failures.

12.5.2 Key Targets

KT1 - Architectures, components, sub-systems enabling virtual development and validation (monitoring device, failure risk):

- Setup of agreed and qualified methods, tools, and processes to validate and homologate ECA vehicles.
- Develop methods to generate safety- and mission-critical scenarios for highly automated driving (HAD).

KT2 - Methods and tools to validate the models used in virtual validation (lifetime monitoring, residual risk, methods, and tools):

- Design the structure of a general data base for validating models used in virtual validation of automated driving functions (database scheme).
- Record data from perception sensors, vehicle actuators, and vehicle controls.
- Record data using the monitoring devices developed in ArchitectECA2030.

KT5 - Methods for shorter validation in respect to acceptable residual risk (methods):

Development of procedures for merging the real-time simulation of driving environments with real tests on the complete vehicle or on one or several subsystems in parallel to reduce testing time and costs, foster collaboration between different stakeholders, and analyze complex interactions in complete sets of tests.

12.6 Homologation Framework Mapping

The automated driving demonstrator addresses the subsystem and system vehicle domain, see Figure 12.

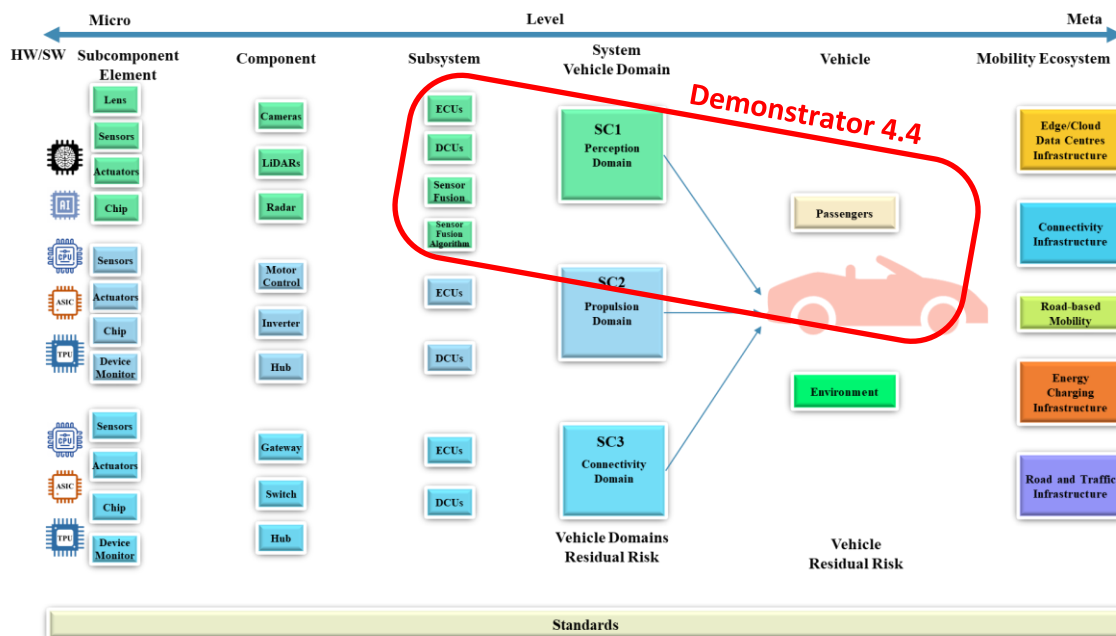


FIGURE 12: SYSTEM STRUCTURE OF MAPPING OF DEMONSTRATOR 4.4 TO THE SYSTEM LEVELS OF HOMOLOGATION FRAMEWORK

12.7 Non-functional Requirements, KPIs, and Measures

In Table 12-1, Table 12-2, and Table 12-3 we present the three most relevant NFRs, KPIs, and measures for the demonstrator 4.4.

TABLE 12-1: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – ADAS FUNCTION ROBUSTNESS

NFR	ADAS Function Robustness
NFR definition	NFR that specifies the robustness requirements of the ADAS functions.
KPI name	Algorithmic robustness.
Description	Robustness requirement of the ADAS functions and related algorithms/driving functions against systemic and component faults, failures, and degradations.
Measure	Sensor fusion with 360° Lidar, intelligent camera, and radar.
Type of measure	ADAS functions using multiple sensor sources and utilizing sensor fusion methods.
Method of collection and measurement	Comparison of single-sensor and sensor-fusion based ADAS functions.
Demonstrator target	Show and monitor the robustness of ADAS functions using redundant sensors and sensor fusion approaches (unscented Kalman filter or similar) against different systemic faults.
KPI for Verification and validation	ADAS function continue functioning despite loss/failure of a sensor.

TABLE 12-2: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – ADAS FUNCTION PERFORMANCE

NFR	ADAS Function Performance
NFR definition	NFR that specifies the performance requirements of the ADAS functions.
KPI name	Algorithmic performance.
Description	Performance requirement of the ADAS functions and related algorithms/driving functions with hard/soft real-time capabilities.
Measure	ADAS functions performing in real time.
Type of measure	Quantified delays.
Method of collection and measurement	Sensor fusion approaches will be implemented in high level programming languages (C++/Python), and cycle times and delays of the algorithms will be measured.
Demonstrator target	Show and monitor the performance of ADAS functions using redundant sensors and sensor fusion approaches (unscented Kalman filter or similar) against different systemic faults.
KPI for Verification and validation	ADAS functions performing at least as good as the single sensor-based implementations.

TABLE 12-3: NFRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – DEVICE DRIVER ABSTRACTION

NFR	Device driver abstraction
NFR definition	Degradation data are sent via a device driver to the application.
KPI name	Microcontroller independent HW interface.
Description	A device driver abstraction model is introduced enabling supplier independent monitoring device functionality (to be standardized within SC5 later on).
Measure	Integration Test Case.
Type of measure	Software.
Method of collection and measurement	Software Test Case Implementation.
Demonstrator target	Monitoring device functionality is realized from the component level to the application level.
KPI for Verification and validation	Transparency on monitoring device.

12.8 Functional Requirements, KPIs, and Measures

IN TABLE 12-4,

TABLE 12-5, AND

Table 12-6 we present the three most relevant FRs, KPIs, and measures for the demonstrator 4.4.

TABLE 12-4: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – SENSOR STATUS AND HEALTH MONITORING

FR	Sensor status and health monitoring
-----------	--

This document and the information contained may not be copied, used or disclosed, entirely or partially, outside of the ArchitectECA2030 consortium without prior permission of the partners in written form.

FR definition	FR that specifies the suitable sensor requirements for the robust ADAS functions.
KPI name	Sensor health status availability.
Description	The information on sensor status, health, and data quality information shall be made available both for the sensor fusion and the monitoring functionality.
Measure	Availability of health status and data quality information.
Type of measure	Real time reporting of relevant status parameters in % scale.
Method of collection and measurement	Sensor fusion approaches will be implemented in high level programming languages (C++/Python) and health status and quality information will be integrated into the algorithms.
Demonstrator target	Robust sensor fusion capability based on the sensor status and quality information that enables the advantage of each sensor type.
KPI for Verification and validation	ADAS function continue functioning despite loss/failure of a sensor.

TABLE 12-5: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – SENSOR DATA FUSION CAPABILITY

FR	Sensor data fusion capability
FR definition	Sensor data fusion capability.
KPI name	Availability of sensor data fusion capability.
Description	The ADAS functions (ACC & LKA) should be able to work with at least two sensor sources including but not limited to an intelligent camera, Lidar, and Radar by the fusion of all sensor data.
Measure	Availability of sensor fusion algorithm.
Type of measure	Availability of real time capable sensor fusion algorithm using sensor object list information as well as sensor status and quality information.
Method of collection and measurement	Sensor fusion approaches will be implemented in high level programming languages (C++/Python), and health status and quality information will be integrated into the algorithms.
Demonstrator target	Robust sensor fusion capability based on the sensor status and quality information that enables the advantage of each sensor type.
KPI for Verification and validation	ADAS functions performing at least as good as the single sensor-based implementations.

TABLE 12-6: FRs, KPIs AND MEASURES FOR DEMONSTRATOR 4.4 – ADAPTIVE SENSOR FUSION CAPABILITY

FR	Adaptive sensor fusion capability
FR definition	Sensor data fusion with adaptive weighting capability.
KPI name	Sensor data fusion with adaptation capability.
Description	In the case of sensor signals with faults, the sensor fusion algorithm should give more weight on the sensor with higher quality data and the weights should be adaptable over-the-air.
Measure	Availability of adaptive sensor fusion algorithm.

Type of measure	Adaption capability for the sensor fusion algorithm taking into account sensor status and quality information and operating conditions.
Method of collection and measurement	Sensor fusion approaches will be implemented in high level programming languages (C++/Python), and health status and quality information will be integrated into the algorithms with adaptation capabilities.
Demonstrator target	Robust sensor fusion capability based on the sensor status and quality information that enables the advantage of each sensor type and adapting to the current operating conditions.
KPI for Verification and validation	Adaption capability for the sensor fusion algorithm considering sensor status and quality information and current operating conditions.

12.9 Mapping to Existing Standards

Standard code	Standard title	Why relevant
SAE J3016	Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles	The SAE J3016 standard does not provide specifications, or otherwise impose requirements on, driving automation systems. Standardizing levels of driving automation and supporting terms serves several purposes, including: 1. Clarifying the role of the (human) driver, if any, during driving automation system engagement. 2. Answering questions of scope when it comes to developing laws, policies, regulations, and standards. 3. Providing a useful framework for driving automation specifications and technical requirements. 4. Providing clarity and stability in communications on the topic of driving automation, as well as a useful short-hand that saves considerable time and effort.
SAE J3018	Guidelines for Safe On-Road Testing of SAE Level 3, 4 and 5 Prototype Automated Driving Systems (ADS)	This standard is not mandatory for the current demonstrator. SAE J3018 serves as basis for safe on-road testing scenarios.

ISO 15622	Intelligent transport systems - Adaptive cruise control systems - Performance requirements and test procedures	This standard covers the performance requirements and specifically performance test procedures and methods for ACC systems.
ISO 20035	Intelligent transport systems - Cooperative adaptive cruise control systems (CACC) - Performance requirements and test procedures	This standard covers the performance requirements and specifically performance test procedures and methods for CACC systems.
ECE/TRANS/ WP.29/2020/ 81	Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems	The intention of the regulation is to establish uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems (ALKS). It contains administrative provisions suitable for type approval, technical requirements, audit and reporting provisions, and testing provisions. In the corresponding VIF demonstrators, the propagation and evaluation of different systemic and synthetically generated faults during the functioning of the ALKS operation will be emulated.
ISO 17387	Intelligent transport systems – Lane change decision aid systems (LCDAS) - Performance requirements and test Procedures	This International Standard specifies system requirements and test methods for Lane Change Decision Aid Systems (LCDAS). The propagation and evaluation of different systemic and synthetically generated faults during ALKS operation will be emulated in the VIF demonstrator.
ISO 17361	Intelligent transport systems - Lane departure warning systems - Performance requirements and test Procedures	This International Standard specifies the definition of the system, classification, functions, human-machine interface (HMI), and test methods for lane departure warning systems. In the demonstrator, the propagation and evaluation of different systemic and synthetically generated faults during the functioning of the ALKS operation will be emulated.
ISO 11270	Intelligent transport systems -Lane keeping assistance systems (LKAS) - Performance requirements and test procedures	This International Standard contains the basic control strategy, minimum functionality requirements, basic driver interface elements, minimum requirements for diagnostics and reaction to failure, and performance test procedures for Lane Keeping Assistance Systems. In the demonstrator, the propagation and evaluation of different synthetically generated faults during the ADAS operation will be emulated.

12.10 Verification and Validation

The demonstrator will be verified in diverse use cases by utilizing simulation approaches. Afterwards, the demonstrators will be validated in real tests.

12.11 Demonstrator Milestones

At the end of each project year, the following milestones are planned to be achieved:

- 1st project year:
 - Complete requirements ready.
- 2nd project year:
 - Sensor drop-out and switch over with ACC and LKA functions demonstrated.
- 3rd project year:
 - Fault injection to a given sensor and monitoring device performance testing during automated driving.

13 SC4 Demonstrators Summarized

Standards vs Demonstrators	4.1	4.2	4.3	4.4
AEC Q100	X	X		
ASAM OpenDrive®			X	
ASAM OpenScenario®			X	
ECE/TRANS/WP.29/2020/81				X
IEEE P2846			X	
ISO 11270				X
ISO 15622				X
ISO 17361				X
ISO 17387				X
ISO 20035				X
ISO 26262:2018 1-12	X	X	X	
ISO/IEC 25010			X	
ISO/IEC TR 24028			X	
ISO TR 4804			X	
SAE J3016				X
SAE J3018				X
SOTIF (ISO PAS 21448)			X	
UL 4600			X	

14 Conclusion

14.1 Contribution to Overall Picture

Monitoring devices are key elements in this project. Our target is to reduce breakdowns and malfunctions of automated driving cars.

Monitoring devices have to be implemented into the system architecture of automated driving vehicles.

Monitoring devices contribute to reducing the residual risk of automated driving cars by turning a potential breakdown situation into a maintenance action.

14.2 Relation to the State-of-the-Art and Progress beyond It

Condition monitoring is a strongly evolving topic. It is frequently embedded in a PHM system. It is getting much more relevance in the case of automated driving cars where no driver can react.

Substantial work has already been done in this field. Nevertheless, there are challenges that go beyond the current state-of-the-art:

- The introduction of a monitoring device implemented in the functional model of an IC.
- Reduction of the residual risk of an automated driving car.
- Implementation of the methodology in the homologation and related standardizations.

Partner/Topic	Description
Hardening automatism for power or motor control design	state-of-the-art: error correcting code (ECC). beyond state-of-the-art: Hardening automatism for power or motor control design to implement automatically bit-flip checking functionality together with a safety controller.
Lifetime drift model	state-of-the-art: lifetime drift model for continuous parameters. beyond state-of-the-art: lifetime drift model for discrete parameters.
Virtual Verification & Validation Framework	state-of-the-art: co-simulation platform, detailed input parameters. beyond state-of-the-art: reduced number of input parameters, combinatorial test cases as input, use the output of the simulation for diagnosis purposes and calculation of residual risks.
Automated Driving Demonstrator	state-of-the-art: LKA based on camera, ACC based on radar. beyond state-of-the-art: smart use of two sensors for LKA and ACC.

14.3 Impact to other WPs, Tasks, and SCs

The work in task 1.4 is the input for the further work in SC4, WP2 and task 2.3. Furthermore, there will be an exchange with the monitoring activities in SC1 – SC3.

Partner/Topic	Description
Hardening automatism	Concept of hardening automatism for power or motor control design incl. BIT-Flip checker functionality.

for power or motor control design	
Lifetime drift model	Concept and development of drift model for discrete parameters.
Virtual Verification & Validation Framework	Methods, architectures, models, and tools for validation of monitoring devices.
Automated Driving Demonstrator	Concept of sensor fusion for LKA and ACC functionalities.

14.4 Contribution to Demonstration

Partner/Topic	Description
Hardening automatism for power or motor control design	Establish hardening automatism for power or motor control design. The monitoring functionality of the Bit-flip checker is specified.
Lifetime drift model	The link between the lifetime drift model and its application at the bit-flip checker is established.
Virtual Verification & Validation Framework	The monitoring device, that uses data from simulation, is specified. Furthermore, we specify a way how to calculate the residual risks based on the test coverage of the defined use cases.
Automated Driving Demonstrator	The monitoring functionality for the use of two sensors for each LKA and ACC is specified.

14.5 Other Conclusions and Lessons Learned

Monitoring devices and functionalities are also established in SCs 1 – 3. We see potential in the ongoing project to intensify our activities with SCs 1 – 3.

Although monitoring devices exist in several fields, their introduction to automated driving cars is a challenge. On the other hand, we see several opportunities to take advantage of already available data and integrate them in a smart way as monitoring devices. Furthermore, this can be included in a PHM system.

15 References

A Framework of Discrete-Event Simulation Modeling for Prognostics and Health Management (PHM) in Airline Industry. **Wang, Z., Cui, Y. und Shi, J. 2017.** Dec. 2017, IEEE Systems Journal, Bd. 11, S. 2227-2238.

A General Stochastic Degradation Modeling Approach for Prognostics of Degrading Systems With Surviving and Uncertain Measurements. **Si, X., Li, T. und Zhang, O. 2019.** Sept. 2019, IEEE Transactions on Reliability, Bd. 68, S. 1080-1100.

A Patent Analysis of Prognostics and Health Management (PHM) Innovations for Electrical Systems. **Liu, Z., et al. 2018.** 2018, IEEE Access, Bd. 6, S. 18088-18107.

Accelerated Degradation Tests: Modeling and Analysis. **Meeker, William Q., Escobar, Luis A. und Lu, Joseph C. 1998.** 1998, Technometrics 40, S. 89-99.

Aptiv Services US, LLC; AUDI AG; Bayrische Motoren Werke AG; Beijing Baidu Netcom Science Technology Co., Ltd; Continental Teves AG & Co oHG; Daimler AG; FCA US LLC; HERE Global B.V.; Infineon Technologies AG; Intel; Volkswagen AG. 2019. *Safety First for Automated Driving.* 2019.

Degradation Modeling Using Stochastic Processes With Random Initial Degradation. **Shen, L., et al. 2019.** Dec. 2019, IEEE Transactions on Reliability, Bd. 68, S. 1320-1329.

Determination of tolerance limits for the reliability of semiconductor devices using longitudinal data. **Hofer, Vera, et al. 2017.** 2017, Quality and Reliability Engineering International, S. 2673-2683.

Economic Analysis of Canary-Based Prognostics and Health Management. **Pecht, M. und Wang, W. 2011.** 7, July 2011, IEEE Transactions on Industrial Electronics, Bd. 58, S. 3077-3089.

European Road Safety Observatory. 2021. *Road safety thematic report - Fatigue.* Brussels : European Commission - Directorate General for Transport, 2021.

Eurostat. 2020. EU population in 2020. *Newsrelease.* s.l. : Eurostat, 10. July 2020.

ISO. 2018. ISO 17359. *Condition monitoring and diagnostics of machines — General guidelines.* s.l. : ISO, 2018.

— **2015.** ISO 18129. *Condition monitoring and diagnostics of machines — Approaches for performance diagnosis.* s.l. : ISO, 2015.

— **2013.** ISO 20958. *Condition monitoring and diagnostics of machine systems — Electrical signature analysis of three-phase induction motors.* s.l. : ISO, 2013.

— **2018-1.** ISO 26262-1. *Road vehicles - Functional Safety - Part 1: Vocabulary.* s.l. : ISO, 2018-1.

— **2018.** ISO 26262-11. *Road vehicles - Functional Safety - Part 11: Guidelines on applications of ISO 26262 to semiconductors.* s.l. : ISO, 2018.

— **2018-3.** ISO 26262-3. *Road vehicles - Functional safety Part 3: Concept phase.* s.l. : ISO, 2018-3.

— **2011.** ISO/IEC 25010. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuARE) - System and software models.* s.l. : ISO/IEC, 2011.

— **2019.** ISO/PAS 21448. *Road vehicles - Safety of the intended functionality.* s.l. : ISO, 2019.

JCGM. 2012. *Evaluation of measurement data - The role of measurement uncertainty in conformity assessment.* s.l. : JCGM, 2012. Standard. JCGM 106.

Reliability Growth Modeling. **Sen, A. 2014.** 2014, Wiley StatsRef: Statistics Reference Online.

Reliability Growth Testing. **Pulcini, G. 2014.** 2014, Wiley StatsRef: Statistics Reference Online.

SAE. 2018. JA6268TM. *Design & Run-Time Information Exchange for Health-Ready Components.* s.l., USA : SAE, 04 2018.

—. **2014.** *Verification & Validation of Integrated Vehicle Health Management Systems and Software.* s.l. : SAE, 2014.

Spline-based drift analysis for the reliability of semiconductor devices. **Hofer, Vera, Nowak, Thomas und Lewitschnig, Horst. 2021.** 2021, Advanced Theory and Simulations, S. submitted, in production.

Statistik Austria. 2020. *Straßenverkehrsunfälle Jahresergebnisse 2019.* s.l. : Statistik Austria, 2020.

UL. 2020. UL 4600. *Standard for Safety for Evaluation of Autonomous Products.* s.l. : UL, 2020.

16 List of Figures

Figure 1: Monitoring Functions in the System Architecture	14
Figure 2: Bathtub Curve - Evolution of Failure Rate Over Time	20
Figure 3: BFC Embedded in an IC	21
Figure 4: System Structure of Automated Vehicles	23
Figure 5: Drift Model for Discrete Parameters	28
Figure 6: System Structure of Automated Vehicles	29
Figure 7: Structure of V3-Framework	35
Figure 8: System Structure of Automated Vehicles	37
Figure 9: Demonstrator 4.4 Structure and Partner Contributions	44
Figure 10: Demonstrator 4.4 Concept Description	45
Figure 11: Demonstrator 4.4 Interface Between OSAM Middleware and Algorithm (ADAS Functions)	45
Figure 12: System Structure of Mapping of Demonstrator 4.4 to the System Levels of Homologation Framework	48

17 List of Tables

Table 4-1: Partner Contribution Overview	6
Table 9-1: NFRs, KPIs and Measures for Demonstrator 4.1 – Reliability	23
Table 9-2: NFRs, KPIs and Measures for Demonstrator 4.1 - Reliability	24
Table 9-3: FRs, KPIs and Measures for Demonstrator 4.1 - Performance	24
Table 9-4: FRs, KPIs and Measures for Demonstrator 4.1 – Failure Handling	24
Table 9-5: FRs, KPIs and Measures for Demonstrator 4.1 - Scalability	25
Table 10-1: NFRs, KPIs, and Measures for Demonstrator 4.2 – Performance Efficiency	29
Table 10-2: NFRs, KPIs, and Measures for Demonstrator 4.2 – Performance Efficiency	30
<i>This document and the information contained may not be copied, used or disclosed, entirely or partially, outside of the ArchitectECA2030 consortium without prior permission of the partners in written form.</i>	57

Table 10-3: FRs, KPIs, and Measures for Demonstrator 4.2 – Continuity in Time	30
Table 10-4: FRs, KPIs, and Measures for Demonstrator 4.2 – Eliminate Measurement Errors	31
Table 10-5: FRs, KPIs, and Measures for Demonstrator 4.2 – Remaining Useful Life (RUL)	31
Table 11-1: NFRs, KPIs and Measures for Demonstrator 4.3 – Robustness.....	37
Table 11-2: NFRs, KPIs and Measures for Demonstrator 4.3 - Maintainability.....	38
Table 11-3: NFRs, KPIs and Measures for Demonstrator 4.3 - Scalability.....	38
Table 11-4: FRs, KPIs and Measures for Demonstrator 4.3 – Input/Output	39
Table 11-5: FRs, KPIs and Measures for Demonstrator 4.3 - Compatibility	39
Table 11-6: FRs, KPIs and Measures for Demonstrator 4.3 - Compatibility	40
Table 12-1: NFRs, KPIs and Measures for Demonstrator 4.4 – ADAS Function Robustness.....	48
Table 12-2: NFRs, KPIs and Measures for Demonstrator 4.4 – ADAS Function Performance	49
Table 12-3: NFRs, KPIs and Measures for Demonstrator 4.4 – Device Driver Abstraction	49
Table 12-4: FRs, KPIs and Measures for Demonstrator 4.4 – Sensor Status and Health Monitoring ...	49
Table 12-5: FRs, KPIs and Measures for Demonstrator 4.4 – Sensor Data Fusion Capability	50
Table 12-6: FRs, KPIs and Measures for Demonstrator 4.4 – Adaptive Sensor Fusion Capability	50

18 Internal Review

Reviewer 1: Petr Blaha (BUT)

Reviewer 2: Dr. Andreas Soppa (VW)

1. Is the deliverable in accordance with:

	Answer	Comments	Type*		Answer	Comments	Type*
(i) the description of work?	yes/ ~~no~~		M/m/a		yes	After discussions.	M
(ii) the international state of the Art?	yes/ ~~no~~ / ~~not relevant~~		M/m/a		yes		m

2. Is the quality of the deliverable in a status that:

	Answer	Comments	Type*		Answer	Comments	Type*
allows to send it to ECSEL JU?	Yes/ ~~no~~	The internal review was realized in two steps. The comments from first stage were successfully integrated.	M/m/a		yes		M
(ii) needs improvement of the writing by the authors of the deliverable?	~~Yes~~ /no	Some minor corrections and typos should be corrected	M/m/a		yes	I have made some lines in the tables.	a

		after second stage.				
(iii) needs further work by the partners responsible for the deliverable?	~~Yes~~ /no		M/m/a	no		M
(iv) Needs to fulfill the following suggestions?	~~Yes~~ /no		M/m/a	no		M

* Type of comments: M = major comment; m = minor comment; a = advise

- Last page of the document is intended to be blank! -