Horizon 2020 Program (2014-2020)

A computing toolkit for building efficient autonomous applications leveraging humanistic intelligence (TEACHING)

D3.1: Initial Report on Engineering Methods and Architecture Patterns of Dependable CPSoS[†]

Contractual Date of Delivery	31/10/2020
Actual Date of Delivery	30/12/2020
Deliverable Security Class	Public
Editor	*Georg MACHER (TUG)*
Contributors	*[(TUG) Jürgen DOBAJ, Matthias SEIDL, Maid DZAMBIC]* *[(I&M) Lorenzo Giraudi, Roberta Peroglio]* *[(ITML) Angela Dimitriou]* *[(HUA) Charalampos Davalas, Dimitrios Michail, Christos Sardianos, Konstantinos Tserpes, Iraklis Varlamis]* *[(AVL) Omar Veledar]* *[(UniPi) Davide Bacciu]* *[(M) Marilina DE GENNARO, Calogero CALANDRA, Sara POTENZA]*
Quality Assurance	*Omar VELEDAR (AVL)*

[†] *The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871385.*

The TEACHING Consortium

University of Pisa (UNIPI)	Coordinator	Italy
Harokopio University of Athens (HUA)	Principal Contractor	Greece
Consiglio Nazionale delle Ricerche (CNR)	Principal Contractor	Italy
Graz University of Technology (TUG)	Principal Contractor	Austria
AVL List GmbH	Principal Contractor	Austria
Marelli Europe S.p.A. (M)	Principal Contractor	Italy
Ideas & Motion	Principal Contractor	Italy
Thales Research & Technology	Principal Contractor	France
Information Technology for Market Leadership	Principal Contractor	Greece
Infineon Technologies AG	Principal Contractor	Germany

Document Revisions & Quality Assurance

Internal Reviewers

1. *Eric ARMENGAUD (AVL List GmbH)*
2. *Omar VELEDAR, (AVL List GmbH)*
3. *Eugen BRENNER (Graz University of Technology)*

Revisions

Version	Date	By	Overview
1.2.2	30/12/2020	Editor	Final
1.2.1	27/12/2020	Reviewer	Comments on draft
1.2.0	23/12/2020	Editor	Review Ready Version
1.1.0	04/12/2020	Editor	1st Version Chapter Inputs
1.0.0	02/11/2020	Editor	Document Init, ToC & Chapter responsibility definition

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACC	Adaptive Cruise Control
AD	Autonomous Driving
ADAS	Advanced Driver Assistant Systems
ADS	Automated Driving Systems
AEBS	Advanced Emergency Braking System
AI	Artificial Intelligence
ALKS	Automated Lane Keep System
ASPICE	Automotive SPICE
API	Application Programming Interface
AV	Autonomous Vehicle
AVPS	Automated valet parking systems
CACC	Cooperative Adaptive Cruise Control System
CPS	Cyber-Physical System
CPSoS	Cyber-Physical System of Systems
CPU	Central Processing Unit
DoS	denial-of-service
DIS	Draft International Standard
Dx.y	Deliverable (x = Work Package number and y = deliverable number)
E/E	Electric/Electronical
EC	European Commission
ECU	Electronic Control Unit
EDR	Event Data Recorder
FMS	Flight Management System
HIDS	Host-based Intrusion Detection System
HITL	human-in-the-loop
HMI	Human-Machine Interface
HOTL	human-on-the-loop
HOIC	human-in-command
HR	heart rate
HRV	heart rate variability
HW	Hardware
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISO	International Standardisation Organisation
LSAD	Low-speed automated driving
LSRA	Limited Speed Range Adaptive Cruise Control
IT	Information Technology
MBSE	Model-Based System Engineering

ML	Machine Learning
NN	Neural Network
ODD	Operational Design Domain
PADS	Partially Automated In-Lane Driving Systems
PALS	Partially Automated Lane Change System
PAPS	Partially automated parking systems
PAS	Public Available Standard
PDCMS	Pedestrian Detection and Collision Mitigation Systems
R2L	remote-to-local
RC	remote controlled
RNN	Recurrent Neural Networks
SAE	Society of Automotive Engineers
SLC	Safety Lifecycle
SoM	System on Module
SotIF	Safety of the Intended Functionality
SotA	State of the Art
SPICE	Software Process Improvement and Capability Determination
SW	Software
SysML	Systems Modelling Language
U2R	user-to-root
UN	United Nations
UNECE	United Nations Economic Commission for Europe
UC	Use Case
UML	Unified Modelling Language
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2x	Vehicle-to-everything
VANET	Vehicular ad-hoc networks
VDA	German Car Manufacturer Association (Verein Deutscher Automobilhersteller)
V&V	Verification and Validation
WCET	Worst Case Execution Time
WD	Working Draft
WIP	Work in Progress
WP	Work Package
XAI	explainable AI

Executive Summary

TEACHING project's workpackage 3 aims to enhance the project's technology brick development by building a dependable engineering environment that supports the development of self-adaptive artificial humanistic intelligence in a dependable manner. In this context, WP3 focuses on the establishment of engineering methods, architectural concepts and design patterns that can be used to develop dependable and AI-based autonomous system development.

Dependability engineering of adaptive, cloud-based and/or AI-based systems is still a topic where first concepts need to be instantiated (like practical processes and methods, covering the whole lifecycle). The assurance of dependability, especially considering novel AI-based and/or dynamical runtime-based approaches, is still an open issue that is lacking in common solution so far.

To that aim, the goal of this deliverable is to identify gaps with existing solutions for the management of CPSoSs throughout their life cycle including design and operational phases (architectural frameworks, conceptual models, process frameworks etc.). Based on this analysis, architectural, process and development framework will be developed to support automated dependability evaluation of CPSoS (Obj. 5 of TEACHING project).

In compliance with its intended purpose for the TEACHING project, this document (D3.1) presents the established body of knowledge of WP3 at Milestone 1. Therefore, this deliverable does not focus on the development of TEACHING technology bricks only, but also enhances the project via a different view focusing on development processes and engineering methods. The technical content of the document also serves the purpose of enhancing other WPs activities concerned with business view perspectives and use-cases, system architecture concepts and SotA, as well as the development of the TEACHING technology bricks.

The content of the deliverable results from activities undertaken in WP3 in the first project phase and covers the following sections: (a) the current state of practice in terms of dependable engineering methods, architectural concepts, as well as regulation activities and industrial working groups, (b) relation to TEACHING project requirements, (c) dependability architectures concepts and architecture pattern for different scenarios, (d) approaches for application of AI for ensuring of CPSoS dependability, and (e) dependability engineering of cloud-connected AI-based systems.

This report depicts the currently established dependability engineering methods and design patterns by WP3 at project milestone 1 and will be elaborated continuously throughout the remaining project duration. Therefore, this deliverable will be updated and enhanced by deliverable D3.2.

1 Introduction

While the TEACHING project's main focus is on mission-critical, energy-sensitive autonomous systems and the development of technology bricks for humanistic AI concepts Workpackage 3 aims to enhance the TEACHING project technology brick development to build a dependable engineering environment to support the development of a self-adaptive artificial humanistic intelligence in a dependable manner. In this context, WP3 focuses on the establishment of engineering methods, architectural concepts and design patterns that can be used to develop dependable and AI-based autonomous system development.

This deliverable focuses on the development of methodologies, architectural frameworks and tools to enforce dependable engineering of novel CPSoS (Obj. 4) and represents the established body of knowledge of WP3 at Milestone 1 of the TEACHING project. WP3 will continue to elaborate this body of knowledge throughout the remaining project duration and therefore outdate this deliverable by deliverable D3.2 at Milestone 2.

The goal of this deliverable is to identify gaps with existing solutions for the management of CPSoSs throughout their life cycle including design and operational phases (architectural frameworks, conceptual models, process frameworks etc.). Based on this analysis, architectural, process and development framework will be developed to support automated dependability evaluation of CPSoS (Obj. 5).

The following sections define the relation to other deliverables of Milestone 1. Section 2 describes the current state of practice in terms of dependable engineering methods, architectural concepts, as well as regulation activities and industrial working groups. Section 3 briefly highlights the TEACHING requirements related to WP3 or affecting the applicable approaches of WP3 for TEACHING technology bricks. Section 4 looks into dependability architectures concepts and describes the pattern for applicability for different scenarios and their impact. Similarly, Section 5 presents the approaches for the application of AI for ensuring of CPSoS dependability. Section 6 enhances the view on the dependability engineering of cloud-connected AI-based systems and Section 7 concludes the deliverable with an outlook on the work for Milestone 2.

The content of the deliverable is resulting from the activities of all WP3 tasks in the first project phase. The intention here is to have a first release version of the WP3 research activities to continue building TEACHING technology bricks based on these methods and patterns and to have a more fluid interaction between the work packages. The concluding section briefly wraps up the main findings of WP3 activities to date.

This report depicts the currently established dependability engineering methods and design patterns by WP3 and will be elaborated continuously throughout the remaining project duration. Therefore, this deliverable will be amended by deliverable D3.2.

1.1 Relationship with other deliverables

In compliance with its intended purpose for the TEACHING project, this document (D3.1) presents the established body of knowledge of WP3 at Milestone 1, which focuses on the development of methodologies, architectural frameworks and tools to enforce dependable engineering of novel CPSoS. Therefore, this deliverable does not focus on the development of TEACHING technology bricks only, but also enhances the project via a different view focusing on development processes and engineering methods. The technical content of the document also serves the purpose of informing other WPs and associated deliverables, which are concerned with business view perspectives and use-cases (D5.1 [1]), system architecture concepts and SotA (D1.1 [2]) and the TEACHING technology bricks (D2.1 [3] and D4.1 [4]). Those related deliverables D1.1, D2.1, D4.1 and D5.1 are listed in Table 1; all of which are grouped with this deliverable as a mean of milestone verification. That is the first project milestone, entitled *Release of the TEACHING design (requirements, specification and architecture)*. The mapping of the viewpoints of the technical WPs and of the dependability engineering approaches that are considered to support the development, as well as the integration intentions of the TEACHING technology bricks in domain use-cases is depicted in Figure 1.

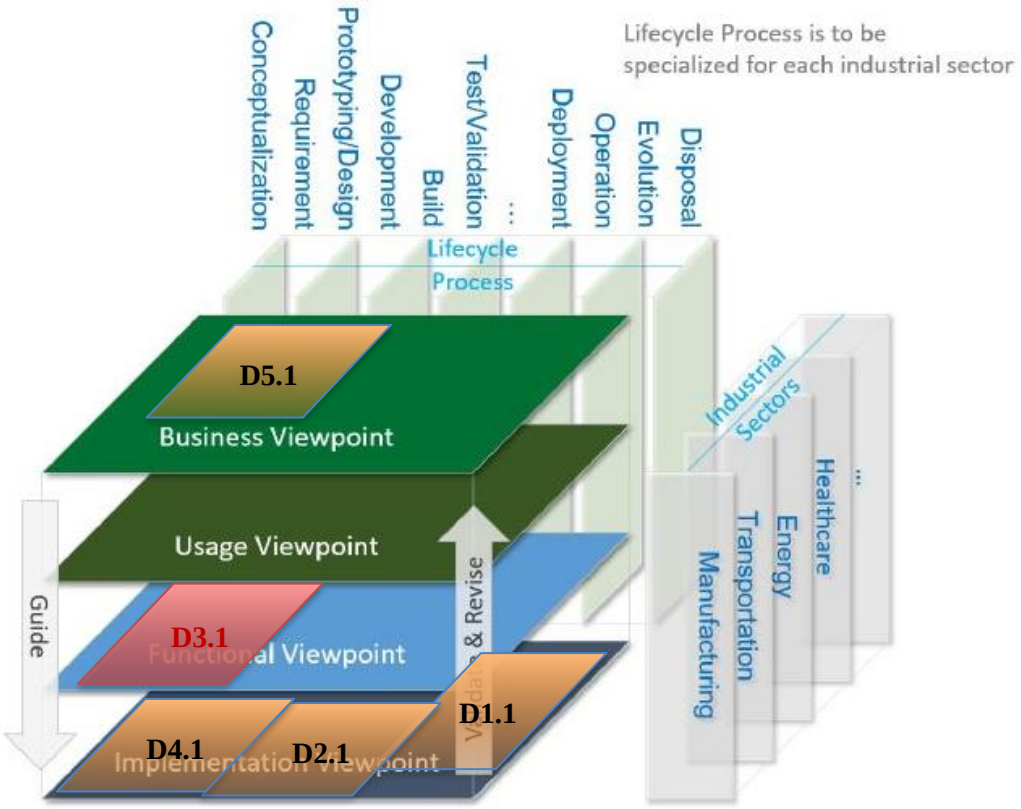


Figure 1 Depiction of the IIRA Viewpoints from [1] and mapping of focus of TEACHING Deliverables MS1

[1] https://iiot-world.com/industrial-iot/connected-industry/iic-industrial-iot-reference-architecture/

Table 1 Deliverable grouping for verification of TEACHING Milestone 1

D1.1	Report on TEACHING related technologies SoA and derived CPSoS requirements
D2.1	State-of-the-art analysis and preliminary requirement specifications for the computing and communication platform
D3.1	Initial Report on Engineering Methods and Architecture Patterns of Dependable CPSoS
D4.1	Report on first release of the AIaaS system
D5.1	Initial use case specifications

2 State-of-the-art Analysis

This section of the document presents state-of-the-art approaches to dependability engineering, regulatory approaches in the automotive domain, and focus group activities. These serve as the basic framework within which WP3 research activities are conducted and possible solutions are evaluated. A detailed analysis of the SotA in terms of research and technology challenges on dependability engineering is available in deliverable D1.1 [2].

2.1 Definition of Dependability

2.1.1 Introduction

Embedded systems play a crucial role in many of the devices that we use today. They are small computers that are much less powerful than standard laptops and desktop computers and are built for performing a specific task. The appliance of embedded systems ranges from simple devices such as coffee makers and air fryers to much more complex systems such as vehicles, airplanes and space shuttles. When multiple embedded systems are interconnected and function together as part of a bigger system, then we talk about distributed embedded systems. In a distributed system, all sub-systems communicate with each other by accessing a network and utilizing a communication protocol for exchanging messages.

As embedded systems are also part of safety-critical applications such as vehicles and airplanes, it must be ensured that the service they provide does not fail under any circumstances. In other words, they must be dependable. There are various definitions of dependability in this sense, all of which revolve around the same idea. In a technical report in 2001, Laprie, Avizienis, and Randell defined dependability as the ability to deliver service that can justifiably be trusted [5]. This defines dependability as a subjective property that cannot be measured and quantified [6]. For making dependability also somewhat measurable, a more recent definition presents dependability as the ability of a system to avoid service failures that are more frequent or more severe than is acceptable [7]. With this definition, we can decide what is too frequent and too severe and express dependability as a probability over time. To understand dependability, we must first understand what is meant under the term “service”.

The **service** delivered by a system can be defined as the behaviour of the system, as it is perceived by a receiving system [5]. Thereby we say that a system provides **correct service** when it implements the system function, or with other words, when the system does what it is supposed to do [5]. The time period for which the system is performing as intended is called **service delivery**, while the time period where no correct service is provided is called **service outage** [5]. This is illustrated in Figure 2.

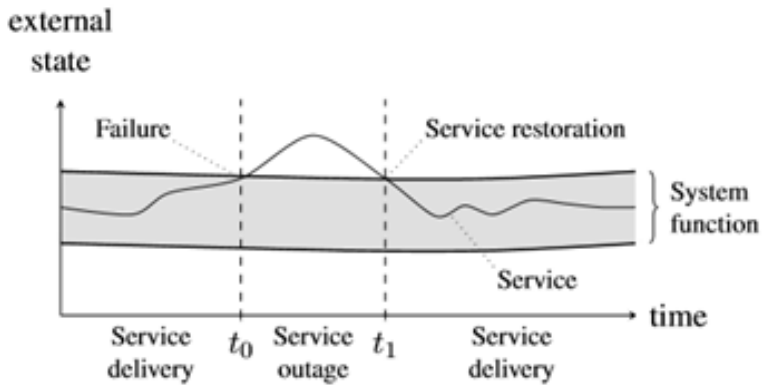


Figure 2 Illustration of service delivery and service outage

The horizontal axis represents the time, while the vertical axis shows the state of a system. The shaded region represents the boundaries of the system states when correct service is delivered. In Figure 2, correct service is delivered until time t0, at which point a failure happens and the system states are no longer in the shaded area. From time t0 to t1, service outage is present. At time t1, service restoration is achieved and correct service delivery continues.

Now, coming back to dependability, it has been defined as a summarizing concept consisting of multiple attributes. Those attributes include:

- **Availability** - "readiness for correct service" [7]. This attribute is often expressed as a function of time and represents the probability of correct service at a given time. Hence, it indicates how likely it is that a system will provide correct service whenever we might want it.
- **Reliability** - "continuity of correct service" [7]. Similar to availability, reliability is also expressed as a function of time. This attribute represents the probability of a correct service being delivered during a specific time interval.
- **Safety** - "absence of catastrophic consequences on the user(s) and the environment" [7]. Safety is also expressed as a function of time and represents the probability that no failures of a system will occur in a given time period, which could lead to catastrophic events. Thus, this attribute indicates how likely it is that no catastrophes will occur, regardless if correct or incorrect service is delivered.
- **Integrity** - "absence of improper system alterations" [7]. Less commonly expressed as a function of time. This attribute includes both intentional and unintentional interference in the system. Thereby, it does not matter if the interference is attempted by any system which is part of the environment or the system itself.
- **Maintainability** - ability to undergo modifications and repairs" [7]. Also not commonly expressed as a function of time. Maintainability indicates, amongst other things, how easy it is to restore a system which provides incorrect service to provide correct service again.

Besides dependability, another concept, which is often addressed when designing fault-tolerant systems, is security. Security is a composite of the dependability attributes availability and integrity, with the addition of the attribute of confidentiality.

Confidentiality represents the absence of unauthorized disclosure of information" [7]. Figure 3 shows the overview of the attributes, threats and means to attain dependability and security.

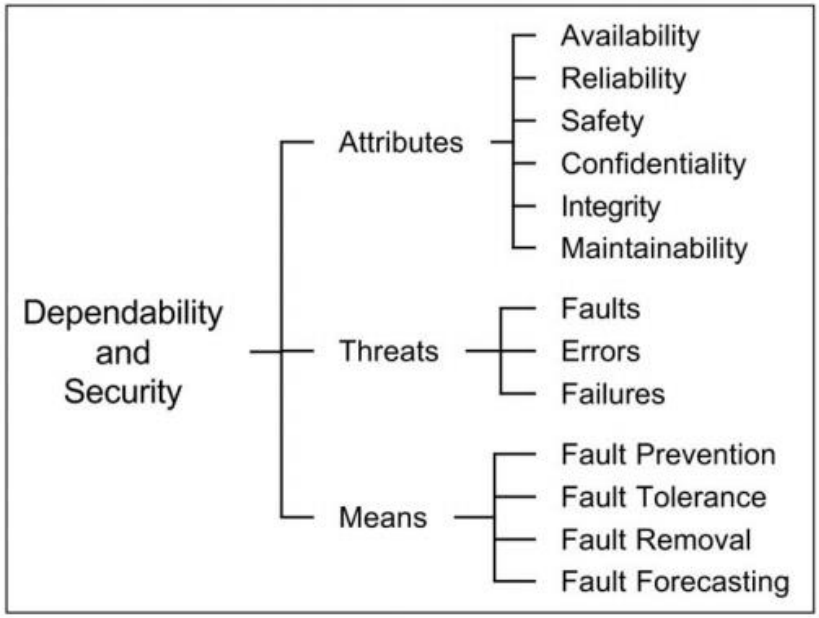


Figure 3 Attributes, threats and means to attain dependability and security

2.1.2 Threats to dependability and security

A system can experience various events that can affect its ability to deliver correct service. Therefore, those events pose a threat to the dependability and security aspect of the system. In the technical paper of Avizienis [7] following threats are defined:

- **Fault** - "the adjudged or hypothesized cause of an error" [7].
- **Error** - "the part of a system's total state that may lead to a service failure" [7].
- **Failure** - "the transition from correct service to incorrect service" [7].

Faults are vulnerabilities in a system that can have a negative impact on the systems state, possibly causing an error. If a fault is present and not causing an error, it is defined as a **dormant fault** [7]. For example, let us assume that we have a line of code, which contains a bug. As long as that line of code is not executed, the fault remains dormant. When the buggy code gets executed, it will activate the fault and hence trigger an **error**. This error can now propagate through other components, causing more and more errors on the way. In the end, the errors reach the service interface where they hamper the system's correct service and cause a **failure**.

This chain of events thus always starts with a fault, which leads to an error and ends with a failure. This is referred to as a **chain of threats** [7].

Figure 4 shows how faults, errors, and failures are linked, and how they can propagate throughout a system.

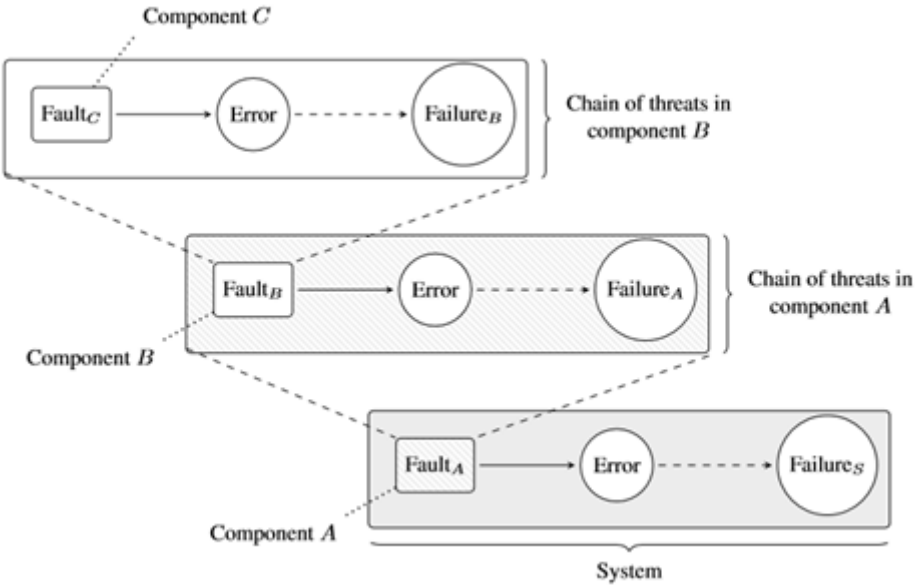


Figure 4 Propagation of the chain of threats throughout a system, leading to failure of the system

The figure illustrates a system that consists of multiple nested components. The component C experiences the chain of threats, which leads to the failure of this component. Since component C is part of component B, the failure of component C is registered as a fault in component B. This leads to the failure of component B, as well as component A, since B is part of component A. As component A is on system level, it leads to failure of the system.

2.1.3 Failure Modes

Failure modes represent the different possibilities in which failures can manifest themselves. They describe how the system is acting when a service outage occurs. For classifying the failure modes, dependability researchers use a scheme, which puts the failure modes into a nested hierarchy, as shown in Figure 5 [6].

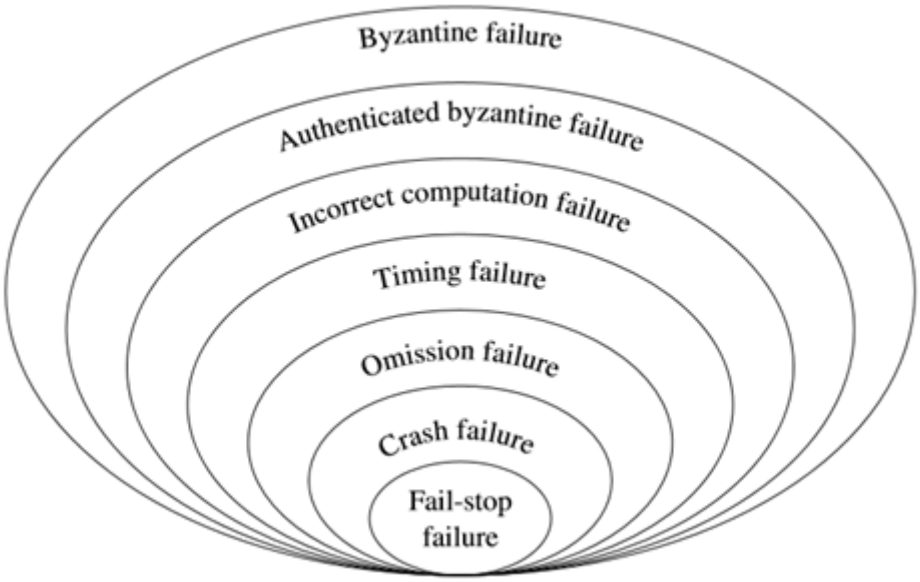


Figure 5 Classification of failure modes

Thereby, the innermost circle represents the failure mode, which is the easiest to deal with (fail-stop), while the outermost circle is the hardest one (byzantine failure). Making a system highly reliable would be very difficult if its components fail to exhibit a byzantine failure mode. Therefore, designers of highly reliable systems should strive for designing components with a fail-stop failure mode if possible and thus ensure high reliability of the system [6].

Byzantine failures are the hardest to deal with because there are no restrictions on how the service is deviating from correct service [6]. In message-passing systems, a byzantine failure could be expressed in a node, which sends messages with the wrong source address. Such behaviour is called **Impersonation** [6]. Furthermore, another scenario is the **two-faced behaviour**, where a node is supposed to send the same message to two recipients, but it transmits an altered message to one of them. Other expressions, which are used for byzantine failure modes, are arbitrary, fail-uncontrolled and malicious failure modes [6].

Authenticated byzantine failure mode is in principle the same as byzantine failure mode, with the exception that no impersonations are present [6]. The term authenticated comes from the fact that this failure mode has a mechanism for verifying the authenticity of messages, making it impossible that one node impersonates another. This mode is also called authentication detectable byzantine failure mode [6].

Incorrect computation failure mode occurs when incorrect service is delivered in form of value deviation, timing or both, but without any impersonations and two-faced behaviours [6]. Value deviation occurs when the message content sent by a node is incorrect. Regarding time deviation, there are two cases: timing deviation and omission. Thereby, timing deviation occurs when a message is sent too soon or too late by a node, while omission is the case when a node delays transmission of a message indefinitely.

Timing failure mode occurs when incorrect service is delivered in regards to the time domain, but the value domain stays intact. In distributed embedded systems there is a special timing failure called babbling-idiot failure mode, which occurs when a node is transmitting messages one after another, without stopping. This is problematic because it can stop the transmission of messages from other nodes. Another name for timing failures is performance failures [6].

Omission failure mode occurs when a node is delaying transmission of messages indefinitely.

Crash failure mode occurs when a node permanently stops the transmission of any message [6]. Other terms used for this failure mode are halt failures, silent failures or simply silence [6].

Fail-stop failure modes occur when a node is experiencing a crash failure, and other nodes are capable of detecting this. Other terms used for this failure mode are stopping failure and signalled failure.

2.1.4 Means to attain Dependability and Security

Faults, errors and failures are the dependability threats, which need to be tackled when building highly reliable systems. Therefore, for protecting a system from such threats, special means and techniques are developed and used by system designers [6]. They enable avoidance of service failures that are more frequent or severe than is acceptable [6]. Following means have been defined:

- Fault prevention - "means to prevent the occurrence or introduction of faults" [7]. This method aims to eliminate the chains of threats by preventing faults from occurring in the first place [6].
- Fault tolerance - "means to avoid service failures in the presence of faults" [7]. The aim of this method is to break the chain of threats and despite the presence of faults, disable the occurrence of the last phase - failure [6].
- Fault removal - "means to reduce the number and severity of faults" [7]. This method aims to identify possible fault-triggering components and removing or replacing them [6].
- Fault forecasting - "means to estimate the present number, the future incidence, and the likely consequences of faults" [7]. This method can be divided into qualitative fault forecasting and quantitative fault forecasting. The aim of qualitative fault forecasting is to identify in which way or what combination components have to fail to cause a system failure [6]. The aim of quantitative fault forecasting is to define the extent to which the dependability attributes are satisfied [6]. An example of quantitative fault forecasting is the reliability analysis of a system, in order to determine the probability that the system will not fail in a given time period [6].

2.1.5 Monitoring

To ensure that a system is fault-tolerant, we must deploy special techniques for the detection of faults in the system itself. For this purpose, monitoring can be utilized. Furthermore, monitoring is also very useful for fault forecasting, as it provides the possibility to learn the correct system behaviour and thereby forecast potential errors/faults that might be observable via historic deviations.

For monitoring of fault-tolerant systems, special requirements must be fulfilled by the chosen monitoring architecture. In “Monitoring Distributed Real-Time System - A survey and future directions" by Alwyn E. Goodloe and Lee Pike, following architectural constraints are proposed that need to be met:

- Functionality - The functionality of the system under observation must not be affected by the monitor, unless the system violates its specification [8].
- Schedulability - The hard real-time guarantees of the system must not be affected by the monitor architecture, unless the system violates its specification [8].

- Reliability - The reliability of the system under observation alone must not be smaller than the reliability of the system under observation in the context of the monitoring architecture [8].
- Certifiability - The source code of the system under observation must not be unduly modified by the monitor architecture [8].

These constraints ensure that the monitor benefits the system under observation and has no impact on the nominal functionality of the system, unless the system is detected to violate its specification [8].

2.2 Dependable System Engineering

Many tools and methods can support dependable Systems Engineering. In general, it is necessary to manage several elements to achieve the goal of a dependable system. Starting with the requirements to a system specification, to testing and operations. A commonly used industrial process is the V-Model. Within the V-Model, it is possible to map each phase of a product to a step inside a typical development situation, as shown in Figure 6.

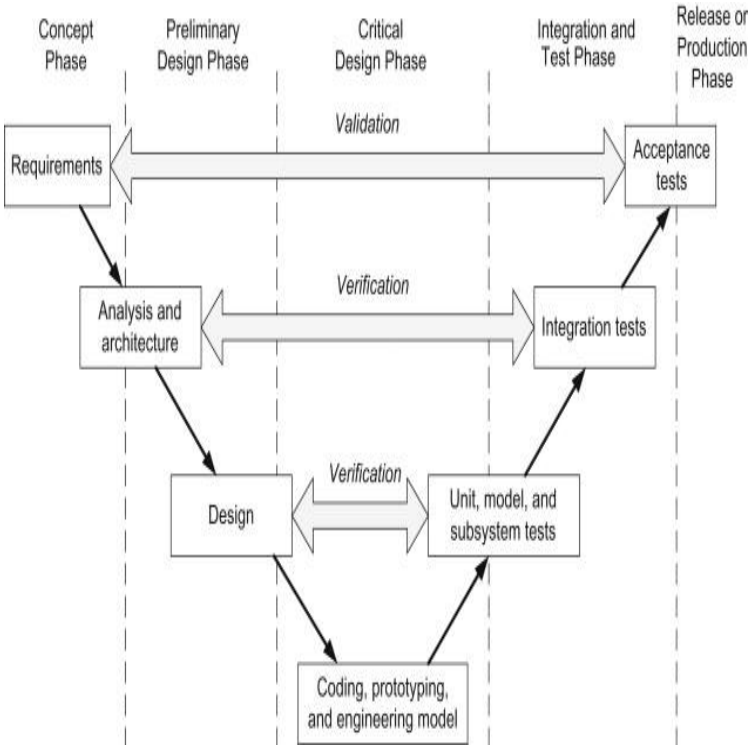


Figure 6 Depiction of V-model development process landscape[2]

The V-model is also used to create dependable systems. One approach for dependable System Development is Model-Based Systems Engineering (MBSE). It is a state-of-the-art method to achieve traceability throughout the whole development process. This traceability is required for the Software Process Improvement and Capability Determination (SPICE / ISO/IEC 15504) and for automotive applications in Automotive SPICE. For proper Model-Based Systems Engineering, it may be necessary to use several models for developing a single system. Some of the models can be created in standardized languages such as UML or SysML. Those

[2] https://www.sciencedirect.com/topics/engineering/v-model

languages are often used for architecture, component design, and higher-level behavioural models, but are not limited to them. Requirements are often modelled in specifically designed tools or even written in domain-specific languages like the Requirements Interchange Format (ReqIF). Such domain-specific languages can be used to create tests from behavioural models of the system and are used to verify and validate the systems under development. When using a model-based approach, the important thing is to guarantee that all data is correctly linked across all involved models throughout the development. Additionally fleet data from the in-use phase should be linked back into the models to verify the models and increase their quality for reuse in the next project. This linkage enables traceability, hence verifiability, and can be used as a safety argument to prove the system's dependability.

2.3 Dependability Engineering Methods

To achieve dependable system engineering several methods have proven to be efficient in the past years. We will focus on one model, the V-model, in particular since it supports dependable systems engineering. The V-model is a graphical representation of a systems development lifecycle and is used in many industries.

As shortly introduced in Chapter 2.2 the V-Model can be mapped to all stages of a product development cycle. The automotive industry uses for example ASPICE, which incorporates the V-Model to map not only different phases of a product development cycle but also different disciplines. It includes the 'Acquisition Process' as well as 'Supply Processes', 'System and Software Engineering' and organizes 'Management Processes' enforces the 'Reuse Process' encourages a 'Process Improvement Process' and is valid beyond the production of a product until the end of life using 'Supporting Processes' [9].

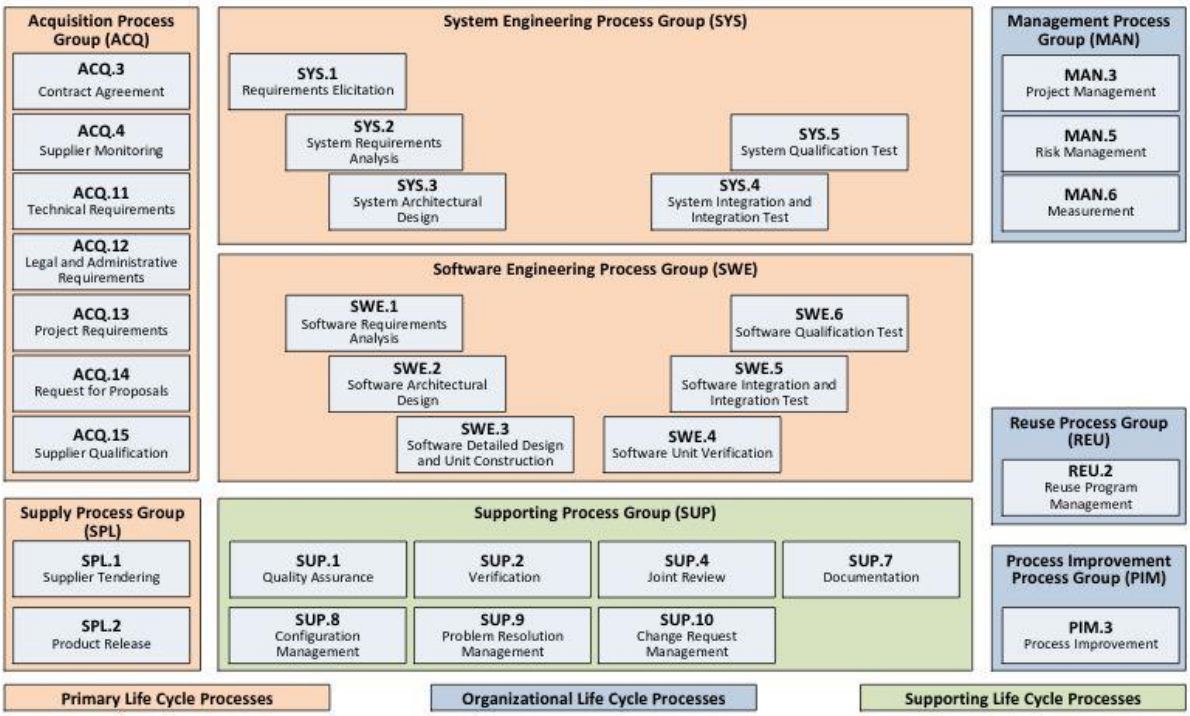


Figure 7 Automotive SPICE Process Reference Model from **[9]**

As the colour coding in the figure above shows, there are in fact several V-Models stacked over each other. All of those processes run in parallel - though some might not have active tasks at

all time throughout the process. The following graphic illustrates several processes that are all run in parallel and following the V-Model.

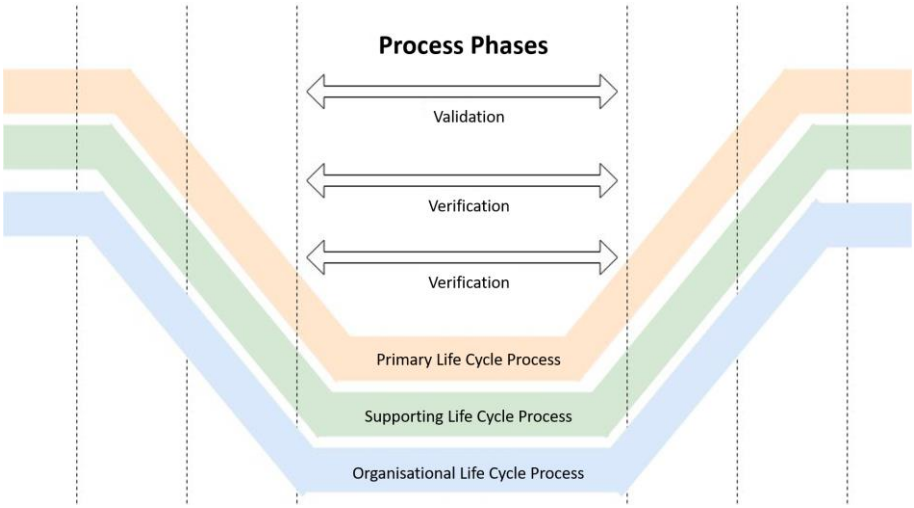


Figure 8 Depiction of the parallel processes in Automotive SPICE

Whereas in reality more often a Horizontal Top-Level Process is used to coordinate several V-Model processes.

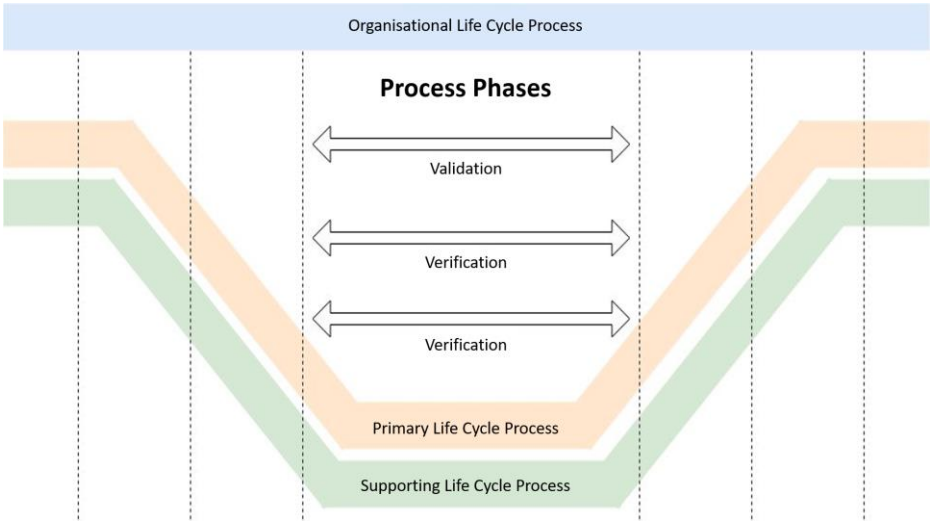


Figure 9 Depiction of the top-level process coordinating the parallel V-Model processes

The flexibility of this V-Model Process enables extensions to be developed. For example the Safety extension of ISO/IEC 15504 Part 10. To enable functional safety there are three main processes defined namely Safety Management Process, Safety Engineering Process and a Safety Qualification Process. ISO 15504 claims to be compatible with IEC 61508 and ISO 26262 which are the two standards for industrial safety and automotive safety.

ISO/IEC 15504-10 Covers the Safety relevant aspects and introduces frameworks and methods for dependable systems engineering. Another part of dependable systems engineering is security. Therefore, the automotive industry is changing and adapted ISO 26262 with security. This extension is compatible with ISO 15504 and covers topics of ISO 27000, which is the IT standard for Security.

2.4 Regulations and Domain Activities

The main regulations that we included in this section are related to Automated Driving. EU countries must comply with the regulations that are provided by UNECE, the **United Nations Economic Commission for Europe ("UNECE"),** which confers the legal basis to establish uniform type approval regulations. The UNECE Regulations contain provisions for:

1. Administrative Procedures for granting type approvals
2. Performance-oriented test requirements
3. Conformity of production ("CoP")
4. Mutual Recognition of Type Approvals

The regulations related to the automotive sector, come from UNECE World Forum for Harmonization of Vehicle Regulations (WP.29).

Contracting Parties (member countries) who signed the Agreement of 1958, 1997, 1998, participate in the WP.29 sessions to establish regulatory instruments concerning motor vehicles and their equipment. The regulations that they can provide are:

- UN Regulations: provisions (for vehicles, their systems, parts and equipment) related to safety and environmental aspects. They include performance-oriented test requirements, as well as administrative procedures. The latter address the type approval (of vehicle systems, parts and equipment), the conformity of production (i.e. the means to prove the ability, for manufacturers, to produce a series of products that exactly match the type approval specifications) and the mutual recognition of the type approvals granted by Contracting Parties.
- UN GTRs: globally harmonized performance-related requirements and test procedures. They provide a predictable regulatory framework for the global automotive industry, consumers and their associations. They do not contain administrative provisions for type approvals and their mutual recognition.
- UN Rules: periodical technical inspections of vehicles in use. Contracting Parties reciprocally recognize (with certain conditions) the international inspection certificates granted according to the UN Rules.

WP.29 has six permanent working Parties (GR), which consider specialized tasks, and among them, we find the GRVA (Working Part about Autonomous Vehicle). Then there are also some Informal Working Groups with a time-limited mandate to deal with certain technical issues.

In general, proposals to WP.29 for new regulatory instruments, such as UN Regulations, UN GTRs and UN Rules (or the amendment of existing ones), are elaborated by the Contracting Parties to one of the UN Agreements administered by WP.29 in the informal working groups (and their subgroups), then discussed by the bodies they report to (typically by the GRs during one of their sessions, and sometimes by WP.29 directly). Finally, the proposals are considered during WP.29 sessions (by the relevant Committee, depending on the UN Agreement concerned) for their final approval.

Once the Regulation has been approved, each Contracting Part must sign it and then prepare a national Law that refers to the UNECE Regulation, to make it mandatory in its Nation.

This would allow all Contracting Parties to refer to the same Regulation, and to automatically validate the Mutual Recognition of Type Approvals.

For Contracting Parties inside the EU, it is the EU Commission that defines the timeline of application of the UNECE Regulation, and the Contracting Parties must respect this deadline.

We collected here the main regulations in a chronological order, starting from the newest ones.

2.4.1 Regulations for AV

UNECE R 155 (2021- the final phase of approval) - ***Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system***

The regulation applies to passenger cars, vans, trucks and buses, light four-wheeler vehicles if equipped with automated driving functionalities from level 3 onwards – this covers the new automated pods, shuttles etc.; trailers if fitted with at least one electronic control unit.

The UN Regulation provides a framework for the automotive sector to put in place the necessary processes to:

- Identify and manage cyber security risks in vehicle design;
- Verify that the risks are managed, including testing;
- Ensure that risk assessments are kept current;
- Monitor cyber-attacks and effectively respond to them;
- Support analysis of successful or attempted attacks;
- Assess if cyber security measures remain effective in light of new threats and vulnerabilities.

All of these will be audited by national technical services or homologation authorities.

The type approval principles under the 1958 Agreement mean that manufacturers will need to demonstrate, prior to putting vehicles on the market, that they fulfil the following requirements:

- Cyber Security Management System is in place and its application to vehicles on the road is available;
- Provide risk assessment analysis, identify what is critical;
- Mitigation measures to reduce risks are identified;
- Evidence, through testing, that mitigation measures work as intended;
- Measures to detect and prevent cyber-attacks are in place;
- Measures to support data forensics are in place;
- Monitor activities specific for the vehicle type;
- Reports of monitoring activities will be transmitted to the relevant homologation authority.

UNECE R 156 *(2021 – under development)* **- *Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to software update and software update management system***

The UN Regulation applies to vehicles permitting software updates of passenger cars, vans, trucks and buses; trailers; agricultural vehicles.

The UN Regulation provides a framework for the automotive sector to put in place the necessary processes for:

- Recording the hardware and software versions relevant to a vehicle type;
- Identifying software relevant for type approval;
- Verifying that the software on a component is what it should be;
- Identifying interdependencies, especially with regards to software updates;
- Identifying vehicle targets and verifying their compatibility with an update;

- Assessing if a software update affects the type approval or legally defined parameters (including adding or removing a function);
- Assessing if an update affects safety or safe driving;
- Informing vehicle owners of updates;
- Documenting all the above.

All of these will be audited by national technical services or homologation authorities.

The type approval principles under the 1958 Agreement mean that manufacturers will need to demonstrate, prior to putting vehicles on the market, that they fulfil the following requirements:

- Software Update Management System is in place and its application to vehicles on the road is available;
- Protect SU delivery mechanism and ensure integrity and authenticity;
- Software identification numbers must be protected;
- Software identification number is readable from the vehicle;
- For Over-The-Air software updates:
- Restore function if update fails;
- Execute update only if sufficient power;
- Ensure safe execution;
- Inform users about each update and about their completion;
- Ensure vehicle is capable of conducting update;
- Inform user when a mechanic is needed.

UNECE WP.29 GRVA *– (2020- not frozen)* ***Proposal for a new UN Regulation on Event Data Recorder***

This Regulation applies to the approval of vehicles of categories M1 and N13 with regard to their Event Data Recorder (EDR).

The regulation provides rules for the type approval of vehicles equipped with EDR, including collection, storage and crash survivability of motor vehicle crash data.

It is not applicable to retro-fitted or aftermarket systems.

The regulation applies to systems and sensors already present in the vehicle and active, and to data, they are producing.

The regulation sets up the requirements about data to be recorded, events to be recorded, locking conditions, overwriting, power failure, crash test performance and survivability.

UNECE R 157 *(2020)* ***- Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to Automated Lane Keeping System***

The UN Regulation establishes strict requirements for Automated Lane Keeping Systems (ALKS) for passenger cars, which, once activated, are in primary control of the vehicle. However, the driver can override such systems and can be requested by the system to intervene, at any moment.

This is the first binding international regulation on so-called "level 3" vehicle automation. The new Regulation therefore marks an important step towards the wider deployment of automated vehicles to help realize a vision of safer, more sustainable mobility for all. It will enter into force in January 2021.

ALKS can be activated under certain conditions on roads where pedestrians and cyclists are prohibited and which, by design, are equipped with a physical separation that divides the traffic moving in opposite directions. In its current form, the Regulation limits the operational speed of ALKS systems to a maximum of 60 km/h.

The European Commission has announced that the Regulation will apply in the European Union following its entry into force.

The Regulation requires that on-board displays used by the driver for activities other than driving when the ALKS is activated shall be automatically suspended as soon as the system issues a transition demand, for instance in advance of the end of an authorized road section. The Regulation also lays down requirements on how the driving task shall be safely handed back from the ALKS to the driver, including the capability for the vehicle to come to a stop in case the driver does not reply appropriately.

UNECE R 152** (2019) - **Uniform provisions concerning the approval of motor vehicles with regards to the Advanced Emergency Braking System (AEBS) for M1 and N1 vehicles

The UN Regulation will lay down the technical requirements for the approval of "vehicle-to-vehicle" and "vehicle-to-pedestrian" AEBS fitted on cars. Such systems employ sensors to monitor the proximity of the vehicle or pedestrian in front and detect situations where the relative speed and distance between the two vehicles or between the vehicle and pedestrian suggest that a collision is imminent. In such a situation, if the driver does not react to the system's warning alerts, emergency braking will be automatically applied to avoid the collision or at least to mitigate its effects.

There were no standard technical requirements guaranteeing the effective performance of such systems so far.

The new UN Regulation will impose strict and internationally harmonized requirements for the use of AEBS at low speeds, even in complex and unpredictable situations such as traffic in urban areas.

The Regulation sets out test requirements for the deployment of AEBS at a range of different speeds, from 0-60 km/h. In addition to cars, the Regulation will be applicable to all light commercial vehicles (vans and minibuses with less than 9 passengers). With this Regulation in Force, most existing systems will have to be updated to meet stricter requirements.

The draft Regulation was approved by the Working Party on Automated/Autonomous and Connected Vehicles (GRVA) under UNECE's World Forum for Harmonization of Vehicle Regulations (WP.29). The new Regulation would enter into force in early 2020.

The European Union and Japan, who together led the development of the Regulation, have announced that AEBS systems would then become mandatory for all new cars and light commercial vehicles (from 2022 in the EU).

EU Commission** (2018) - **Guidelines on the exemption procedure for the EU approval of Automated Vehicles.

Valid for L3 and L4 vehicles, series vehicles. Objective: harmonize in EU the national ad-hoc assessment for automated vehicles, to reach a mutual recognition of such assessment.

8 safety harmonized requirements should be considered during the assessment:

- system performance in the automated mode (definition of the OD)
- driver /operator/passenger interaction (capability to inform the human driver of the situation, when it is necessary to take the control of the vehicle)
- transition of the driving tasks
- minimum risk maneuver (MRM)
- installation of Data Storage Systems
- Cybersecurity
- Safety assessment and tests
- Information provision to AV users.

2.4.2 Standards for AV

Standards represent the worldwide scientific state of the art about a topic. They are not mandatory by law, but they represent the best way to design/develop a product.

The following standards have been ordered chronologically, starting from the newest.

They are specific for Autonomous vehicle design /development/testing, for specific functionalities related to the Autonomous vehicle, and to safety and cybersecurity functions.

ISO 23374 ITS - Automated valet parking systems (AVPS) — System framework, communication interface, and vehicle operation

ISO DTR 4804 – Road Vehicles. Safety and Cyber security for Automated Driving Systems-Design, Verification and Validation – Recommendations and guidance of the steps for developing and validating automated driving systems based on basic safety principles derived from worldwide applicable publications (various legal frameworks from around the world, ethics reports, etc.). These principles provide a foundation for deriving a baseline for the overall safety requirements and activities necessary for the different automated driving functions including human factors as well as the verification and validation methods for automated driving systems focused on vehicles with level 3 and level 4 features according to SAE J3016:2018.

ISO SAE DIS 21434 - Road vehicles — Cybersecurity engineering

Guideline for the organization management of cybersecurity (CSMS), and for the operative cybersecurity activities to be performed for automotive product development

ISO/WD PAS_5112 Road vehicles -- Guidelines for auditing cybersecurity engineering

Guideline to perform cybersecurity audit to a Company, to evaluate the compliance to CSMS defined in the UN ECE Reg 155

VDA - Automotive Cyber Security Management System Audit

This document provides the questionnaire and a rating scheme, to perform a cybersecurity audit covering the expected requirements of the UNECE R155 about CSMS.

ISO/DIS 22737 ITS- Low-speed automated driving (LSAD) systems for predefined routes - Performance requirements, system requirements and performance test procedures

***ISO 21202 PALS Partially Automated Lane Change System - Functional / operational requirements and test procedures** (2020)*

PALS perform part or all of lane change tasks under the driver's initiation and supervision. PALS are intended to function on roads with visible lane markings, where non-motorized vehicles and pedestrians are prohibited.

***ISO 20900 ITS- Partially automated parking systems (PAPS) - Performance requirements and test procedures** (2019)*

The document addresses light vehicles, e.g. passenger cars, pick-up trucks, light vans and sport utility vehicles (motorcycles excluded), equipped with partially automated parking systems (PAPS). This document establishes minimum functionality requirements that the driver can expect and the manufacturer needs to take into account. Possible system configuration includes the following two types:

- Type 1: System supervised by the conventional driver located in the driver's seat;
- Type 2: System supervised by the remote driver (present within or outside the vehicle) that is not necessarily located in the driver's seat. The vehicle remains in the line of sight of the remote driver.

For both types, minimum requirements and conditions of safety, system performance and function including HMI information content and description of system operating states are addressed. The requirements include the driver who supervises the safety throughout the system maneuvers. System test requirements are also addressed including test criteria, method, and conditions.

VDA - Standardization Roadmap for Automated Driving (2019)

Identifies the main standardization bodies for automotive engineering, electronics engineering and information technology. The three worlds are connected because, starting from assisted to automated driving, requirements for in-vehicle communication with regards to higher bandwidth and lower latency become critical, as well as error free data transmission.

ISO PAS 21448 - Road Vehicles - Safety of the Intended Functionalities (SOTIF) (2019)

Guidance on the applicable design, verification and validation measures needed to achieve the SOTIF. The absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable misuse by persons is referred to as the Safety Of The Intended Functionality (SOTIF). ISO PAS 21448 does not apply to faults covered by the ISO 26262 series because it does not address potential risks that arise from malfunctions of the safety-related E/E system. ISO PAS 21448 is intended to be applied to intended functionality where proper situational awareness is critical to safety, and where that situational awareness is derived from complex sensors and processing algorithms; especially emergency intervention systems and systems with levels of automation 1 to 5 on the OICA / SAE standard J3016automation scales.

ISO 20035 ITS- Cooperative Adaptive Cruise Control System (CACC) Performance requirements and test procedures (2019)

Cooperative Adaptive Cruise Control (CACC) system is an expansion to existing Adaptive Cruise Control (ACC) control strategy by using wireless communication with preceding vehicles (V2V) and/or the infrastructure (I2V). Both multi vehicle V2V data and I2V infrastructure data are within the scope of this document. When V2V data is used CACC can

enable shorter time gaps and more accurate gap control, which can help increase traffic throughput and reduce fuel consumption. It can also receive data from the infrastructure, such as recommended speed and time gap setting, to improve traffic flow and safety.

This document addresses two types of Cooperative Adaptive Cruise Control (CACC): V2V, and I2V. Both types of CACC system require active sensing using for example radar, LIDAR, or camera systems. The combined V2V and I2V CACC is not addressed in this document. The following requirements are addressed in this document:

- classification of the types of CACC;
- definition of the performance requirements for each CACC type;
- CACC state transitions diagram;
- the minimum set of wireless data requirements;
- test procedures.

CACC:

- does only longitudinal vehicle speed control;
- uses time gap control strategy like ACC;
- has similar engagement criteria as ACC.

Coordinated strategies to control groups of vehicles, such as platooning, in which vehicle controllers base their control actions on how they affect other vehicles and may have a very short following clearance gap are not within the scope of this document. CACC system operates under driver responsibility and supervision. This document is applicable to motor vehicles including light vehicles and heavy vehicles.

ISO 26262 – Road Vehicles – Functional Safety (2018)

This standard is intended to be applied to safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production road vehicles, excluding mopeds. This document does not address unique E/E systems in special vehicles such as E/E systems designed for drivers with disabilities.

This document addresses alterations to existing systems and their components released for production prior to the publication of this document by tailoring the safety lifecycle depending on the alteration. This document addresses the integration of existing systems not developed according to this document and systems developed according to this document by tailoring the safety lifecycle.

This document addresses possible hazards caused by malfunctioning behaviour of safety-related E/E systems, including the interaction of these systems. It does not address hazards related to electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy and similar hazards, unless directly caused by malfunctioning behaviour of safety-related E/E systems.

This document describes a framework for functional safety to assist the development of safety-related E/E systems. This framework is intended to be used to integrate functional safety activities into a company-specific development framework. Some requirements have a clear technical focus to implement functional safety into a product; others address the development process and can therefore be seen as process requirements in order to demonstrate the capability of an organization with respect to functional safety.

SAE J3164 - Taxonomy and definitions for Terms Related to Automated Driving System Behaviours and Maneuvers for On Road Motor Vehicles (2018)

This document provides definitions, taxonomies, and best practices for behaviours and maneuvers of on-road automated driving systems (ADSs) for automation levels 3 ("Conditional Automation"), 4 ("High Automation"), and 5 ("Full Automation").

SAE J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (2018)

This SAE Recommended Practice describes motor vehicle driving automation systems that perform part or all of the dynamic driving task (DDT) on a sustained basis.

ISO 21717 PADS Partially automated in-lane driving system - Performance requirements and test procedures (2018)

Basic control strategy, minimum functionality requirements, basic driver interface elements, minimum requirements for diagnostics and reaction to failure, and performance test procedures for Partially Automated In-Lane Driving Systems (PADS).

ISO 15622 ACC- Performance requirements and test procedures (2018)

Basic control strategy, minimum functionality requirements, basic driver interface elements, minimum requirements for diagnostics and reaction to failure, and performance test procedures for Adaptive Cruise Control (ACC) systems.

ACC systems are realised as either Full Speed Range Adaptive Cruise Control (FSRA) systems or Limited Speed Range Adaptive Cruise Control (LSRA) systems. LSRA systems are further distinguished into two types, requiring manual or automatic clutch. Adaptive Cruise Control is fundamentally intended to provide longitudinal control of equipped vehicles while travelling on highways (roads where non-motorized vehicles and pedestrians are prohibited) under free-flowing and for FSRA-type systems also for congested traffic conditions. ACC can be augmented with other capabilities, such as forward obstacle warning. For FSRA-type systems, the system will attempt to stop behind an already tracked vehicle within its limited deceleration capabilities and will be able to start again after the driver has input a request to the system to resume the journey from a standstill. The system is not required to react to stationary or slow-moving objects.

ISO 19237 PDCMS Pedestrian protection - Performance requirements and test procedures (2017)

Operation, minimum functionality, system requirements, system interfaces, and test procedures for Pedestrian Detection and Collision Mitigation Systems (PDCMS)

SAE J3131 - Automated Driving Reference Architecture (WIP - 2016)

SAE J3131 defines an automated driving reference architecture that contains functional modules supporting future application interfaces for Levels 3 - 5. The architecture will model scenario-driven functional and non-functional requirements, automated driving applications, functional decomposition of an automated driving system, and relevant functional domains (i.e., functional groupings).

SAE J3114 - Human Factors Definitions for Automated Driving and Related Research Topics (2016)

The aim of this Information Report is to provide terms and definitions that are important for the user's interaction with L2 through L4 driving automation system features.

SAE J3061 - Cybersecurity Guidebook for Cyber Physical Vehicle Systems (2016)

The first guideline for automotive cybersecurity, showing the approach to the cybersecurity for E/E devices, the interaction with the Functional Safety, and the management of all cybersecurity activities on the product.

SAE J3092 - Dynamic Test Procedures for V&V of ADT (2015)

This document provides dynamic test procedure information and guidelines for verification and validation (V&V) of automated driving systems (ADSs). The levels of automation addressed in this document include level 3, 4, 5.

SAE J3018 - Safety-Relevant Guidance for On-Road Testing of SAE Level 3, 4, and 5 Prototype Automated Driving System (ADS)-Operated Vehicles (2015)

This document provides guidelines for the safe conduct of on-road tests of vehicles equipped with prototype conditional, high, and full (levels 3-5) automated driving systems (ADSs), as defined by SAE J3016

IEC 62508 - Guidance on human aspects of dependability (2010)

Guidance on the human aspects of dependability, and the human-centred design methods and practices that can be used throughout the whole system life cycle to improve dependability performance

2.4.3 Regulations, Standards and Guidelines for AI

UNECE WP.29 released a first Informal document, WP.29-175-21, about the ***Artificial intelligence and vehicle regulation***, where it connects AI to two specific applications of automotive sector:

- HMI enhancements for infotainment and vehicle management
- Development of self-driving (building of HD maps, surrounding detection using sensor data fused with Deep Learning algorithms, driving policies for automated driving using Deep Learning)

The impact of AI on vehicle driving is also studied in the Informal Working Groups for:

- HMI distraction
- Performance of automated vehicle.

Now there is not jet a Regulation specific for AI from the UNECE.

Nevertheless, in the last two years, the European Commission has been very active in the study of AI and its impact on citizens' lives. European Commission created an independent Group of High Level Experts for Artificial Intelligence, who realised a set of guidelines for AI.

Ethics Guidelines for Trustworthy AI, 2019 - *EU Independent High Level Expert Group on Artificial Intelligence*

The aim of the Guideline is to highlight that AI systems need to be *human-centric*, resting on a commitment to their use in the service of humanity and the common good, with the goal of improving human welfare and freedom.

So, a Trustworthy AI should be:

- lawful, complying with all applicable laws and regulations;
- ethical, ensuring adherence to ethical principles and values;
- robust, both from a technical and social perspective

The Guidelines set up a framework to achieve the Trustworthy AI, working on the *ethical* and *robust* principle cited above. These two principles, guided by the framework, become an operational activity to be followed in the development of AI systems.

The Guideline starts from the Foundation of Trustworthy AI: the fundamental rights described in EU Treaties and EU Charted and international Human rights law, are re-written for the AI development:

- **respect for human dignity:** Human dignity encompasses the idea that every human being possesses an "intrinsic worth", which should never be diminished, compromised or repressed by others – nor by new technologies like AI systems. AI systems should hence be developed in a manner that respects, serves and protects humans' physical and mental integrity, personal and cultural sense of identity, and satisfaction of their essential needs.
- **freedom of the individual, human** beings should remain free to make life decisions for themselves. In an AI context, freedom of the individual, for instance, requires mitigation of (in) direct illegitimate coercion, threats to mental autonomy and mental health, unjustified surveillance, deception and unfair manipulation.
- **respect for democracy**, justice and the rules of the law all governmental power in constitutional democracies must be legally authorized and limited by law. AI systems must also embed a commitment to ensure that they do not operate in ways that undermine the foundational commitments upon which the rule of law is founded, mandatory laws and regulation, and to ensure due process and equality before the law.
- **equality, non-discrimination and solidarity** *including the rights of persons at risk of exclusion.* Equal respect for the moral worth and dignity of all human beings must be ensured. This goes beyond non-discrimination. In an AI context, equality entails that the system's operations cannot generate unfairly biased outputs.

From the fundamental rights, the four ethic principles for AI must be:

1. ***Respect for human autonomy:*** Humans interacting with AI systems must be able to keep full and effective self-determination over themselves and be able to partake in the democratic process. AI systems should not unjustifiably subordinate, coerce, deceive, manipulate, condition or herd humans. Instead, they should be designed to augment, complement and empower human cognitive, social and cultural skills. The allocation of functions between humans and AI systems should follow human-centric design principles and leave meaningful opportunity for human choice. This means securing human oversight over work processes in AI systems.
2. ***Prevention of harm:*** AI systems should neither cause nor exacerbate harm or otherwise adversely affect human beings. This entails the protection of human dignity as well as mental and physical integrity. AI systems and the environments in which they operate must be safe and secure. They must be technically robust, and it should be ensured that they are not open to malicious use. Vulnerable persons should receive greater attention and be included in the development, deployment and use of AI systems.
3. ***Fairness:*** The development, deployment and use of AI systems must be fair. Ensuring an equal and just distribution of both benefits and costs, and ensuring that individuals and groups are free from unfair bias, discrimination and stigmatization. If unfair biases can be avoided, AI systems could even increase societal fairness. Equal opportunity in terms of access to education, goods, services and technology should also be fostered.
4. ***Explicability:*** Processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected.

The Guideline transforms then the four ethical principles in seven ethical requirements for AI:

1. **Human agency and oversight**

AI systems should support human autonomy and decision-making, as prescribed by the principle of *respect for human autonomy*. Users should be able to make informed autonomous decisions regarding AI systems. They should be given the knowledge and tools to comprehend and interact with AI systems to a satisfactory degree. Oversight may be achieved through governance mechanisms such as a human-in-the-loop (HITL), human-on-the-loop (HOTL), or human-in-command (HIC) approach.

2. **Technical robustness and safety –** linked to the principle of *prevention of harm*

AI systems should be robust to attack and security (attack to data, to model, to the infrastructure, SW and HW).

AI Systems should have safeguards that enable fall back plan in case of problems and for general safety (ask the driver to keep the control, or go from a statistical to a rule-based procedure).

AI systems shall be accurate - make correct judgments, correct predictions, decisions, based on data or models.

Results of AI systems shall be reproducible, as well as reliable (a reliable AI system works properly with a range of inputs and in a range of situations. Reproducibility describes whether an AI experiment exhibits the same behaviour when repeated under the same conditions.

3. **Privacy and data governance** – *linked to the principle of prevention of harm*

AI systems must guarantee privacy and data protection throughout a system's entire lifecycle (information initially provided by the user; information generated about the user over the course of their interaction with the system).

It must be ensured that data collected about users will not be used to unlawfully or unfairly discriminate against them.

Processes and data sets used must be tested and documented at each step such as planning, training, testing and deployment.

Data protocols governing data access should be put in place.

4. **Transparency** – *linked with the principle of Explicability*

Traceability - the data sets and the processes that yield the AI system's decision should be documented to the best possible standard to allow traceability and increase transparency.

Explainability - the decisions made by an AI system can be understood and traced by human beings

Communication - humans have the right to be informed that they are interacting with an AI system. The AI system's capabilities shall be communicated to the human user.

5. **Diversity, non-discrimination and fairness** - *linked to the principle of fairness*

Avoidance of unfair bias in the data collection and in the algorithm programming

Accessibility and universal design - systems should be user-centric and designed in a way that allows all people to use AI products or services, regardless of their age, gender, abilities or characteristics),

Stakeholder participation - stakeholders who may directly or indirectly be affected by the system throughout its life cycle shall be consulted during the design and development of the AI system.

6. **Societal and environmental wellbeing** – *linked to the principle of fairness and prevention of harm*

Sustainability and environmental friendly AI – a critical examination of the resource usage and energy consumption during training, opting for less harmful choices.

Social impact – the effect of AI on the society must be monitored

Society and democracy – the effects of AI systems on institutions, democracy and society at large must be monitored.

7. **Accountability** - *linked to the principle of fairness*

Auditability - enablement of the assessment of algorithms, data and design processes,

Minimization and reporting of negative impact,

Addressing the trade-offs in a rational way, between auditability and minimization

The application of the ethical requirements to an AI development can be done both with technical and non-technical methods.

Technical methods include:

- Architectures for Trustworthy AI, where the ethical requirements become specific procedure or constraints to be applied in the architecture definition of an AI system, with some black lists of restrictions and behaviour that the AI should never transgress, and the monitoring of the behaviour must be achieved with a dedicated process.
- Ethics and rule of law by design, which provide explicit links between abstract principles that the AI system must respect and specific implementation decisions. Some of these methods already used are privacy by design and security by design.
- Explainable AI (XAI), which are AI system that should explain why they are behaving in a certain why providing an interpretation. This kind of XAI is very complex to develop.
- Testing and validation of AI networks are fundamental to understand if the training and deployment of the AI system conducted to a reliable, stable and robust system, which behaves as expected also after its deployment. The testing process should be done from different groups, to have higher kinds of representation, adversarial testing teams should be available to try to break the system and find vulnerabilities.
- Quality of service indications shall be defined for AI system to ensure that there is a baseline understanding as to whether they have been tested and developed with security and safety considerations in mind.

Non-technical methods include:

- Regulations, that support AI trustworthiness, like product safety legislation, data protection legislation, etc.
- Codes of conduct, that are present in all stakeholders companies, should present a connection with the AI implementation done from the stakeholder
- Standardization works as a quality management system for who products / uses the AI system. Specific standards for trustworthy AI are still not available, while many reference standards like the ones related to safety, technical robustness, must be applied.
- Certification, complemented by Accountability of governance frameworks, will guarantee the respect of the ethical requirements from AI systems developed inside an Organization.

The guideline concludes with a preliminary proposal of assessment for AI systems.

Whitepaper on Artificial Intelligence *- European Commission - (2020)*

The goal of the European Commission is to create a unique "**ecosystem of trust**", based on:

- Ethics Guidelines for Trustworthy AI
- Existing EU legislation on fundamental rights (data protection, privacy, non-discrimination, consumer protection, product safety and liability rules)
- Human-centric approach (Consumers expect the same safety level with or without AI).

A regulatory framework should concentrate on how to minimize the various risks of potential harm, in particular the most significant ones, among the following:

- *material* harm (**safety** and **health** of individuals, including loss of life, damage to property)
- *immaterial* harm (**loss of privacy**, limitations to the right of freedom of expression, human dignity, discrimination for instance in access to employment)

The main risks related to the use of AI concern the application of rules designed to protect fundamental rights (including personal data and privacy protection and non-discrimination), as well as safety and liability-related issues.

The breaches of fundamental rights due to the use of AI are as an example coming from:

- flaws in the overall design of AI systems (including human oversight)
- the use of data without correcting possible bias.

The complexity and the opacity of an AI system may make it hard to verify compliance with rules of existing EU law meant to protect fundamental rights.

AI technologies may present new safety risks for users when they are embedded in products and services. For example, as a result of a flaw in the object recognition technology, an autonomous car can wrongly identify an object on the road and cause an accident involving injuries and material damage.

A lack of clear safety provisions tackling these risks may, in addition to risks for the individuals concerned, create legal uncertainty for businesses that are marketing their products involving AI in the EU.

Market surveillance and enforcement authorities may find themselves in a situation where they are unclear as to whether they can intervene, because they may not be empowered to act and/or don't have the appropriate technical capabilities for inspecting systems.

The difficulty of tracing back potentially problematic decisions taken by AI systems and referred to above in relation to fundamental rights applies equally to safety and liability-related issues.

The existing body of EU product safety and liability legislation, including sector-specific rules, further complemented by national legislation, is relevant and potentially applicable to several emerging AI applications:

- General Product Safety Directive (Directive 2001/95/EC)
- Race Equality Directive (Directive 2000/43/EC)
- Directive on equal treatment in employment and occupation (Directive 2000/78/EC)
- Directives on equal treatment between men and women in relation to employment and access to goods and services (Directive 2004/113/EC; Directive 2006/54/EC)
- Unfair Commercial Practices Directive (Directive 2005/29/EC) and the Consumer Rights Directive (Directive 2011/83/EC)
- General Data Protection Regulation
- Data Protection Law Enforcement Directive (Directive EU 2016/680)
- from 2025, European Accessibility Act will apply (Directive (EU) 2019/882)

The **EU new regulatory framework** would apply to products and services relying on AI.

The new regulatory framework for AI should be effective to achieve its objectives while not being excessively prescriptive so that it could create a disproportionate burden, especially for SMEs.

To strike this balance, the regulatory framework will be defined following a risk-based approach.

A risk-based approach requires clear criteria to differentiate between the different AI applications, in relation to the question of whether they are 'high-risk'.

A given AI application should generally be considered **high-risk** considering whether both the sector and the intended use involve significant risks, from the viewpoint of protection of safety, consumer rights and fundamental rights.

A given AI application will be considered high-risk when it meets the following two cumulative criteria:

1. the AI application is employed in a sector where significant risks can be expected to occur. The sectors covered should be specifically and exhaustively listed in the new regulatory framework and reviewed periodically. For instance, healthcare; transport; energy and parts of the public sector.
2. the AI application in the sector in question is, in addition, used in such a manner that significant risks are likely to arise. The assessment of the level of risk of a given usage could be based on the impact on the affected parties.

The mandatory requirements contained in the new regulatory framework on AI would in principle apply only to those applications identified as high-risk in accordance with the two cumulative criteria.

Following the key features given by the High-Level Expert group, some specific requirements are provided:

- Requirements about training data: data should be sufficiently broad and cover all relevant safety scenarios; data should be sufficiently representative for gender, ethnicity and other groups to avoid any kind of discrimination; privacy and personal data should be adequately protected during the use of AI-enabled products and services.
- Requirements about data and record-keeping; data related to potentially problematic actions or decisions by AI systems should be traced back and verified; data set used to train and test the AI systems should be accurately recorded; documentation on the programming and training methodologies shall be available.
- Requirements about information to be provided: Ensure clear information is provided as to the AI system's capabilities and limitations; citizens should be clearly informed when they are interacting with an AI system and not a human being.
- Requirements about robustness and accuracy; AI systems shall be robust and accurate, outcomes shall be reproducible; AI systems shall adequately deal with errors or inconsistencies during all life cycle phases; AI systems shall be resilient against both overt attacks and more subtle attempts to manipulate data or algorithms themselves, and the mitigating measures shall be taken in such cases.
- Requirements about Human oversight: the output of the AI system does not become effective unless it has been previously reviewed and validated by a human; a human can monitor the AI system while in operation and shall be able to intervene in real time and deactivate the AI system.
- specific requirements for certain particular AI applications, such as those used for purposes of remote biometric identification: EU data protection rules prohibit in principle the processing of biometric data for the purpose of uniquely identifying a natural person, except under specific conditions. Under the Law Enforcement Directive, there must be a strict necessity for such processing, in principle an authorization by EU or national law as well as appropriate safeguards. AI can only be used for remote biometric identification purposes where such use is duly justified, proportionate and subject to adequate safeguards.

In a future regulatory framework, each obligation should be addressed to the actor(s) who is (are) best placed to address any potential risks: the developer, the deployer (the person who uses an AI-equipped product or service) and potentially others (producer, distributor or importer, service provider, professional or private user). Requirements shall be applicable to all relevant economic operators providing AI-enabled products or services in the EU, regardless of whether they are established in the EU or not.
Conformity assessment would be necessary to verify and ensure that certain of the above-mentioned mandatory requirements applicable to high-risk are complied with.
The prior conformity assessment could include procedures for testing, inspection or certification. It could include checks of the algorithms and of the data sets used in the development phase.
The interested economic operators could decide to make themselves subject, on a voluntary basis, either to the above-mentioned requirements or to a specific set of similar requirements especially established for the purposes of the voluntary scheme.
The economic operators concerned would then be awarded a quality label for their AI applications.
A new legal instrument that sets out the voluntary labelling framework for developers and/or deployers of AI systems shall be created in this case. The decision of comply with the requirements will be voluntary, but in case of participation, the list of the requirements would be binding.

Policy and Investment Recommendations for Trustworthy AI, *2019* - *EU Independent High Level Expert Group on Artificial Intelligence*

Following the Ethics Guidelines for Trustworthy AI, this document contains the proposed Policy and Investment Recommendations for Trustworthy AI, addressed to EU institutions and Member States.
The recommendations focus on four main areas where the High Group of Experts believe Trustworthy AI can help in achieving a beneficial impact:

- humans and society at large
- private sector
- public sector
- Europe's research and academia
- availability of data and infrastructure,
- skills and education,
- appropriate governance and regulation,
- funding and investment.

2.4.4 Working Groups working on specific (sub-) contexts

Marelli is participating in the Working Group ISO/TC22 about the standards:

- **ISO 26262:2011/ISO26262:2018** - Road vehicles — Functional safety
- **ISO/TR 4804:2020** - Road vehicles — Safety and cybersecurity for automated driving systems
- **ISO/PAS 21448:2019** - Road vehicles — Safety of the intended functionality

With the aim of standardization concerning safety for evaluating the performance of road vehicles and their equipment.

2.4.5 Open issues of WGs

Open points raised from the latest discussions on WG on functional safety are related to the following topics:

Automated Driving

In order to minimize the potential for fault propagation and limit complexity, the development of safety-related systems is moving towards full dependent and closed systems. However, the large number of intercommunicating nodes of ADSs limits the ordinary applicability of functional safety. ADSs require new approaches to real-time fault tolerance and reasoning about consequences of faults because the fault tolerance of ADSs cannot be solved solely as a software problem since these systems work on the tight coordination among hardware, software and physical elements.

Improving ODD management: a safe state of transition should be achieved every time the ODD recognition process recognizes the performance limits of the autonomous driving system operates at. The driving functions should adapt to ODD parameters and in edge cases, a reduction of automated driving level should be allowed (e.g. from level 4 to level 3 including the take-over requests by the driver).

Connected Vehicle including End2End Safety

Increasing interlacing of automotive systems with networks (such as Car2X), new features like autonomous driving, and online software updates, may involve security risks and automated remote attacks to car fleets. Security risks and remote attacks can affect also the safety-related functions of the vehicle. For these reasons, a combined approach for safety and cyber security analysis is required

Link to SOTIF

Once the hazard analysis and risk assessment is carried out, a triggering condition analysis according to the safety goals should be performed. It would be useful to describe which is the most suitable method to define the SOTIF requirements in order to discover the weaknesses of the system design and reduce Area 3 to an acceptable level already to first phase of system development without waiting for driving tests, simulation, endurance testing, etc.

Safety demonstration for AI/ML

The goal is to manage the risk, evaluating the risk caused by missing recognition by Neural Networks. Need to coordinate with ISO/TR 4804, ISO/PAS 21448 and the AI safety initiative

2.4.6 Dependability Engineering Methods for AI-based system

In the past years, by processing the complex algorithms and actuation implemented by electrical systems, make the safety of the road vehicles much more critical respect to the past. Considering a huge incensement in the number of advanced functionalities included in the vehicles, an acceptable level of the safety for the road vehicles requires the avoidance of unreasonable risk caused by every hazard associated with the intended functionality and its implementation, especially those due to performance limitations.

In general, for majority of the systems, applying ISO26262 standard which addresses the absence of unreasonable risk due to hazards caused by malfunctioning behaviour of electrical systems is the best method of creating the safety case, which is an argument that functional safety is achieved for items, or elements, and satisfied by evidence compiled from work

products of activities during development. In this case, the safety case shall be made based on the following concepts of the V cycle of ISO26262:

- Item definition
- Hazard evaluation and risk assessment
- Defining technical safety concepts and requirements
- Defining HW and SW requirements
- Performing safety analysis
- Applying mitigations and safety mechanisms in architecture and
- Testing and validation for the whole system

However, for some systems, which rely on sensing the external or internal environment, there can be potentially hazardous behaviour caused by the intended functionality or performance limitation of a system that is free from the faults addressed in the ISO 26262. Example of such limitations includes Machine learning algorithms and AI system.

Therefore, when developing a safety-critical AI, there are a set of requirements, which must be enforced and are formally defined in the safety standards. One of the main tools for determining these requirements is the use of the safety case, which can encapsulate all safety arguments for the AI. With the development of the safety case, it must show that the AI-based system is acceptably safe and validates the actions (i.e., the output data) of the AI-based system in order to justify the use of it within the safety critical applications.

At this point, creating a safety case shall be based on a SOTIF standard (ISO 21448- ISO/TC 22/SC 32/WG 8 N 701), addressing the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or from reasonably foreseeable misuse by persons, demonstrating that all necessary safety measures are appropriately applied for AI.

In the other words, both ISO 262626 and SOTIF evaluate potential risks which can affect the vehicle safety, while ISO 26262 is much suitable for safety cases related to Functional Safety for road vehicles and deals with risks due to malfunctions of the E/E system, SOTIF investigates the possible behaviours that differ from the intended/desired behaviour of a functionality of a vehicle. Combining these two dependability domains will result in the definition of a safe function and mean that weaknesses of the technologies have been considered (SOTIF) and that possible E/E faults can be controlled by the system or by other measures (ISO 26262). Following this approach will give us the opportunity to create a safety case to guarantee a dependable system for CPSOS application including even the AI sub-system.

Marelli has been contributed to creating the mentioned safety case, particularly for the AI system, by providing a safety case checklist based on SOTIF, which will address the following concept of V cycle of SOTIF:

- Function, system specification and design (intended functionality content)
- Identification and Evaluation of hazards caused by intended functionality
- Identification and Evaluation of performance limitations and potential triggering conditions
- Functional modifications to reduce SOTIF risks
- Definition of the verification and validation strategy
- Evaluate known hazardous scenarios (Area 2)
- Evaluate unknown hazardous scenarios (Area 3)
- Methodology and criteria for SOTIF release

This checklist will be used for performing safety assessment on the whole project to evaluate the dependability of AI system by ensuring that all the activities and documentation for the Safety Lifecycle (SLC) phase of AI system have been completed as per requirements; to help prevent the failures from being introduced.

3 Workpackage related Requirements

This section lists all requirements that have a direct or indirect effect on WP3 and the dependability engineering methods to be developed. The selection of requirements is based on the requirements document release version 1.0.

Many of the requirements listed have only an indirect reference to the design methodologies and architecture pattern exclusively via the requirements for the TEACHING building blocks, but support for these must be possible through the WP3 dependability approaches. General CPSoS applications and system requirements are analysed in details in deliverable D1.1 [2]. These requirements are also taken into consideration for their impact on the development environment, tools, methods, and processes.

Requirements #2 - #8 focus on features of the TEACHING technology bricks. Therefore, influence the technology brick development in a way that either hardware or software should be compatible with robust partitioning. These isolation, WCET, and temporal safety considerations need to be supported by WP3 related development methods evolution.

Requirements #24 - #35 describe features of dependable autonomous driving and handover to manual mode. These requirements therefore are directly related to regulations and standards for autonomous vehicles (see sections 2.4.1and 2.4.2 in this document) and have to be supported on product, but also on engineering method level (see also ISO26262 [10] requirements for engineering processes).

Requirements #36 - #38 and #41 - #45 are directly geared to identify and analyse applicability of development processes and development method types in focus of WP3. The requirements focus is on methods, tools and development processes for dependable AI usage, runtime adaptation and provisioning of confidence metrics and measures.

Requirements #75, #76, #102, #104, and #105 are related to dependable AIaaS communication and security of the communication to support runtime integration of cloud-based & AI systems. The support of these requirements with engineering methods, tools and development processes is a key research challenge related to WP3 and not yet supported with SotA methods.

For a detailed list of requirement, please check deliverable D5.1 [1] and/or the requirements document release version 1.0.

Table 2 TEACHING Requirements related to WP3

Req ID	Requirements Title
2	Spatial Isolation
3	Temporal Isolation
4	Compliance with safety measures
5	Tight standalone WCET upper-bounds
6	Reasonable concurrent WCET upper-bounds
7	Monitoring features & interference channels identification
8	Synchronous global system clock
24	Determine location
25	Perceive relevant objects
26	Predict the future behaviour of relevant objects
27	Create a collision-free and lawful driving plan
28	Correctly execute and actuate the driving plan
29	Communicate and interact with other road users
30	Determine if specified nominal performance is not achieved
31	Detect when degradation is not available
32	Ensure safe mode transitions and awareness
33	React to insufficient nominal performance and other failures via degradation
34	Reduce system performance in the presence of failure for the fail-degraded mode
35	Perform ODD functional adaption within reduced system constraints
36	Dependability measures definition for trust in AI
37	HW/SW Requirementanalysis from AI application perspective
38	Identification of applicable domain safety & security standards
41	Identify threats and risks
42	Identification of AI testing methods
43	Definition of SW Update Procedure
44	Methods for Runtime adaptation
45	Confidence Metric/Measure for AI decision
75	Communication of the AIaaS modules with the vehicle
76	Communication of the vehicle with the AIaaS modules
102	Secure access from application to the adaptive system of the vehicle
104	AIaaS subsystem to manage internal module violations
105	Non-impairment of dependability
106	Annotated data for AIaaS (avionics traces)

4 Design

In this section, general process engineering perspectives and the three established main dependability architecture perspectives for the TEACHING project are described in details.

1. Dependability of AI decision making
2. AI for Dependability
3. Dependable Connected Cloud

The description of the dependability architecture perspectives includes a general depiction of the context and specifics of the applicable cases for the three dependability architecture perspectives, as well as a description of benefits and limitations form the application of the detailed dependability architecture perspective patterns.

The section is concluded by a review of industrial specifications for engineering methods and the mapping of the dependability architecture perspectives. Baseline technologies of HW supporting AI systems, sensors for human monitoring and software tools for AI development are analysed in deliverable D1.1 [2].

4.1 General Architecture

The general TEACHING platform conceptual architecture serving as basis concept for the project and the architectural considerations in this chapter are detailed in deliverable D1.1 [2]. To ensure system correctness and establish trust in systems, a comprehensive set of methods, tools, and engineering approaches was developed and continuously improved in the past decades. However, with the recent introduction of non-deterministic components (e.g., machine learning and artificial intelligence) into dependable systems, new challenges arise. Several questions need to be answered regarding dependability and standard compliance, including process engineering aspects and technical engineering aspects.

From a **process engineering perspective**, the non-deterministic system behaviour is addressed by developing new standards, such as SOTIF (Safety Of The Intended Functionality). SOTIF is a branch of technical product safety. It focuses on the undefined question of how an intended functionality is to be specified, developed, verified and validated to be considered sufficiently safe. Currently, the standard *"ISO 21448 - Road vehicles - Safety of the intended functionality"* is being developed especially for the automotive sector, and therefore, it is closely related to the *"ISO 26262 - Road vehicles - Functional safety"* and *"ISO/SAE 21434 - Road vehicles - Cybersecurity engineering"*. Hence, industry and science are working on integrated process engineering approaches that support the development of dependable products that rely on non-deterministic algorithms.

From a **technical engineering perspective**, it is necessary to distinguish between component functionality and component integration. There is no difference between using (the output/result of) deterministic and non-deterministic functions from a purely functional point of view. However, **there is a difference in how deterministic and non-deterministic functions shall be integrated into dependable systems**.

When **integrating a deterministic function**, we can be sure that the function will always produce the same output for a given input. Hence, it is possible to design a system where we can predict the system behaviour under **all considered circumstances**, enabling the construction of sufficiently safe products. For that purpose, process engineering provides methods, tools, and strategies to support technical engineering during design, development, implementation, and testing.

On the other hand, when **integrating a non-deterministic function**, it may happen that even for the same input, the function produces different outputs on different runs. Hence, it is not possible with traditional processes and engineering approaches to design a system where we can predict the system behaviour **under all considered circumstances**.

Next, we will discuss the technical aspects and influence of integrating non-deterministic functions into dependable systems.

The following two criteria can be used to classify the set of observable scenarios resulting from a system action: **safe or unsafe** and **known or unknown** scenarios, leading to the four categories of scenarios shown on the left of Figure 10 from [11]. The general goal in dependability (and safety) engineering is to decrease areas 2 and 3.

The usage of non-deterministic functions increases areas 2 and 3, because non-determinism introduces uncertainty. This introduced uncertainty increases the unknown system behaviour, which may also increase the area of unsafe system behaviour. Since some innovative features in vehicles and aircraft rely on non-deterministic functions, these functions must be integrated appropriately to not violate system safety.

1. The **risk of known unsafe** scenarios, for example, can be mitigated by classical/traditional safety measures such as limiting the operational design domain.
2. To limit the **risk of unknown** system behaviour,
    a. one can, e.g., reduce the non-determinism of the function itself, or
    b. one can reduce the effect of non-determinism on safety by design.

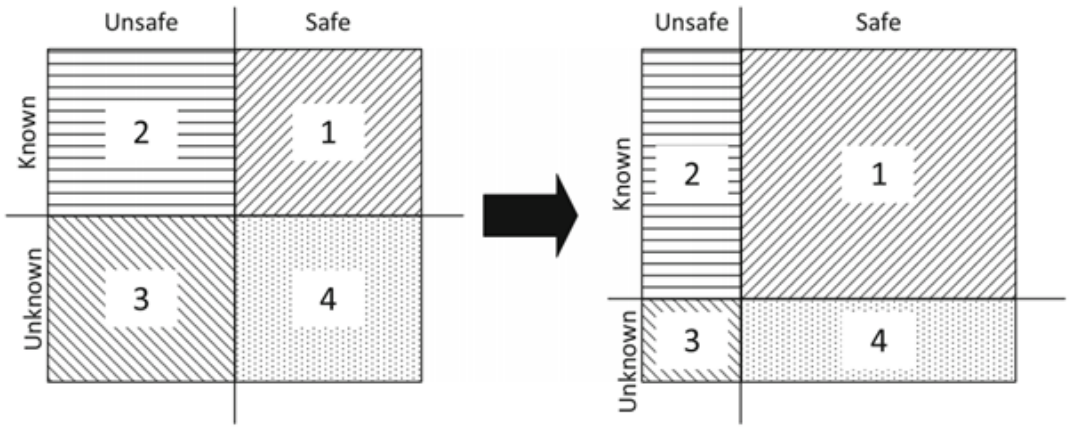


Figure 10 To build sufficiently safe systems, dependability and safety engineering try to minimize the areas 2 and 3 (image from **[11]**).

Although some approaches were developed to address point 2.a [12] [13], such as the prediction or classification of the dependability of neural networks, these approaches are still in their first stages of development. Therefore, it remains challenging to verify neural network behaviour because

- their decisions are not explainable,
- they cannot be exhaustively tested, and
- finite test samples cannot capture the variation across all operating conditions.

Subsequently, we will discuss three system types or views (subsequently termed “dependable architecture perspective”) that use neural network technology at different levels within the system to perform their intended functionality. For each system type, we first explain the architecture and functionality. Then, we discuss critical aspects (i.e., the non-deterministic parts) that affect system dependability, and finally, we propose our first ideas of how those critical aspects could be addressed on the architecture level.

4.2 Dependable Architecture Perspective 1: Dependability of AI decision making

Context. To build autonomous systems, industry and science rely on artificial intelligence (AI) capabilities as decision-making units. An example of such an autonomous system is shown in Figure 11, where AI systems should replace the human driver in fully autonomous vehicles. To that purpose, the AI system must be able to percept and interpret the vehicle environment for calculating the input values for the setpoint generator in every specific driving situation.

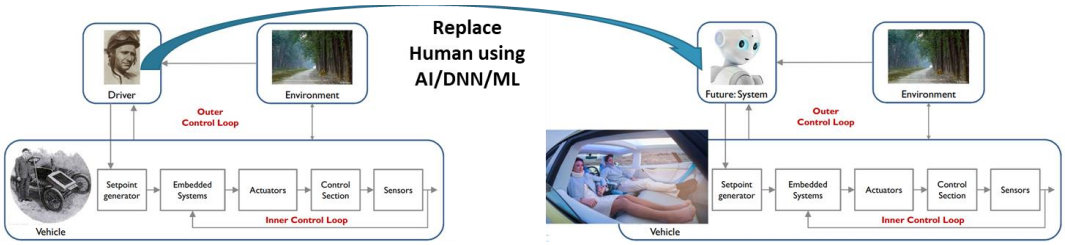


Figure 11 AI systems should replace the human driver in the future fully autonomous vehicles.

In traditional systems, the human driver was responsible for generating adequate inputs in all driving situations for the setpoint generator by actuating the provided interfaces (i.e., steering wheel, gas pedal, braking pedal, etc.) When replacing the human driver with an AI system, the AI system itself becomes responsible for generating adequate inputs for every driving situation. Since the generated inputs have a critical impact on system safety, it must be guaranteed by the vehicle vendor (i.e., the original equipment manufacturer) that AI-generated inputs do not violate system safety.

Problem. AI-based systems are non-deterministic systems, and to the point of writing, **it is not possible to verify AI safety**, because AI decisions are not explainable, they cannot be exhaustively tested, and finite test samples cannot capture the variation across all operating conditions an AI system will be faced during its lifetime. **However, when building safety-critical systems, compliance with safety standards and related standards is required**. **Therefore, it is necessary to provide an argument for system safety if an AI system is part of the critical signal path** [14]**.**

Solution. To still leverage the capabilities of AI-based systems as safety-critical decision components, three different architectural concepts have been discussed within the TEACHING project.

All three concepts have in common that the output of the AI-based system is considered unsafe. Therefore, the AI-based system is strictly separated from safety-critical system domains. The AI-based system output can be integrated into the safety-critical system domains only via well-controlled interfaces.

In Figure 12 - Figure 16, the blue components are considered to be unsafe, while the orange components are considered to be safe.

Concept 1: Human-in-the-Loop. In this concept, the AI-based system is responsible for observing and analysing specific tasks or components and recommends human-readable actions. As a "safe" decision gate, the human decides whether the recommendations of the AI should be applied and how they should be applied.

Positive Consequences

+ The necessity of human intervention allows one to apply traditional safety measures to guaranty system safety (attention: see also the negative consequences).
+ The responsibility of analysing complex situations and tasks is transferred to the AI algorithm, which frees up resources of the human, otherwise dedicated to the analysis.

Negative Consequences

- Suppose a wrong decision of the AI algorithm (i.e., detection or non-detection of a critical situation) could violate system safety. In that case, the AI algorithm itself must be considered as a safety-critical component, and traditional safety measures are no longer applicable.
- The system does not operate autonomously because human intervention is required.

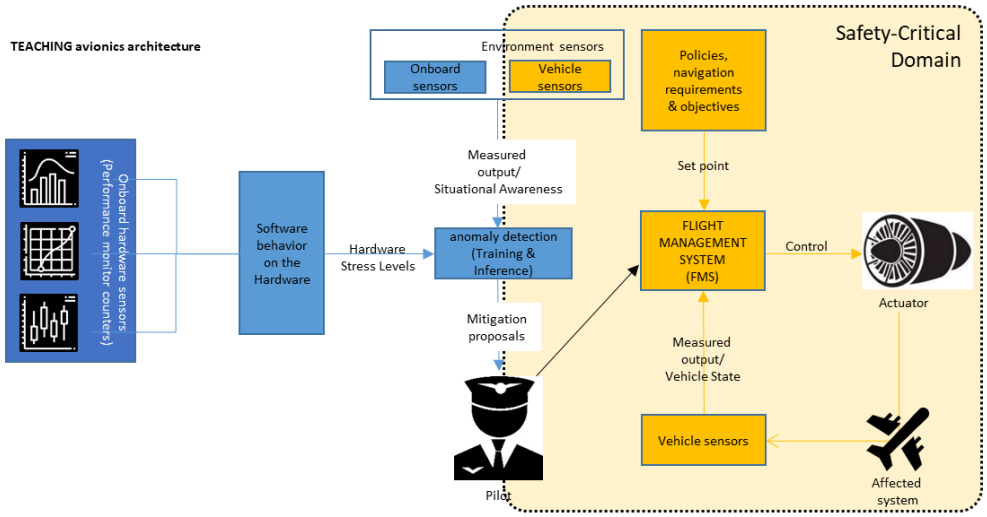


Figure 12 Concept 1: Human in the decision loop.

Concept 2: Policy-based decision integration. In this concept, the AI-based system is responsible for observing and analysing specific tasks or components and recommends machine-readable actions that can be translated into a finite set of policies and objectives. These policies and objectives are then used to influence the setpoint generation of the safety-critical system domain.

Positive Consequences

+ The finite set of policies and objectives can be analysed for safety, and traditional safety techniques can be applied to guaranty system safety.
+ The responsibility of analysing complex situations and tasks is transferred to the AI algorithm, which frees up resources of the human, otherwise dedicated to the analysis.
+ The system operates autonomously because no human intervention is required to integrate the actions recommended by the AI algorithm.
+ The policy-based approach can be implemented in a resource-efficient manner.

Negative Consequences

- Suppose a wrong decision of the AI algorithm (i.e., detection or non-detection of a critical situation) could violate system safety. In that case, the AI algorithm itself must be considered as a safety-critical component, and traditional safety measures are no longer applicable.
- Since the set of possible actions is limited to a finite number of policies and actions, the AI algorithm's capabilities might be restricted by this limitation.

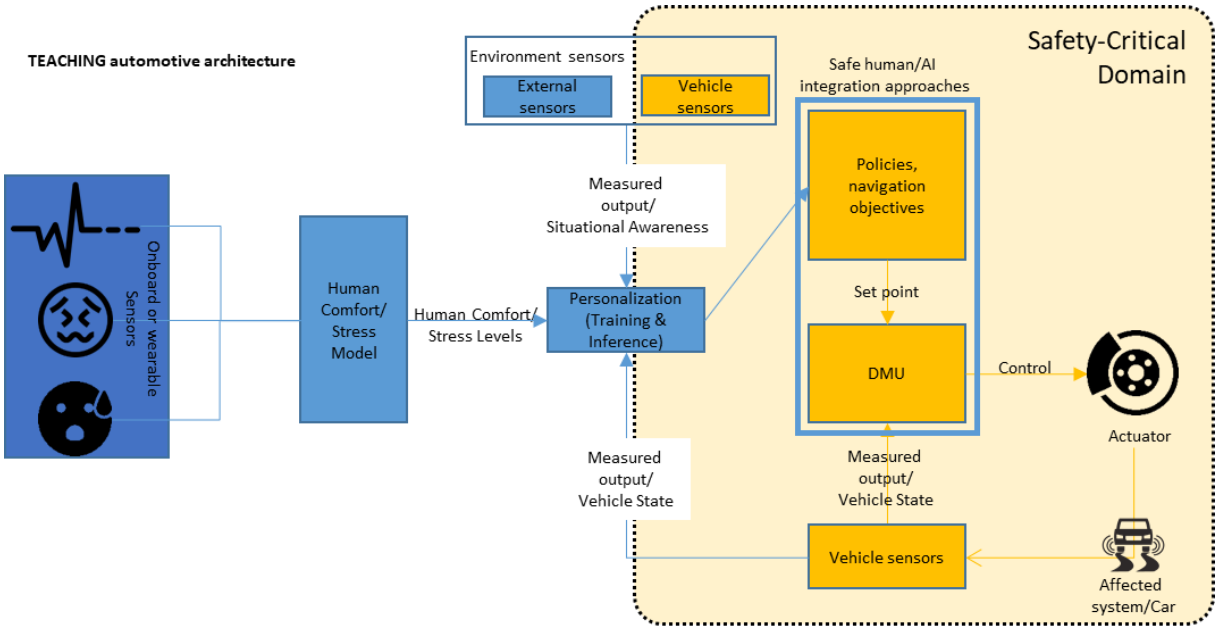


Figure 13 Concept 2: Policy-based integration of the AI-based system into the safety-critical domain.

Concept 3: Model-based decision integration. In this concept, the AI-based system is responsible for observing and analysing specific tasks or components and recommends machine-readable actions. Instead of mapping these actions to a finite set of policies or objectives, the model-based integration approach compares the non-deterministic output of the AI-based system with the output of a deterministic model running aside the AI-based system.

The AI-based and deterministic models are designed for the same objectives, while the deterministic model is also designed to meet classic safety systems requirements. Hence, the deterministic model can be used to validate the AI-based system's output to ensure system safety.

Positive Consequences

+ The system running the deterministic model can be analysed for safety, and traditional safety techniques can be applied to guaranty system safety.
+ Since the deterministic model is less restrictive than the policy-based approach, the AI algorithm's capabilities are less restricted.
+ The system operates autonomously because no human intervention is required to integrate the actions recommended by the AI algorithm.
+ A wrong decision of the AI algorithm (i.e., detection or non-detection of a critical situation) does not violate system safety. Hence, the AI algorithm does not need to be considered as a safety-critical component.

Negative Consequences

- The limitations of the deterministic model might restrict the capabilities of the AI algorithm.
- Two nearly "equally intelligent" systems must be developed.
- Two resource-intensive systems must be executed side-by-side in a synchronous manner.

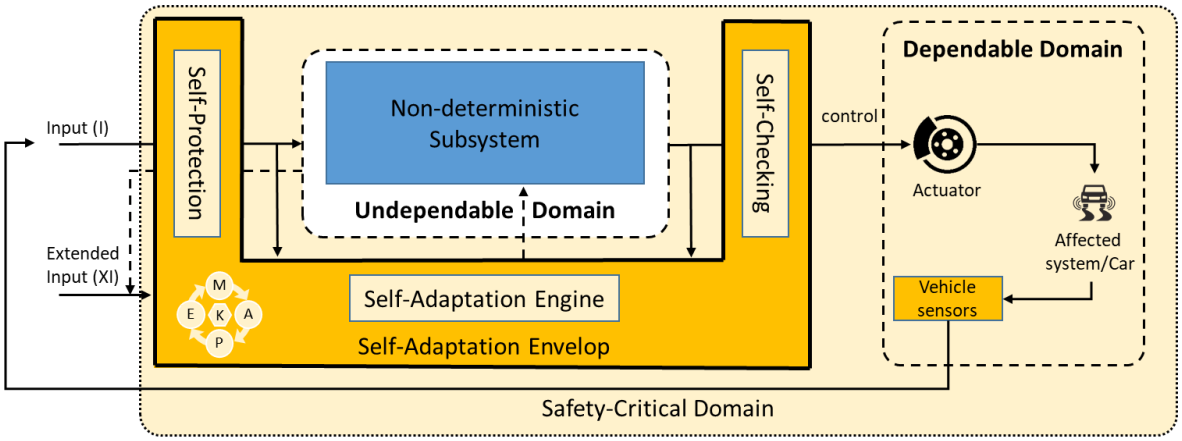


Figure 14 Concept 3: Model-based integration of the AI-based system into the safety-critical domain.

4.3 Dependable Architecture Perspective 2: AI for Dependability

Context. Artificial intelligence (AI) is used to enhance the dependability of systems. To that purpose, the AI monitors and learns the behaviour of a (dependable) system under observation (SUO). In case the AI detects abnormal behaviour, countermeasures can be either recommended or automatically triggered. In case the countermeasures and recommendations are safety-critical, the aspects discussed in the Dependable Architecture Perspective 2 shall be considered.

Problem. The monitoring system and the AI algorithm should learn the normal/expected system behaviour under real operating conditions without influencing the functionality and dependability of the SUO:

- **Functionality** - The functionality of the system under observation must not be affected by the monitor unless the system violates its specification [8].
- **Schedulability** - The hard real-time guarantees of the system must not be affected by the monitor architecture unless the system violates its specification [8].
- **Reliability** - The reliability of the system under observation alone must not be smaller than the system reliability under observation in the context of the monitoring architecture [8].
- **Certifiability** - The source code of the system under observation must not be unduly modified by the monitor architecture [8].

Solution. The avionics use case uses the Human-in-the-Loop Architecture Concept discussed in the Dependable Architecture Perspective 2. In this concept, the AI-based system is responsible for observing and analysing specific tasks or components and recommends human-readable actions. As a "safe" decision gate, the human decides whether the recommendations of the AI should be applied and how they should be applied.

Positive and Negative Consequences

The dependability analysis and the mentioned problems will be examined for their positive and negative consequences in the remainder of the project.

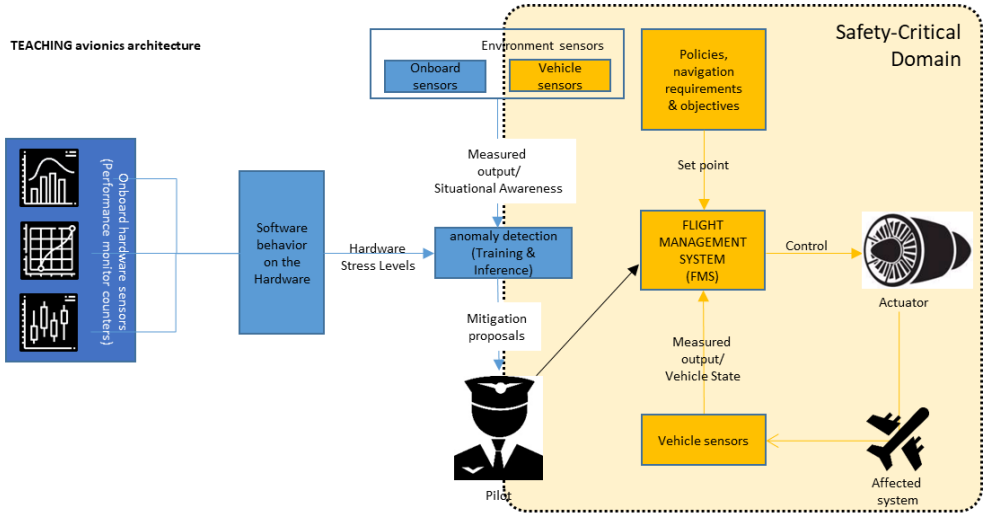


Figure 15 AI to increase system dependability.

4.4 Dependable Architecture Perspective 3: Dependable Connected Cloud

Context. The trained artificial intelligence (AI) runs on deeply embedded systems on the field layer and is responsible for controlling/enhancing specific low-level system functions, like those described in the Dependable Architecture Perspectives 1 and 2. Feedback from the field layer is continuously transferred to edge or cloud devices to continue learning and improve the field layer AI algorithms. The newly trained AI algorithms are validated and transferred back to the field layer for integration and deployment.

Problem. Whether the field layer AI algorithm is safety-critical or not. It is of utmost importance that the continuous learning and updating of the algorithm do not compromise system dependability on both the field and the edge/cloud layers.

Solution. The dependable connected edge/cloud approach provides added value to both previously discussed perspectives. It ensures that systems, once deployed, can be improved, retrained, and tailored to meet future requirements unknown at system design and development time. Self-adaptive system concepts across all layers are expected to provide the necessary flexibility on the architecture and software level to meet these demanding requirements.

Positive and Negative Consequences

The dependability analysis will be examined for their positive and negative consequences in the remainder of the project. A more detailed discussion of the edge and cloud layer algorithms for AI training and validation follows in Section 5 and Section 6.

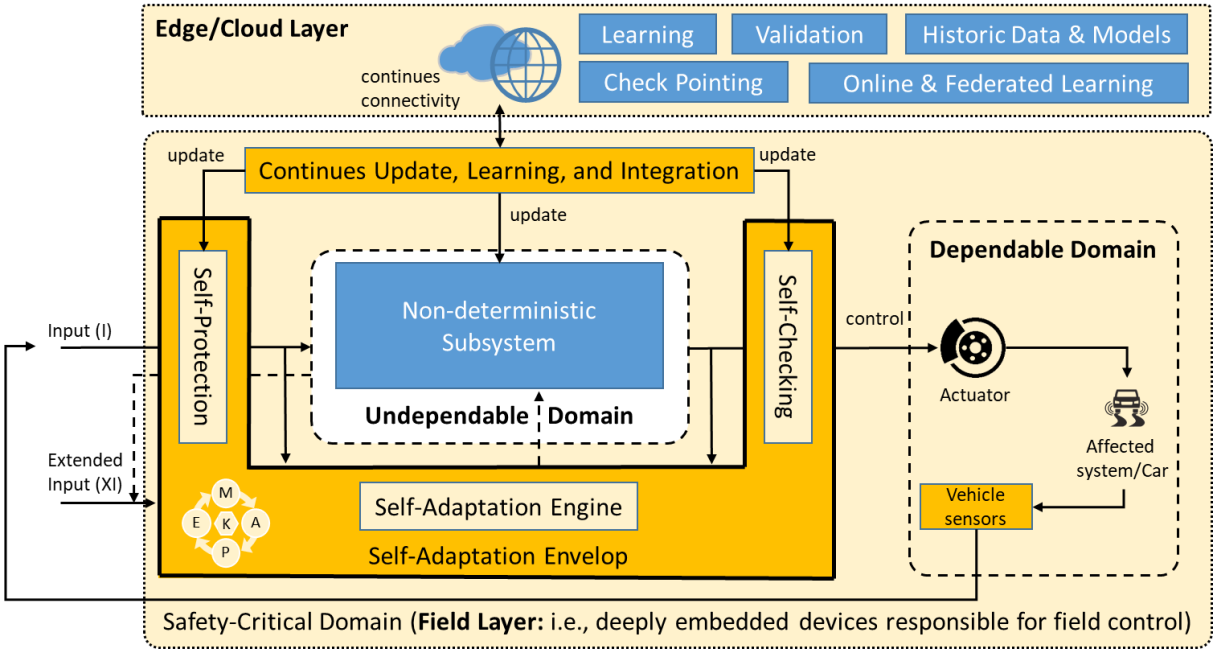


Figure 16 Dependable connected cloud for continuous learning and improvement.

4.5 Specification of industrial dependability engineering approaches

Currently, one of the most widespread safety norms in the automotive sector is ISO 26262 [10]. It is strongly connected to the V-model approach, which describes a methodical and rigid way to address product development with respect to hardware and software aspects. Still, ISO 26262 mainly addresses safety and reliability, while other aspects that are considered relevant in dependable systems (such as confidentiality, which is one of the main issues arising from the connectivity of the vehicles) are kept out of the scope.

While the V-model is mostly regarded as a golden standard for the development of safe products, it also creates a considerable overhead in both time and resources. These overheads are the reason why in recent years a discussion around the adaption of a "custom" agile process has been spreading. The concept behind this would be to keep a light, consistent, and always updated documentation to support the safe processes. However, this is highly debated and seldom adopted because of its pioneering spirit and its uncertain practicality.

The automotive industry is thus mainly focused on the classical ISO 26262 with a V-model approach that is applied in large as well as medium-small companies of the sector. In particular, in many companies, like I&M, an internal methodology based on the A-SPICE V-model has been adopted in 2020. This methodology applies the classical V-model, required also from some customers asking for eventual ISO 26262 applications, while also partly relieving the documentation process while the product is actually implemented.

Concerning the future years, the fate of ISO 26262 is uncertain, since its models would be obsolete in a completely autonomous system. There are three main reasons for this:

- firstly, because it does not take into account any interaction with the driver/passenger, which could be fundamental in some situations;
- secondly, because in a highly-connected system any single fault could be impactful on other subsystems without some specialized solutions;

- and finally, because of the necessity for a completely autonomous vehicle to be able to provide a service even in case of failure (a complete shut down of the system could also be dangerous).

For the latter, the ISO 21448 norm (SOTIF) [11] is a viable solution that could be applied in the future for developing systems suitable for completely autonomous driving.

Agile development methods are newer approaches that are in use for development of critical automotive systems in recent days. Nevertheless, compliance with automotive standards and mapping to established SotA practices is mandatory. Frequently automotive industry makes use of agile methods such as Scrum and Kanban. Since Automotive SPICE [9] solely defines "what" is to be done and not "how" the process is to be implemented the application of agile methods is also suitable.

Agile engineering can be assigned to both the "what" and the "how" level, although most agile practices take place mainly at the "how" level. In practice, agile practices implement some of the SPICE principles, and these serve as an abstraction of the agile elements.

Neither logical points in time at which work products should be available, nor other types of activity sequences are defined. Instead, ASPICE requires the selection and use of a reasonably chosen life cycle model that defines such sequences. The actual and sensible choice is a decision where a healthy interplay between Agile Engineering and ASPICE is possible. Great potential for agile process models lies in the increase of process transparency.

Nevertheless, certain areas in the agile manifesto do not receive sufficient attention and thus encounter ASPICE non-compliance. The biggest difficulty here is quality assurance. An agile team relies on the offered freedom to constantly optimise ways of working. This freedom is restricted by external ASPICE requirements; therefore, a recommendation is to only concentrate on the results at the end of an iteration.

5 AI approaches for ensuring CPSoS dependability

5.1 Introduction

The exponential growth of digital infrastructure and connected devices in recent times increases enormously the cyber threats surface. CPSoS systems are inherently exposed to cyber threats because of their complexity. Every IT component of a CPSoS running a piece of software of any kind can be compromised resulting in an alternated and unexpected behaviour. The probability of such an eventuality rises as the number of network connections of a CPSoS component to other internal or external CPSoS components increases. This is why the first cybersecurity protection measure to safeguard an IT system is its isolation both physically and network-wise from its external environment. However, when such an approach is not possible, as in TEACHING's automotive and avionics use cases, advanced strategies need to be employed to strengthen the defence against cybersecurity risks.

Cybersecurity mechanisms are indispensable for ensuring the dependability of a CPSoS system and in particular, the confidentiality, integrity and availability aspects entailed by the concept of dependability, as extensively presented in the deliverable D1.1 (to be submitted together with this deliverable). Confidentiality risks appear mainly when sensitive information is transferred between CPSs' components (e.g., eavesdropping attacks). Integrity vulnerabilities render a system susceptible to malicious alternations of its deployed software resulting in behaviours that deviate from the system's normal functional state (e.g., data injection attacks). On the other hand, the broadly observed category of denial-of-service (DoS) cyberattacks can stress and exhaust a CPS's resources (throughput, memory, CPU), compromising its availability (e.g., traffic flooding attacks).

The main concern regarding counteracting cyberattacks is the difficulty in promptly discovering an attack before it has laid its impact on the targeted system. The increasing motivation and resourcefulness of attackers across the internet create a range of ever-expanding and ever-evolving threats that closely follow new trends and innovations in technology.

In this setting, the shortage of talent in cybersecurity professionals has been unfortunately observed in the last decades[3]. Machine learning (ML) solutions, albeit vulnerable themselves to cyber attacks, seem unequivocally to be the only way to effectively detect and accordingly defend the multitude of cyberattacks that are continuously active worldwide, in order to avoid potentially catastrophic results in CPSoSs and critical infrastructure.

5.2 Machine Learning in cybersecurity

Supervised, semi-supervised (adaptive) and unsupervised methods have been applied for cybersecurity purposes since the '80s. Because of the lack of large amounts of labelled data, unsupervised methods focusing on anomaly detection have prevailed throughout the years. Using 1-class classifiers, these methods attempt to let an AI module learn a system's normal behaviour, and based on this knowledge; infer that an abnormal behaviour has occurred. Nevertheless, rule-based security threats detection, based on past events (attacks' signatures) recorded in relevant repositories had proven to be greatly more effective than ML until recently. In recent years, though, with the surge of Big Data, data in large amounts and of better quality has been available, while the advancements in ML have facilitated novel approaches to show promising results.

[3] https://cisomag.eccouncil.org/cybersecurity-artificial-intelligence/

For *intrusion detection*, ML extends the exploitation of simple rules-based logic relying on traditional misuse detection [15] to promptly respond to various types of attacks such as denial-of-service (DoS), scanning, user-to-root (U2R), remote-to-local (R2L), fuzzers, analysis, backdoor, etc. Features of network traffic (e.g., protocol, service, number of login attempts, packets per flow, bytes per flow, source address, destination address, source port, destination port, etc.) are used to build an anomaly detection system that is able to discover suspicious patterns of behaviour. Machine learning techniques such as neural networks, clustering algorithms and one-class support vector machine have been proposed in the literature. They perform well in identifying intrusion patterns not known in the past, but they suffer from high rates of false-positives. Combining misuse detection with anomaly detection results in improved hybrid methods of intrusion detection.

Malware detection has long worked on the basis of signature-based systems. Nowadays, that malware evolves at a rapid pace producing a multitude of variants for every single malware, combining known signatures with ML techniques creates a more powerful defence line for such threats. Malware is analysed statically or dynamically, that is with or without execution of the relevant code. Code representation relies on the n-gram model or the graph-based model. The classification of a piece of software as being malware or not has been studied using various ML techniques, such as Bayesian network, naïve Bayes, C4.5 decision tree variant, logistic model trees, random forest tree, k-nearest neighbour, multilayer perceptron, simple logistic regression, support vector machine, and sequential minimal optimization [16].

Relevant to malware detection but from a preventive perspective, the *code vulnerabilities' discovery* is another field that works towards counteracting cyberattacks. In this domain, traditional approaches include software penetration testing, fuzz-testing and static data-flow analysis. Adopting data mining and ML techniques, several approaches in the literature attempt to build vulnerability and fault prediction models on software features (e.g., complexity, code-churn etc.) and text (e.g., using n-grams), syntactic, API, or data-flow analysis to assess the quality of a piece of software and reveal its weaknesses. Statistical correlation analysis, logistic regression, Bayesian networks, support vector machine, random forest and genetic algorithms are some of the approaches that have been tested with variable results [17].

Fraud detection works towards discovering fraudulent transactions with a system, constituting another cybersecurity measure that can strengthen the defence line of a computer system against cyberattacks. The concept of fraud in cybersecurity refers to the act of deceptively gaining access to personalized services through impersonation of the grantee of the service. Discovering fraudulent transactions focuses on the analysis of data objects inter-connections and inter-dependencies based on identifying network structures or on patterns of user behaviour and interactions [18]. Artificial neural networks exhibit the most promising results in fraud detection and are generally preferred, with decision trees, support vector machine, naïve Bayes, random forest and k-nearest neighbour methods following [19] in performance and popularity.

Eavesdropping, *man-in-the-middle*, *jamming* and *spoofing* attacks are highly probable to occur in CPSoS, where systems are building their connections in an ad-hoc and dynamic fashion. A CPSoS, which by nature relies on a collection of IoT devices for a multitude of monitoring tasks, is additionally vulnerable to cyber threats (network, software or privacy) because of the dynamicity of the network where IoT elements belong to, as well as due to the limited resources of these devices in terms of processing power. Computational-intensive and latency-sensitive tasks are highly prohibitive for such devices. A variety of ML methods (supervised, unsupervised and reinforcement learning) have been employed in this field [20], for learning-based authentication, learning-based access control, secure IoT offloading with learning and learning-based IoT malware detection. A combination of all these precaution measures can

create a safety net for protecting a heterogeneous and highly volatile environment of IoT devices.

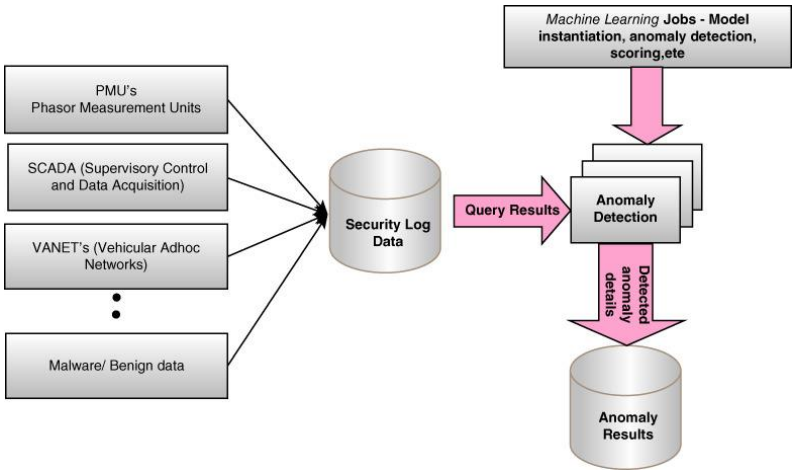


Figure 17 Anomaly detection using ML techniques **[21]**

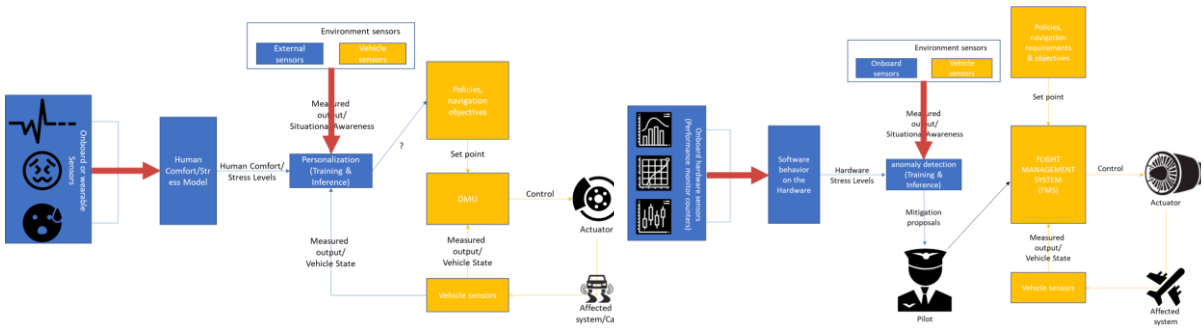
In conclusion, ensuring cybersecurity in computer systems of all kinds is a rather complex task that requires multiple layers of proactive and reactive measures to be taken for being effective. Nevertheless, whatever the particularities of each case of a cyberattack category may be, the detection of a threat, attack or vulnerability comes down to the broad range of methods that fall into the category of anomaly detection [21]. Machine learning techniques are the only way of analysing large amounts of data emerging from monitoring computer systems with the aim of identifying anomalous behaviour of any sort. Data collection of good quality from the system activity logs, appropriate pre-processing, feature extraction and construction of an accurate machine learning model are the steps towards achieving the desired results, as depicted in Figure 17.

5.3 Addressing dependability from a cybersecurity perspective in TEACHING

The use cases that TEACHING is focusing on are based on autonomous driving and aviation cyber blackbox. Cybersecurity is essential for ensuring the dependability of automotive and avionics systems in order to protect these CPSoSs from exhibiting dangerous or even catastrophic behaviours. Task 3.4 concentrates on the security aspects of dependability for automotive and avionics systems, i.e., availability, confidentiality and integrity, and in particular its cybersecurity component.

Vehicular ad-hoc networks (VANET) employed in autonomous transportation systems face particular challenges that add to the generic anticipated threats outlined in the previous section [22]. Timing attacks, routing attacks and internal vehicle network attacks are only a few of them. Many ML methods, which consider the ad-hoc network connectivity in this setting, have been recently tested for VANET cases with encouraging results. Misuse, anomaly and hybrid detection approaches presented in the literature propose among others a Bayes-learning-based alert correlation algorithm for coordinated attacks, an NN-based security system for data injection attacks, support vector machine for malicious attacks detection, random forest for identifying malicious nodes in the VANET and K-means for unsupervised learning of anomalous traffic discovery [23].

In modern avionics environments, Air-to-Air and Air-to-Ground connectivity capabilities, which tend to become a standard, enlarge the cyberattack surface of aircrafts. However, security assessment is a relatively new concept in aviation, although avionics do not suffer less from vulnerabilities that burden CPSs, in terms of network connectivity, hardware, software and data exchange [24]. A common approach to tackle the relevant security challenges is the deployment of a host-based intrusion detection system (HIDS) on the aircraft. One-class support vector machine, automata and timed automata have been proposed as part of a HIDS system [25]. Nonetheless, for the avionics sector, there is a largely unexplored research area in exploiting AI and ML for effectively addressing cybersecurity.



(a) automotive use case (b) avionics use case

Figure 18 Anomaly detection in sensors output for TEACHING use cases

In TEACHING, the focus on cybersecurity is on data provided by the sensors incorporated in the CPSoS systems of the two project use cases. The high-level system architecture of the automotive and avionics CPSoSs to be studied in TEACHING is depicted in Figure 18. Although different, the similarities of the two architectures in terms of their main structural components (i.e., sensing, cyber and control layers) are evident. Cybersecurity will naturally be addressed at the data entry points of these systems (denoted with red arrows in Figure 18). Anomaly detection will be performed at the sensing layer of the CPSoSs (i.e., on the data ingested to the system by on-board, environmental and wearable sensors), which enables the communication of the system with its environment, but which at the same time increases the risks for cyberattacks.

In the project, recurrent neural networks (RNN) [26] and reservoir computing (RC) [27] methods are going to be extensively used as described in the TEACHING software architecture (D4.1 to be submitted together with this deliverable). RNNs, and by extension RC, constitute a type of machine learning technique that demonstrates dynamic characteristics interesting for anomaly detection. In particular, these machine learning approaches prove suitable for time series data. Instead of classifying a single observation coming from the current state of the system independently, a sequence of states is considered, and as a result, the correlation of these states is captured. The models built following this ML paradigm allow recent observations being weighted as more important, but without neglecting knowledge gained by past observations. These properties are expected to produce interesting results in investigating cybersecurity attacks in the automotive and avionics domains [28].

In order to succeed with this goal, appropriate datasets need to be provided by the pilot partners for the sensor equipment that is going to be deployed for both pilots. In addition, this data needs to be labelled to some extent, so that the assessment of the applied methods in terms of their effectiveness and efficiency is feasible. In absence of such information, benchmark data is going to be used.

6 Dependability engineering of cloud-connected AI-based systems

In this section, we focus on cloud-connected AI-based systems and how the different components, running either close to the user, possibly on edge devices or on the cloud, interact in order to provide an increased level of dependability. Security and dependability in AI-based systems is a very active research field with new techniques being constantly developed. Thus, our engineering strategies and patterns must be specific enough to increase dependability but at the same time be generic enough in order to be able to incorporate and integrate future techniques.

When a system operates autonomously, making decisions using feedback from several AI components, the complexity of detecting events and/or conditions that impact normal operation becomes even more challenging. A system should be able to adapt to a new reality in a constantly changing environment and maintain its basic capabilities in tandem, using a dependability mechanism.

6.1 Failure Detection & Operational Compliance

A validation mechanism or an extra layer of validation functionality is needed with a high-level goal of preserving compliance to a predefined set of rules, which define the basic system's functionality. Some events of non-compliance might be the result of operational mistakes from a human user, a service failure at a lower level such as a hardware fault or a malicious attack from some intelligent adversary. Providing fault-tolerance at the system level naturally entails fault-tolerance at the AI level.

Failure scenarios, for example in autonomous vehicles if the vehicle drives closer to the lanes than usual or has an abrupt change of speed, could theoretically be identified either by input sensors or by leveraging user related input sensors which help classify the user's overall experience in terms of comfort (or the lack of it). Additionally, potential failure scenarios should also be registered when the system's response deviates significantly from responses during training. Warning systems, checking the input data feed, can also provide a basis for a plan on which each action is based on a scenario related to a specific problem (abrupt change in camera input versus change in acceleration).

Each machine learning component performing inference on the user's device should be accompanied by an additional component performing basic validation. The exact nature of the validation mechanism, enforcing constraints on the response of the component, highly depends on the ML algorithm used. Additionally, when relying on the user's explicit feedback, the decisions of the AI-based components must first be transformed into *explainable* before presented to the user. All these mechanisms must be lightweight enough to be integrated into the energy- and/or efficiency- aware environment where the user resides. Additional, more computationally intensive constraints' validation can be performed in the cloud on a periodic basis.

During its lifetime, an ML-based CPSoS will encounter situations where the input data distribution will not be the same as the distribution of its training dataset. Thus, early detection of concept drift and in general, the detection of situations where an ML model could potentially deteriorate quickly is of paramount importance. Furthermore, such a change in the input distribution can be gradual or abrupt. Drift detection methods such as ADWIN [29] and ECDD [30] (Ross et Al) can work as an alert when corruption occurs in terms of catastrophic interference.

Security and trust of CPSoS and ML-Based methods are still in their infancy. Nevertheless, the assessment of the system's performance and the identification of failures is crucial not only for deploying mitigation actions and performing self-optimization, but also for collecting relevant samples (input data, environmental conditions, system state, sensors etc.) which can be further analysed in the cloud where processing power is abundant.

6.2 CheckPointing (digital twin)

The automated AI failure assessment should be accompanied by necessary policies for self-optimization. Energy-aware strategies should be deployed closer to the end-user where the main responsibility would be the identification of failures and the collection of relevant samples. The fusion and integration of such valuable data in newer versions of the AI models must be autonomously processed in higher levels of the infrastructure, where more processing power is available.

As such, according to the digital twin approach, the cloud or a potential edge infrastructure is employed for regularly fail-proofing the current model under a set of benchmarks. Copies of the current model will be frequently pushed to the cloud/edge infrastructure, where its performance and operation will be certified based on a set of simulated runs under predefined extreme conditions. A potential failure to pass the certification tests will result in rolling back to a working version of the model. These simulated runs can employ either a model-based approach for simulation, or perhaps use a data-driven approach with specially crafted private examples, which help stress test the models.

6.3 Consensus

Combining several classifiers has been extensively used in the pattern recognition literature under various names such as combination of multiple classifiers, classifier fusion, committees of neural networks and classifier ensembles, to name a few. While originally used to improve the overall accuracy of the classifier, they can also be used for robust classification in adversarial environments [31] [32]. An approach, which includes an ensemble of models/systems, uses the cloud infrastructure as a host for the ensemble. The component close to the user could periodically communicate with the aforementioned ensemble, which consists of a set of baseline models and/or a set of verified/certified copies of a former model. A common decision or a model correction could be the output of this communication.

6.4 Online & Federated Learning

Many real-world CPSoS can be viewed as adaptable systems whose behaviour changes using rounds of self-optimization. In the online setting, data observed from the devices running close to the user are used in order to train the ML model locally, thus enabling the algorithm to dynamically adapt to new patterns. In a CPSoS where user input can be easily collected explicitly or implicitly using sensors, such online training can be beneficial. Unfortunately, online learning may be prone to catastrophic inference. Similar problems arise when federated learning is used.

Preserving former knowledge of an ML-based system is strongly connected to the mitigation of catastrophic interference/forgetting, which is still an open research problem. A dependable system can be build using a combination of known methods for catastrophic forgetting that can either work at runtime (such as verification, or adaptation of new data) or at a specific point in time, either by an user or by another system such as reintroducing a subset of former knowledge in order to repair any irreversible changes in the system.

Possible suggestions could be *Episodic Memory* or *Rehearsal* methods. These methods hold a small part of samples from older tasks or states and re-introduce them to an ever-learning system in order to mitigate catastrophic interference. Depending on the context, such methods have been successfully deployed in incremental learning [33] [34]. These methods store a representative amount of previous data points and combine the ability to transfer knowledge from a previous state of the same model, also known as *knowledge distillation* or *dark knowledge*. Motivated by security, Tramèr et al [35] introduce *ensemble adversarial training*, a technique that augments training data with perturbations transferred from other models. Papernot et al [36] developed a defensive mechanism called defensive distillation to reduce the effectiveness of adversarial samples on deep neural networks. They also employ transfer learning and borrow ideas from distillation. Finally, another possible solution is a controlled change in the system as described by Kirkpatrick et al. [37] using elastic weight consolidation on which the changes on a system are constrained by an information matrix in order to maintain former knowledge and learn new tasks simultaneously.

From the above discussion, we conclude that CPSoS need to incorporate support for components, which (a) re-introduce former knowledge and (b) create constraints on learning in order to avoid forgetting. Such components need to exist both close to the user where learning happens and on a much larger scale in the cloud where it is easier to safely store old examples and retrain large models. Additionally, during the interaction between the different tiers of the infrastructure where model handoff takes place, either from the user device to the cloud or vice-versa, a certification mechanism ensuring integrity and standards compliance needs to be foreseen.

7 Conclusion

The aim of TEACHING project's work package 3 is to enhance the project's technology brick development by building a dependable engineering environment that supports the development of self-adaptive artificial humanistic intelligence in a dependable manner. In this context, WP3 focuses on the establishment of engineering methods, architectural concepts and design patterns that can be used to develop dependable and AI-based autonomous system development.

Dependability engineering of adaptive, cloud-based and/or AI-based systems is still a topic where first concepts need to be instantiated (like practical processes and methods, covering the whole lifecycle). The assurance of dependability, especially considering novel AI-based and/or dynamical runtime-based approaches is still an open issue that is lacking in common solution so far.

The goal of this deliverable was to identify gaps with existing solutions for the management of CPSoSs throughout their life cycle including design and operational phases (architectural frameworks, conceptual models, process frameworks etc.). Based on this analysis, architectural, process and development framework will be developed to support automated dependability evaluation of CPSoS (Obj. 5 of TEACHING project).

In compliance with its intended purpose, this document presented the established body of knowledge of all WP3 activities at Milestone 1. The intention was to have a first release version of the WP3 research activities to continue building TEACHING technology bricks based on these methods and patterns and to have a more fluid interaction between the work packages.

The technical content of the document provided the current state of knowledge on: (a) the current state of practice in terms of dependable engineering methods, architectural concepts, as well as regulation activities and industrial working groups, (b) relation of TEACHING project requirements to WP3 engineering methods, (c) description and conception of dependability architectures concepts and architecture pattern for different scenarios, (d) detailing of approaches for the application of AI for ensuring of CPSoS dependability, and (e) development of dependability engineering of cloud-connected AI-based systems.

WP3 will continue to elaborate this body of knowledge throughout the remaining project duration and therefore outdate this deliverable by deliverable D3.2 at Milestone 2.

This report depicts the currently established dependability engineering methods and design patterns by WP3 at project milestone 1 and will be elaborated continuously throughout the remaining project duration. Therefore, this deliverable will be amended by deliverable D3.2.

References

[1] TEACHING Consortium, „D5.1 - Initial Use Case Specifications," TEACHING Consortium, 2020.

[2] TEACHING Consortium, „D1.1 - Report on TEACHING related technologies SoA and derived CPSoS requirements," TEACHING Consortium, 2020.

[3] TEACHING Consortium, „D2.1 - State-of-the-art analysis and preliminary requirement specifications for the computing and communication platform," TEACHING Consortium, 2020.

[4] TEACHING Consortium, „D4.1 - Report on first release of the AIaaS system," TEACHING Consortium, 2020.

[5] A. Avižienis, J.-C. Laprie und B. Randell, „Fundamental Concepts of Dependability," Computing Science Newcastle upon Tyne, University of Newcastle upon Tyne, 2001.

[6] D. Gessner, „Doctoral Thesis: "Adding fault tolerance to a flexible real-time Ethernet network for embedded systems"," Universitat de les Illes Balears, Spain, 2017.

[7] A. Avižienis, J.-C. Laprie, B. Randell und C. Landwehr, „Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing 1.1 (2004), 2004.

[8] A. E. Goodloe und L. Pike, „Monitoring Distributed Real-Time Systems: A Survey and Future Directions," National Aeronautics and Space Administration, Hampton, Virginia, 2010.

[9] The SPICE User Group, Automotive SPICE Process Assessment / Reference Model V3.0, VDA, 2015.

[10] ISO - International Standardization Organisation, ISO 26262 Road vehicles - Functional safety, ISO - International Standardization Organisation, 2018.

[11] ISO - International Standardisation Organisation, „ISO 21448 - Road vehicles — Safety of the intended functionality," ISO - International Standardisation Organisation, 2019.

[12] M. O'Brien, . W. Goble, G. Hager und J. Bu, „Dependable Neural Networks for Safety Critical Tasks," Machine Learning (cs.LG); Machine Learning (stat.ML) arXiv:1912.09902 [cs.LG], 2019.

[13] C.-H. Cheng, G. Nührenberg und C.-H. Huang, „Towards Dependability Metrics for Neural Networks," Machine Learning (cs.LG); Machine Learning (stat.ML) arXiv:1806.02338 [cs.LG], 2018.

[14] R. Messnarz, C. Kreiner, G. Macher und A. Walker, „Extending Automotive SPICE 3.0 for the use in ADAS and future self-driving service architectures," Wiley Journal of Softwrae: Evoluation and Process, DOI: 10.1002/smr.1948, 2017.

[15] P. Mishra, V. Varadharajan, U. Tupakula und E. S. Pilli, „A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," IEEE

Communications Surveys & Tutorials, vol. 21, no. 1, pp. 686-728, doi: 10.1109/COMST.2018.2847722, 2019.

[16] Ö. A. Aslan und R. Samet, „A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249-6271, 2020, doi: 10.1109/ACCESS.2019.2963724, 2019.

[17] S. M. Ghaffarian und H. R. Shahriari, „Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey," ACM Comput. Surv. 50, 4, Article 56 (November 2017), 36 pages, 2017.

[18] T. Pourhabibi, K. L. Ong, B. H. Kam und Y. L. Boo, „Fraud detection: A systematic literature review of graph-based anomaly detection approaches," Decision Support Systems, vol. 133, 2020.

[19] J. O. Sinayobye, F. Kiwanuka und S. Kaawaase Kyanda, „A State-of-the-Art Review of Machine Learning Techniques for Fraud Detection Research," IEEE/ACM Symposium on Software Engineering in Africa (SEiA), Gothenburg, 2018.

[20] L. Xiao, X. Wan, X. Lu, Y. Zhang und D. Wu, „IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," IEEE Signal Processing Magazine, vol. 35, no. 5, 2018.

[21] A. Handa, A. Sharma und S. K. Shukla, „Machine learning in cybersecurity: A review," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9, no. 4, 2019.

[22] Z. El-Rewini, K. Sadatsharan, N. Sugunaraj, D. F. Selvaraj, S. J. Plathottam und R. P., „Cybersecurity Attacks in Vehicular Sensors," IEEE Sensors Journal, vol. 20, no. 22, pp. 13752-13767, 2020.

[23] F. Tang, Y. Kawamoto, N. Kato und J. Liu, „Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches," Proceedings of the IEEE, vol. 108, no. 2, pp. 292-307, 2020.

[24] E. Blasch et al. , „Cyber Awareness Trends in Avionics," IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 2019.

[25] M. Gatti und A. Damien, „AI, Connectivity and Cyber-Security in Avionics," 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019.

[26] S. C. Kremer und J. F. Kolen, „Field Guide to Dynamical Recurrent Networks," Whiley-IEEE , 2001.

[27] H. Jaeger und M. Lukosevicius, „Reservoir computing approaches to recurrent neural network training," Computer Science Review, 3(3), 2009.

[28] J. Goh, S. Adepu, M. Tan und Z. S. Lee, „Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks," IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, 2017.

[29] A. Bifet und R. Gavalda, „Learning from time-changing data with adaptive windowing," Proceedings of the 2007 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2007.

[30] Ross, Gordon J. et al., „Exponentially weighted moving average charts for detecting concept drift," Pattern recognition letters 33.2 (2012): 191-198, 2012.

[31] B. Battista, G. R. Fumera und Fabio, „Multiple classifier systems for robust classifier design in adversarial environments," International Journal of Machine Learning and Cybernetics 1.1-4 , 2010.

[32] B. Battista, G. R. Fumera und Fabio, „Adversarial pattern classification using multiple classifiers and randomisation," oint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, 2008.

[33] Rebuffi Sylvestre-Alvise et al., „iCaRL: Incremental classifier and representation learning," Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017.

[34] Z. Li und D. Hoiem, „Learning without forgetting," IEEE transactions on pattern analysis and machine intelligence 40.12: 2935-2947, 2017.

[35] Tramèr Florian et al., „Ensemble adversarial training: Attacks and defenses," arXiv preprint arXiv:1705.07204, 2017.

[36] Papernot Nicolas et al., „Distillation as a defense to adversarial perturbations against deep neural networks," 2016 IEEE Symposium on Security and Privacy (SP), 2016.

[37] Kirkpatrick James et al., „Overcoming catastrophic forgetting in neural networks," Proceedings of the national academy of sciences 114.13 (2017): 3521-3526, 2017.