SAFETY ASSURANCE FRAMEWORK FOR CONNECTED, AUTOMATED MOBILITY SYSTEMS

D3.2

Report on Requirements on Scenario Concepts, Parameters and Attributes

Project short name
SUNRISE

Project full name
Safety assUraNce fRamework for connected, automated mobIlity SystEms

Horizon Research and Innovation Actions | Project No. 101069573
Call HORIZON-CL5-2021-D6-01

Funded by
the European Union

ccam-sunrise-project.eu/

Dissemination level	Public (PU) - fully open
Work package	WP3: CCAM V&V Methodology for Safety Assurance
Deliverable number	D3.2: Report on Requirements on Scenario Concepts, Parameters and Attributes
Deliverable responsible	Erwin de Gelder, TNO
Status - Version	Final – V1.0
Submission date	20/03/2024
Keywords	Scenario, Safety Assurance, Requirements, Interfaces, CCAM system

Authors

Role	Name
Main author:	Erwin de Gelder (TNO)
Contributing authors:	Emre Kaynar (AVL), Raul Ferreira (CAF), Ilias Panagiotopoulos (ICCS), Daniel Becker, Jobst Beckmann (ika), Martin Skoglund (RISE), Edwin van Hassel (SISW), Sjef van Montfort (TNO), Fouad Hadj Selem, Ghada Ben Nejma (Vedecom), Marcos Nieto Doncel (Vicomtech), Antonio Bruto da Costa, Jason Zhang (WMG), Martin Kirchengast, Patrick Weißensteiner (VIF)

Quality Control

	Name	Organisation	Date
Peer review 1	Jorge Lorente Mallada	Toyota Motor Europe	02/02/2024
Peer review 2	Stefan de Vries (project coordinator)	IDIADA	02/02/2024
Peer review 3	Jobst Beckmann (WP3 lead)	ika	04/02/2024
Peer review 4	Stefan de Vries (project coordinator)	IDIADA	18/02/2024

Version history

Version	Date	Author	Summary of changes
0.1	29/12/2023	All authors	Initial version, ready for internal review
0.2	26/01/2024	All authors	Version ready for external review

0.3	14/03/2024	All authors	Resolved comments from peer reviews 1 - 3.
1.0	19/03/2024	Erwin de Gelder	Resolved comments from peer review 4. Version ready for submission.

Legal disclaimer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
AD	Automated Driving
ADS	Automated Driving System
AEB	Autonomous Emergency Braking
CCAM	Connected, Cooperative, and Automated Mobility
COTSATO	COncretizing Test Scenarios and Associating Test Objectives
ISMR	In-Service Monitoring and Reporting
NATM	New Assessment/Test Method for Automated Driving
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
PDF	Probability Density Function
SAF	Safety Assurance Framework
SCDB	SCenario DataBase
SUNRISE	Safety assUraNce fRamework for connected, automated mobIlity SystEms
SUT	System Under Test
UC	Use Case
V&V	Verification and Validation

EXECUTIVE SUMMARY

The safety assurance of Connected, Cooperative, and Automated Mobility (CCAM) systems is crucial for their successful adoption. The **S**afety ass**U**ra**N**ce f**R**amework for connected, automated mob**I**lity **S**yst**E**ms (SUNRISE) project develops a Safety Assurance Framework (SAF) that enables the safety assurance of CCAM systems. Due to the infeasibility and impracticality of assuring safety solely through test drives, scenario-based testing forms a substantial part of the SAF. This deliverable contributes to the development of the SAF and the scenario-based methodology that is part of the SAF in four different ways:

1. The first contribution is a list of requirements related to the **concept of a scenario** within the context of the SAF. While the requirements state that the scenario concept should be broad enough to consider both abstract scenario descriptions as well as concrete scenario descriptions, the requirements strive for an unambiguous description of a scenario. Furthermore, the scenario concept should not limit the inclusion of relevant attributes, such as different types of actors, different environments, and different environmental conditions. Other requirements focus on the consistent formulation of scenarios, such that multiple stakeholders “talk the same language”.

2. It has been proven to be useful to parameterize scenarios, such that different scenarios can be created by only altering the values of the **scenario parameters**. Thus, the concept of scenario parameters is widely adopted. To promote consistent usage of scenario parameters, the second contribution of this deliverable is a set of requirements related to scenario parameters. The requirements for the scenario parameters describe what information should be provided, such as a clear description of the meaning of the scenario parameters. Other requirements are focussing on the consistent usage of scenario parameters.

3. To describe the range of values that - possibly dependent - scenario parameters can take, **parameter spaces** are defined. The third contribution consists of requirements related to contents and properties of such parameter spaces.

4. The fourth contribution is the definition and description of the **interfaces between the different components of the SAF**, since they are closely related to the terminology of the scenario concept, scenario parametrization and parameter space. The definition of these interfaces should foster the collaboration between different stakeholders that are responsible for the different components, as the interface definitions provide clarity on the required information.

The development of the SAF itself is out of scope of this deliverable, but the requirements and interface definitions presented in this deliverable provide a substantial step towards the creation of the SAF. As such, the requirements provide valuable input to the development of the scenario concept and the interface definitions that are a major part of the SAF. As a result, this deliverable contributes to defining the safety assurance of CCAM systems, thus aiding to the actual deployment of these systems.

1 INTRODUCTION

1.1 Project intro

Safety assurance of Connected, Cooperative, and Automated Mobility (CCAM) systems is a crucial factor for their successful adoption in society, yet it remains a significant challenge.

CCAM systems need to demonstrate reliability in all driving scenarios, requiring robust safety argumentation. It is already acknowledged that for higher levels of automation, the validation of these systems by means of real test-drives would be infeasible. In consequence, a carefully designed mixture of physical and virtual testing has emerged as a promising approach, with the virtual part bearing more significant weight in this mixture for cost efficiency reasons.

Several worldwide initiatives have started to develop test and assessment methods for Automated Driving (AD) functions. These initiatives have already moved from conventional validation to a scenario-based approach and combine different test instances (physical and virtual testing) to avoid the million-mile issue.

The initiatives mentioned above provide new approaches to CCAM validation, and many expert groups formed by different stakeholders are already working on CCAM systems' testing and quality assurance. Nevertheless, the fact that there is a lack of a common European validation framework and homogeneity regarding validation procedures to ensure safety of these complex systems, hampers the safe and large-scale deployment of CCAM solutions. In this landscape, the role of standards is paramount in establishing common ground and providing technical guidance. However, standardising the whole pipeline of CCAM validation and assurance is in its infancy, as many of the standards are under development or have been very recently published and still need time to be synchronised and established as common practice.

Scenario Databases (SCDBs) are another issue tackled by several initiatives and projects, that generally tends to silo solutions. A clear concrete approach should be used (at least at the European level), dealing with scenarios of any possible variations, including the creation, editing, parameterisation, storing, exporting, importing, etc. in a universally agreed manner.

Furthermore, validation methods and testing procedures still lack appropriate safety assessment criteria to build a robust safety case. These must be set and be valid for the whole parameter space of scenarios. Another level of complexity is added, due to regional differences in traffic rules, signs, actors, and situations.

Evolving from the achievements obtained in HEADSTART and taking other initiatives as a baseline, it becomes necessary to move to the next level in the concrete specification and demonstration of a commonly accepted **Safety Assurance Framework** (**SAF**) for the safety validation of CCAM systems, including a broad portfolio of Use Cases (UCs) and comprehensive test and validation tools. This will be done in **SUNRISE**, which stands for **S**afety ass**U**ra**N**ce f**R**amework for connected, automated mob**I**lity **S**yst**E**ms.

The SAF is the main element to be developed in the SUNRISE project. As the following figure indicates, it takes a central role, fulfilling the needs of different automotive stakeholders that all have their own interests in using it.

Figure 1: SAF stakeholders

The **overall objective** of the SUNRISE project is to accelerate the safe deployment of innovative CCAM technologies and systems for passengers and goods by creating demonstrable and positive impact towards safety, specifically the EU's long-term goal of moving close to zero fatalities and serious injuries by 2050 (Vision Zero), and the resilience of (road) transport systems. The project aims to achieve this by creating and sharing a European federated database framework centralising detailed scenarios for testing of CCAM functions and systems in a multitude of relevant test cases, based on a harmonised simulation and test environment with standardised, open interfaces and quality-controlled data exchange.

Following a common approach will be crucial for present and future activities regarding the testing and validation of CCAM systems, allowing to obtain results in a standardised way, to improve analysis and comparability, hence maximising the societal impact of the introduction of CCAM systems.

Figure 2 shows the general overview of the SUNRISE project.

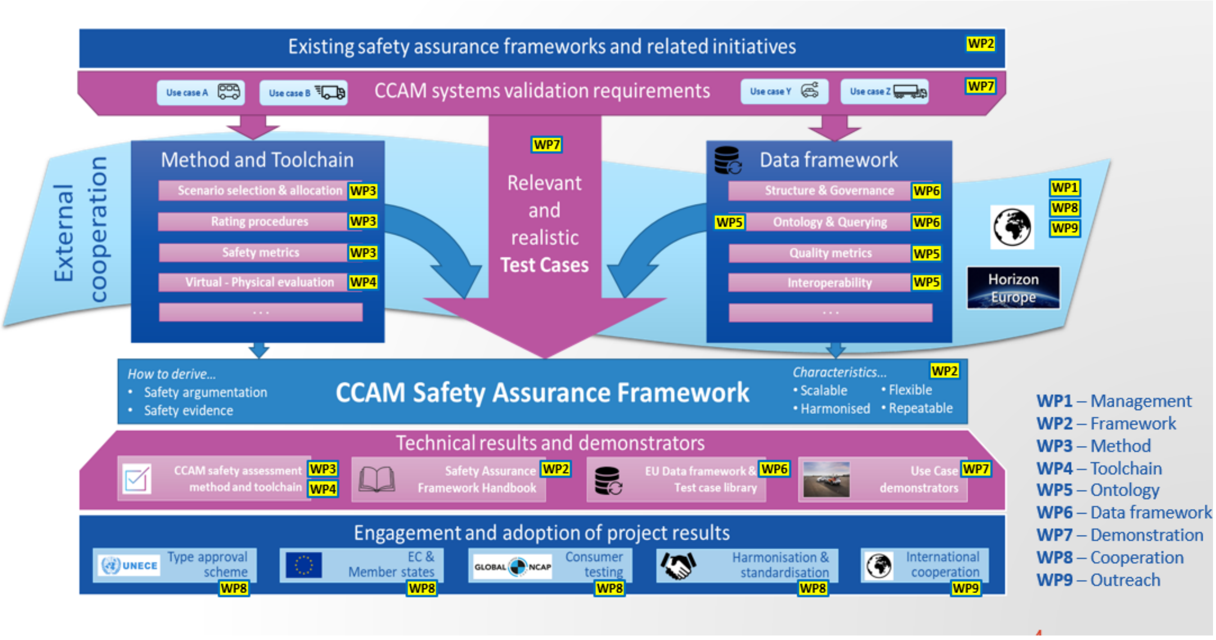

Figure 2: Overview of the SUNRISE Project

1.2 Purpose of the deliverable

Work Package 3's part in SUNRISE is to define and condense an overall CCAM Verification and Validation (V&V) methodology to support the safety argumentation based on data- and knowledge-driven, scenario-based testing while the overall responsibility for the development of the SAF is handled by Work Package 2 "CCAM safety assurance framework".

The purpose of this deliverable is to support the development of the SAF by setting requirements to three constituents that play a key role in the methodology component of the SAF. These three constituents are:

- The scenario concept, which forms the backbone for the descriptions of scenarios that are used within the SAF;
- The scenario parameters, which are an important attribute of the scenarios; and are of relevance in the parametrisation step of the SAF.
- The parameter spaces, which define the possible values that the scenario parameters can take.

In addition, this deliverable defines the interfaces between the components of the SAF. A description of each component is provided, and it is described what information is needed at each of the interfaces. If possible, some information on how this information could be represented is mentioned.

When developing the scenario concept, describing the scenario parameters and parameter spaces, and developing the SAF, the presented requirements should be fulfilled. Only in case of good reasons, e.g., because a change in the SAF (note that at the time of writing, the SAF is not finalized) renders a given requirement obsolete, one may choose not to comply with a requirement. In this case, an appropriate reason needs to be provided.

In addition to the requirements, this deliverable also provides an outlook on the consequences of the requirements on the scenario concept, scenario parameters, parameter spaces, and the interfaces of the SAF components.

The partner contributions to this deliverable are summarized in Table 1.

Table 1: Partner contribution to D3.2.

Partner	Role
TNO	Leading the task and contributing to all requirements.
AVL	Contributing for defining scenario concept, parameters, and spaces as well as requirements for SAF interfaces which are inside the scope of Task 3.2
Continental France	Contributions for defining scenario concept properties, and concept functionalities.
ICCS	Contribute to the definition and discussion of all requirement clusters. Supporting the Task as Work Package 7 leader.
ika	Contribute to the definition and discussion of all requirement clusters. Supporting the Task as Work Package 3 leader.
RISE	Contribute to scenario parameters requirements, scenario concept requirements and process interface requirements.
Siemens NL	Contribute to the definition and refinement of requirements for the scenario concept and scenario coverage.
Vedecom	Contribute to requirements descriptions on parameters space property and the alignment with the Task 6.1.
Vicomtech	Contribute to requirement descriptions on scenario parameters, and verify these requirements are aligned with expected functionalities of the SUNRISE Data Framework (under development in WP6)

Virtual Vehicle Research	Contribute to the definition of requirements for scenario concept, scenario parameters, and scenario parameter spaces.
University of Warwick	Contributing to all requirements, requirement refinement, requirement descriptions, scenario concepts, and the safety-assurance framework (alignment with Task 2.2).

1.3 Intended audience

This deliverable serves multiple stakeholders. The main stakeholder is the SUNRISE project itself, as this deliverable presents the requirements for the scenario concept that underlies the methodology component of the SAF that will be developed in the SUNRISE project, including requirements for the scenario parameter and parameter spaces that are associated with the scenario concept. In addition, the requirements for the interfaces of the SAF components contribute to the development of the SAF. This deliverable can be used as input for the Tasks 2.2 (Definition of a scalable, comprehensive, harmonised safety assurance framework), 3.3 (Scenario selection and subspace creation methodology), and 5.2 (Harmonisation of data framework and SCDB content).

Other stakeholders for which this deliverable is intended are the SCDB owners. The SCDBs are a crucial part of the SAF. This document provides requirements for the scenario concept. It is important that the scenarios from the SCDBs that are used within the SUNRISE SAF, the scenario parameters that are part of the scenarios, and the parameter spaces that define the possible values the scenario parameter can take are not conflicting with the requirements presented in this deliverable. Moreover, some interfaces that are defined in this deliverable directly concern the data framework which the SCDBs are an integral part of.

Finally, this deliverable presents the interfaces between the main components of the SAF, that is being developed as part of Task 2.2 (Definition of a scalable, comprehensive, Harmonised safety assurance framework) of the SUNRISE project. As such, this deliverable is relevant for each stakeholder that plays a role in the CCAM systems' safety assurance following the SUNRISE SAF.

1.4 Structure of the deliverable and its relation with other work packages/deliverables

This deliverable is structured as follows. Chapter 2 provides the necessary background and context of this deliverable. The approach that has led to the requirements and the SAF interface definitions listed later in this deliverable is presented in Chapter 3. The requirements for the scenario concept, the scenario parameters, and the parameter spaces are listed in Chapters 4, 5, and 6, respectively. Chapter 7 presents the interface definitions of the main SAF components. Before concluding this deliverable in Chapter 9, implications of the listed requirements and SAF interface definitions are discussed in Chapter 8.

The following deliverables are related to this deliverable (D3.2):

- D2.3: Final SUNRISE safety assurance framework. This is the deliverable from Task 2.2 and describes the SAF that has been discussed earlier in this chapter.

- D3.1: Report on baseline analysis of existing Methodology. D3.1 is a result from Task 3.1 and contains a description of the state-of-the-art assessment methodologies at the start of the SUNRISE project. A substantial part of the requirements listed in this deliverable originate from the current state of the art listed in D3.1.

- D3.4: Report on Subspace Creation Methodology. D3.4 is the deliverable from Task 3.3. The subspace creation is an important component of the SAF and, consequently, this deliverable is related to D3.4.

- D5.1: Requirements for CCAM safety assessment data framework content. Since scenarios are an integral part of the data framework, some of the requirements listed in D5.1 also apply for the scenario concept that is considered in D3.2.

- D5.2: Harmonised descriptions for content of CCAM safety assessment data framework. D5.2 is a result from Task 5.2 and contains a description of the scenario concept for which this deliverable provides requirements.

- D6.2: Define and development of SCDB input and output standards and interfaces. The requirements listed in D3.2 may address the input and output formats. In addition, the interfaces that are developed as part of D6.2 are defined in D3.2.

2 BACKGROUND

This chapter provides the background information that lays out the context of this deliverable. First, the meaning of terms scenario, scenario concept, scenario parameter, parameter space, and interface are explained in Section 2.1. Next, the UCs that are considered in the SUNRISE project are shortly discussed in Section 2.2. Section 2.3 explains the MS5 version of the SAF for which this deliverable describes the most important interfaces.

2.1 Terminology

In the realm of scenario-based safety assurance for Automated Driving System (ADS), the term "scenario" refers to a description of a temporal and spatial traffic constellation. It includes the specific set of conditions or events that an ADS may encounter during its operation. The term "scenario" can be further subdivided into concrete, logical, abstract, and functional scenarios. A functional scenario provides a conceptual description in natural language, without any fixed parameter values. An abstract scenario formalizes the conceptual nature of the functional scenario, translating ideas into a structured form. A logical scenario defines various parameters that can assume different values, thus forming parameter spaces. A concrete scenario is described by specifying every parameter explicitly, offering a precise account of the scenario.

A “scenario concept” is a specific systematic from which a group of concrete, logical, abstract, or functional scenarios can be derived. The main benefit of a scenario concept is the ability to provide coverage for an ODD using only a finite number of scenarios. Within a scenario concept, parameter spaces are consistent, allowing for a common methodology to be applied. An example of an application of a scenario concept is the challenger concept [1] where the goal is to describe all challenging interactions on the highway. This is realized by defining a group of nine scenarios.

A further distinction is made between a “scenario” and a “test scenario”. Following ISO 34501 [2], a test scenario is a “scenario intended for testing and assessment of ADSs”. Thus, a test scenario can be regarded as a special type of a scenario. Therefore, a “test scenario” is always a “scenario”, but the converse is not necessarily true.

A “test case” contains a concrete “test scenario” and additional information that is needed for execution of the test and the assessment and evaluation of the results of the “test”. Here, “test” is a broader concept that represents the entire testing process. That is, “test cases” serve as the building blocks that contribute to the successful execution of a “test”.

"Scenario parameters" denote the variables within logical and concrete scenarios that can include environmental conditions, traffic conditions, road geometry, vehicle parameters and infrastructural elements. Examples are:

- The initial speed of the ego vehicle [m/s]
- Precipitation [mm/h]

- Road curvature [1/m]

The scenario parameters are used to change the characteristics of a logical scenario. When fixing all the parameters in a scenario one obtains a concrete scenario.

The "parameter space" encapsulates the entire range and combinations of these scenario parameters, forming a comprehensive spectrum of possible circumstances an ADS might face.

An "interface" is defined as the shared boundary across which two or more separate components meet. Within the SAF, the interfaces of the different components serve as the connection point where, e.g., the scenarios, their parameters, and their corresponding data are communicated, aiding in the development, testing, and validation of the ADS's safety systems.

2.2 UCs

One of the objectives of the SUNRISE project is to demonstrate the SAF in a representative set of UCs to prove the robustness, repeatability, and versatility of the SAF when it is applied to different real-world and virtual-testing environments, by using the V&V toolchain developed in WP4.

In the deliverable D7.1, four heterogeneous CCAM UCs with different automation levels and types of ODD in various mixed traffic situations have been defined, as follows:

- UC 1: urban AD perception validation;
- UC 2: traffic jam AD validation;
- UC 3: highway (co-operative) AD validation; and
- UC 4: freight vehicle automated parking validation

For the UCs, a set of high-level validation requirements for testing a broad range of ADSs, covering both their functional and non-functional aspects, has been defined.

The main objectives of UCs are not to develop flawless technological functions ready for the streets. Rather, the aim is to investigate and develop effective and efficient methods for third-party assessment of assurance cases based on evidence gathered through a scenario-based testing approach relevant to the examined ADSs. Further investigation topics within the UCs explore how evidence provided by tests can be confined within the area of validity for the claims they are intended to support. Also, safety metrics related to test cases, parameter spaces and the ODDs of interest and should be defined within the UCs.

Furthermore, advancement within the assessment and audit area is essential for ensuring the ADS meets the necessary safety standards and can be certified for use on public roads. Thus, the purpose of developing UCs is to create sample use-cases that can be used as a basis for drawing broader conclusions about the assessment procedure's effectiveness. The above UCs serve as a tool for validating the SUNRISE SAF, by refining the assessment procedure to ensure that they can efficiently and effectively provide evidence of the ADS's safety, based on the novel scenario-based testing approach.

2.3 SAF

The SUNRISE SAF is a harmonised structure of processes and procedures that allows stakeholders to determine whether a CCAM system meets a set level of safety, for public road introduction and during deployment.

The SAF is based on the multi-pillar approach from UNECE's New Assessment/Test Method for Automated Driving (NATM) [3]. The SAF includes an audit of the safety assessment process, the safety management system as followed by the CCAM manufacturer, and the used test methods and test tools. The framework adopts a scenario-based approach, where tests (virtual testing, physical testing, and a combination of both) are based on scenarios taking into account the CCAM system's ODD for assessing the safety of the CCAM systems under test. Additionally, it provides procedures to monitor that the CCAM's safety is maintained during the system's lifecycle. The scenarios that feed into the SAF are knowledge based and/or data driven.

The SAF includes, but is not limited to, processes to:

- generate the relevant scenarios,
- query relevant test scenarios,
- derive test cases,
- allocate test cases to the different test methods,
- execute the test cases,
- assess and analyse the test results to come to a statement about safety assurance, and
- monitor the CCAM system and provide information of the system to the stakeholders to ensure safety during deployment.

3 APPROACH

The approach that has led to the requirements listed in the following chapters can roughly be divided into three phases. The first phase was the collection of the inputs from the partners that were involved in Task 3.2. To ensure that the different views on the (use of) scenarios and the SAF are represented, a diverse group of partners is responsible for this task, ranging from academia, research institutes, and industry. This first phase led to a large list of requirements that were possibly unclear, inconsistent, or even conflicting. The main goal in this phase, however, was to collect a diverse set of requirements, rather than a final set of requirements that are consistent and without any conflict.

The second phase consisted of discussions among the different partners that were involved in this task. The objective of this phase was to come to a consensus regarding the requirements, to rephrase the requirements so that they are clear for all partners, and to resolve any inconsistencies and conflicts among the requirements. Additionally, in this phase, the requirements were grouped into clusters. For an overview, see Figure 3.

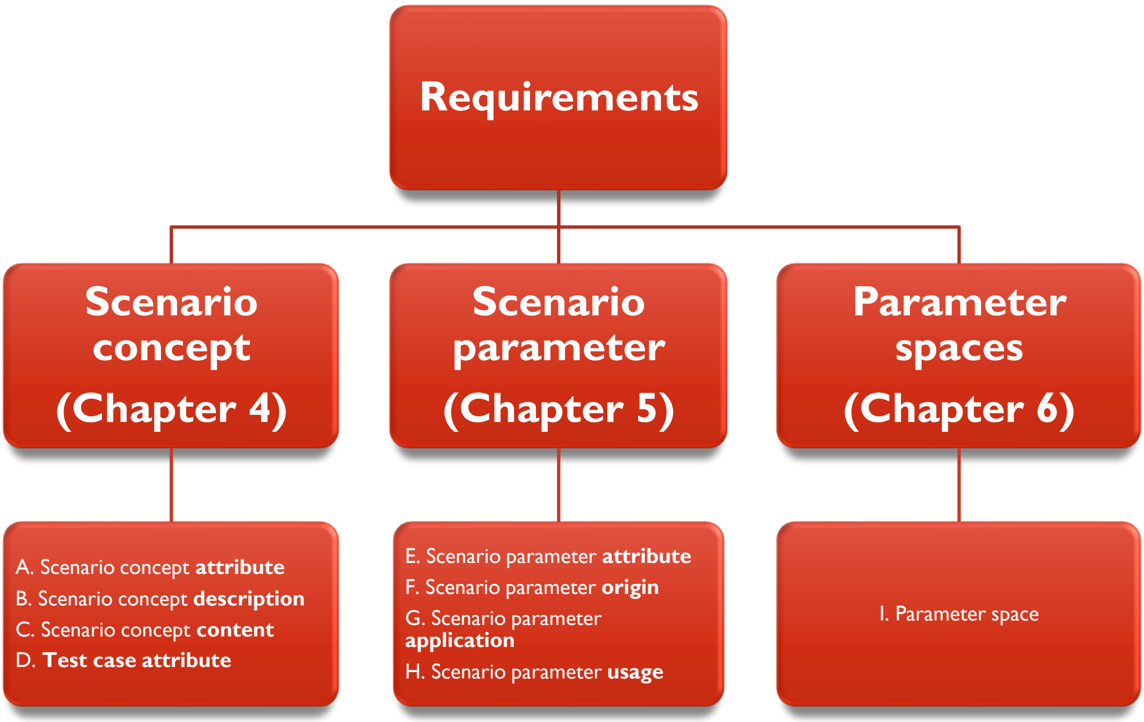

Figure 3: Overview of the clusters of the requirements.

In the third phase, for each of the requirements from the second phase, a description has been added. This description may contain some additional background and context information, as well as a rationale for the reason the requirement has been added. This description – after various review rounds – is presented in this deliverable.

The approach that has led to interface definitions of the SAF is different because at the start of the project, there was no SAF available yet, as this was the main objective of Task 2.2. Initially, a similar approach has been adopted. A high-level description of the SAF became available, however, with only few months remaining for this task. This resulted in some additional interfaces for which no input had been collected in the first phase of the approach. In these cases, a proposal for the interface definitions had been made by one of the partners, after which the other partners could review this. Updates were made based on the reviews and this approach iterated until no more comments were left.

A close communication was established with Task 2.2, so the latest status of the SAF is well reflected in this deliverable.

4 REQUIREMENTS FOR SCENARIO CONCEPT

The requirements for the scenario concept are grouped into four clusters. First, these clusters are described in Section 4.1. Sections 4.2 to 4.5 present the requirements belonging to these clusters.

4.1 Requirement clusters for scenario concept

The requirements presented in this chapter are grouped into four clusters. These are the six clusters:

A. *Scenario concept attribute:* The first cluster contains requirements regarding the attributes of the scenario concept. These requirements are related to the "things" that a scenario concept must "have", just like the two wheels that a bicycle must "have". In this context, an attribute refers to an object, element, or characteristic of the scenario concept.

B. *Scenario concept description:* The second cluster concerns requirements of the description of the scenario concept. These requirements relate to the way the scenarios are represented, e.g., using a specified syntax or formalized language.

C. *Scenario concept content:* The third cluster concerns requirements on the actual content of scenario descriptions. Note that this should not be confused with the way this content is formalized (see cluster B) or how this content is attributed to the scenarios (see cluster A). For example, it is required that the scenario concept contains a scenery (see requirement C.2), but it is not specified how this scenery must be (formally) described or how the scenery is attributed to the scenario concept. The latter may be achieved by having the attribute "scenery" for each scenario, but stakeholder can also use other means to represent the scenery.

D. *Test case attribute:* Strictly speaking, the fourth cluster does not contain requirements for the scenario concept. Instead, it contains requirements for a test case. This has been added to this document as these requirements are relevant for the overall SAF, but do not concern the scenario concept itself. More specifically, these requirements relate to the attributes of a test case.

4.2 Requirements for scenario concept attribute (cluster A)

4.2.1 Requirement A.1

Requirement: A scenario shall not include any metrics.

Description: Scenarios used for testing and validation purposes shall not incorporate specific metrics, including but not limited to criticality or performance metrics. Scenarios should be designed to assess the vehicle's ability to navigate various situations without specifying quantitative measures. This requirement aims to maintain a metric-free approach to scenario definition, allowing for a comprehensive safety evaluation of CCAM systems without bias towards specific performance indicators. Furthermore, this ensures that the respective scenarios can be defined and used across multiple UCs, with different requirements for evaluation.

4.2.2 Requirement A.2

Requirement: A scenario shall have the possibility to include tags.

Description: The scenarios used for CCAM safety assurance shall support the inclusion of tags within their definition. These tags may encompass elements from recognized taxonomies such as ISO 34503 [4], ASAM OpenODD, ASAM OpenLABEL, and ISO 34504 [5]. These tags shall enable a precise scenario categorization, facilitating dedicated matching with the ADS target ODD and behaviour competencies. By incorporating industry-standard tags, interoperability and compatibility with diverse AD environments is enabled, ensuring that the defined scenarios align closely with the intended operational and behavioural parameters of the ADS. This promotes a harmonized and systematic approach to scenario representation, facilitating effective communication and collaboration within the CCAM SAF.
In addition, providing the scenarios with tags improves the searching or querying of SCDBs that are connected to the SUNRISE data framework.

4.2.3 Requirement A.3

Requirement: A scenario shall include information of its source(s). For example, from SCDB or expert knowledge.

Description: Each scenario used within the CCAM SAF shall incorporate information regarding its source(s). This information may originate from sources such as an SCDB or expert knowledge. Including the source attribution ensures transparency and traceability along the safety assurance processes. By documenting whether a scenario is derived from a predefined database or expert input, an effective tracking of the scenario's origin is enabled. Furthermore, it assists in assessing the reliability and relevance of the included scenarios. This requirement enhances the overall accountability of the safety assurance process, allowing stakeholders to understand the basis for scenario selection and promoting a comprehensive understanding of the testing methodology employed.

4.2.4 Requirement A.4

Requirement: A concrete scenario shall define the various entities in the SUT's environment.

Description: The purpose of this requirement is to have a clear overview of the entities in the SUT's environment. These entities could be dynamic entities such as pedestrians, cyclists, and other vehicles, as well as static entities such as the road furniture.

4.3 Requirements for scenario concept description (cluster B)

4.3.1 Requirement B.1

Requirement: The description of a scenario shall be human interpretable.

Description: The scenario descriptions, used for simulation and testing of CCAM systems and ADSs, must be presented in a format that is easily understandable by humans that are experts in the field. This may include plain text descriptions, schematic drawings, videos, or a combination of these that convey the essential aspects of the scenario.

4.3.2 Requirement B.2

Requirement: Scenario concepts should be described on abstract, logical, or concrete level.

Description: The purpose of this requirement is that a SAF shall be able to support all the three levels that a scenario is described. Following ISO 34501 [2], a scenario description at a concrete level refers to a description with explicit parameter values that describe the physical attributes of a scenario. A logical scenario is more general in the sense that it contains the parameter of the scenario, but instead of values, these parameters are defined as ranges, possibly with probability distributions. An abstract scenario description is even more generalized, as this it is a general description of a scenario, i.e., without the scenario parameters being defined. Despite being so general, an abstract description is formalized.

4.3.3 Requirement B.3

Requirement: Logical and concrete scenario descriptions shall support the ability to define and use parameters.

Description: Scenario descriptions within the SAF at a logical or concrete level should be adaptable to dynamic changes through parameterization. This ensures that key elements such as speed, location, weather conditions, and the behaviour of entities could be adjusted. This facilitates the creation of a diverse set of scenarios for testing purposes.

4.3.4 Requirement B.4

Requirement: The scenario concept shall be defined in a standardized format.

Description: The requirement for defining scenario concepts in a standardized format within the SAF is essential to promote consistency, clarity, and efficiency in testing and validating CCAM systems. By adhering to a standardized format, developers and stakeholders can communicate and interpret scenarios uniformly, reducing the risk of misunderstandings and ambiguities in safety-critical contexts. This standardization enhances collaboration and ensures that safety-critical information is clearly articulated, facilitating accurate assessments of the CCAM system's performance. Additionally, a standardized format aids in the repeatability of tests and analyses, making it easier to compare results and track changes over time. This requirement, therefore, contributes to the reliability and effectiveness of the SAF, enhancing the overall safety of CCAM systems.

Note that the scenario concept is the main subject of Task 5.2 of the SUNRISE project, and the standardization is an important topic of this task.

4.3.5 Requirement B.5

Requirement: The description of an abstract, logical, and concrete scenario should have a formal syntax and semantics.

Description: An abstract, logical, and concrete scenario must adhere to a structured format such that it can be interpreted by a computer agent. E.g., this enables the automatic processing of abstract, logical, and concrete scenarios in simulators.

4.3.6 Requirement B.6

Requirement: A concrete scenario description must enable unique interpretation by a computer and human. This may be achieved by conforming to an agreed-on ontology.

Description: The purpose of this requirement is to ensure that different instances of the same scenario are similar on a scenario level. Different instances could be, e.g., physical tests at different test tracks, simulations using different software tools, or tests being set up by different people. This could be achieved by having an ontology which includes all relevant types of information that could affect the outcome of a test using a given SUT, as well as each scenario definition following this ontology by providing all required information, thus leaving no room for different interpretations.

4.4 Requirements for scenario concept content (cluster C)

4.4.1 Requirement C.1

Requirement: The scenario concept shall provide the ability to describe dynamic objects' behaviour.

Description: Within the SAF for CCAM systems, this requirement states that the scenario concept must possess the capability to describe the behaviour of dynamic objects. This can be achieved in various ways, such as detailing the full trajectory of dynamic objects or utilizing a driver behaviour model to simulate their movements. The scenario content should enable the representation of how dynamic objects, like other vehicles or pedestrians, behave within a given scenario. This capability is crucial for evaluating the adaptability and responsiveness of CCAM systems to dynamic elements in diverse driving situations.

4.4.2 Requirement C.2

Requirement: The scenario concept shall include a scenery.

Description: The scenery refers to the "part of the surrounding environment that remains unchanged during a scenario" [2]. Scenery in the scenario concept is necessary for realistic testing of CCAM systems. Within varied environmental conditions, such as different landscapes, junctions lighting and road surfaces, a scenery supports the testing of CCAM systems, such as the sensor performance in various environments. This enhances the validation of autonomous and connected vehicles while closing the gap between real world and computer aided tests.

4.4.3 Requirement C.3

Requirement: It shall be possible to define agent movements in concrete and logical scenarios as manoeuvres and trajectories.

Description: In the context of testing scenarios for ADSs, a key requirement is clear and straightforward: agent movements in both concrete, detailed scenarios and logical representations must be precisely defined as relative manoeuvres and trajectories. This outlines exactly how agents behave, considering their interactions. In general, it emphasizes the importance of accurately describing how agents behave, ensuring a systematic evaluation of how CCAM systems respond in various scenarios, whether detailed or conceptual.

4.4.4 Requirement C.4

Requirement: The scenario concept shall be able to describe V2X communication.

Description: This requirement specifies that the scenario concept must have the capability to describe Vehicle-to-Everything (V2X) communication. When outlining scenarios for testing, it should be possible to include details about how vehicles communicate with each other and with infrastructure elements. This ensures that the SAF accounts for scenarios where vehicles exchange information, contributing to a comprehensive evaluation of CCAM systems

4.4.5 Requirement C.5

Requirement: The abstract, logical, and concrete scenario content must have formal syntax and unambiguous semantics

Description: This requirement points out that regardless of whether the scenario is expressed in a high-level abstract form, a logical representation, or a concrete detailed description, it must follow a structured language. The formal syntax ensures a consistent and clear representation, while unambiguous semantics contribute to a precise understanding of the intended meaning. These characteristics are essential for effective communication and interpretation of scenarios, enhancing the systematic validation and verification processes within the SAF.

4.4.6 Requirement C.6

Requirement: A scenario shall have end conditions (implicit or explicit). If the end conditions are satisfied, the scenario simulation/test is terminated.

Description: This requirement dictates that a scenario must include end conditions where it defines when the scenario shall conclude. If the specified end conditions are met, the scenario shall be terminated. In case of a test scenario, this ensures that the data gathering during test is concluded at the correct moment and ensures that no unnecessary simulation time will be considered.

4.4.7 Requirement C.7

Requirement: Each logical scenario shall specify the valid range for each parameter.

Description: The parameter ranges quantify the magnitudes that are meant to be represented by that parameter in that logical scenario. It is essential to know the units (cf. requirements E.4 and E.5) and the range of each parameter as this defines which concrete scenarios, i.e., a scenario with specific parameter values, are included in the logical scenario and which are not.

4.4.8 Requirement C.8

Requirement: Each logical scenario may specify the probability distribution of the parameters.

Description: Knowing the probability distribution of the scenario parameters is essential to implement the sampling and to document which are the expected values of that parameter in that logical scenario. It also enables the quantification of the likelihood of the combinations of parameter values, thereby enabling the risk quantification [6]. Note that these pdfs may be straightforward parametrized functions but can also be as complex as (deep) neural networks.

4.4.9 Requirement C.9

Requirement: The scenario representation shall allow for an explicit description of unknowns (e.g., if the view onto an area was obstructed).

Description: This requirement mandates that the scenario representation must enable an explicit description of unknowns. For instance, if there is an obstruction in the view of a certain area, the scenario description should explicitly account for this uncertainty. This ensures that the representation of scenarios is comprehensive and realistic, allowing for the explicit inclusion of factors that may be unknown or obstructed during actual operation. Explicitly describing unknowns enhances the accuracy and completeness of scenario-based testing for CCAM systems.

4.4.10 Requirement C.10

Requirement: Scenario content in abstract, logical, and concrete descriptions shall be able to be associated with ODD attributes and behaviour concepts.

Description: This requirement emphasizes the need for scenario content to seamlessly connect with ODD and behaviour concepts across various description levels – abstract, logical, and concrete. This ensures that scenarios are aligned with standard taxonomies like ISO 34503 [4] and OpenLABEL. Such alignment facilitates a structured and standardized approach, enhancing the clarity and coherence of scenario descriptions within the SAF. This association enables a more systematic evaluation of CCAM systems by grounding scenario content in well-defined operational and behavioural contexts.

4.4.11 Requirement C.11

Requirement: A scenario shall be able to describe human factor elements.

Description: The scenarios within the SAF shall consider and represent aspects related to how people behave, act, or interact in driving situations, such as jaywalking, harsh braking, or drowsy driving. This ensures a complete evaluation of how CCAM systems respond to the complexities of human involvement on the road.

4.4.12 Requirement C.12

Requirement: The scenario concept shall include a description of environmental conditions.

Description: Considering environmental conditions like precipitation, lighting, snow, the tests will represent more of the real-world conditions. The environmental conditions take a significant part of an ODD of the CCAM system under consideration. The environmental conditions typically influence the performance of a CCAM system. Hence, it is necessary to assess the performance of a CCAM system under various environmental conditions. To cover all environmental conditions in the scenario concept of the SAF, environmental conditions must be represented in the scenario concept.

4.5 Requirements for test case attribute (cluster D)

4.5.1 Requirement D.1

Requirement: A test case shall include a description of the expected behaviour of the SUT.

Description: To enable the selection of relevant test cases, and to quickly get a first impression of the results of a test, it is required to know what kind of behaviour of the SUT is to be expected because of this test case. For example, when testing an Autonomous Emergency Braking (AEB) system, a test case where the ego vehicle approaches a static or slower car, the expected behaviour is that the AEB system will make the ego vehicle brake to avoid or mitigate the collision.

4.5.2 Requirement D.2

Requirement: A test case should contain an indication on whether the test case is safety critical (e.g., false negative test) or performance critical (e.g., false positive test).

Description: For making a choice on which scenarios / test cases to evaluate in a certain phase of the development of the SUT, it is required to know which aspects can be tested with which scenarios / test cases. For example, when evaluating an AEB system, it is relevant to know whether a test case tests whether the system brakes when it should (safety critical, to avoid false negatives), or whether the system does not brake when it should not (comfort/performance critical, to avoid false positives).

4.5.3 Requirement D.3

Requirement: A test case shall define which outputs are the most relevant (e.g., impact velocity), and, if possible/relevant, give limit values for pass/fail.

Description: To set up a test, it needs to be known what signals are to be recorded. The main set of signals will depend on the test case, and thus should be defined as part of the test case. These signals will also be those that are to be compared to pass/fail/scoring criteria, to evaluate the test result. Wherever possible, e.g., in the case of test cases derived from a protocol (e.g., Euro NCAP), including the pass/fail/scoring criteria in the test case definition will facilitate the assessment.

5 REQUIREMENTS FOR SCENARIO PARAMETER

The requirements for scenario parameter are grouped into four clusters. Section 5.1 explains these four clusters. In the remaining sections of this chapter, the requirements of these four clusters are presented.

5.1 Requirements clusters for scenario parameter

The requirements for the scenario parameters are grouped into the following four clusters:

E. *Scenario parameter attribute:* The first cluster contains requirements regarding the attributes of the scenario parameters. Similar to cluster A in Section 4.2, these requirements are related to "things" that a scenario parameter must "have".

F. *Scenario parameter origin:* The requirement in the second cluster states how scenario parameters should be obtained and/or derived.

G. *Scenario parameter application:* The third cluster of requirements relates to the way the scenario parameter could be applied. Thus, these requirements specify the potential UCs that the scenario parameters should be able to cover.

H. *Scenario parameter usage:* The last requirement cluster for the scenario parameters considers requirements on the usage of scenario parameters. These requirements are presented to promote consistent usage of the scenario parameters.

5.2 Requirements for scenario parameter attribute (cluster E)

5.2.1 Requirement E.1

Requirement: Each scenario parameter shall have a human-readable description that describes the scenario parameter.

Description: Scenario parameters are the variables that define the conditions and events of a scenario. They can include the location, time, weather, traffic, road type, vehicle type, and other factors that affect the behaviour of the SUT and outcome of the scenario. Each scenario parameter shall have a human-readable description that describes the scenario parameter. The description should explain the purpose, range, and unit of the parameter, as well as how it is measured or calculated. The description should also provide examples or references to illustrate the parameter value or meaning. The human-readable description of the scenario parameter helps to ensure the validity, completeness, and consistency of the scenario-based testing for the safety assurance of CCAM systems.

5.2.2 Requirement E.2

Requirement: Each scenario parameter shall have a domain that specifies the permissible set or range of values, where this domain can be discrete, continuous, or a combination of both.

Description: Each parameter has a domain that specifies the permissible values that it can take in a scenario. The domain is important because it defines the scope and the variability of the testing. For instance, the domain of the environment parameter could be {sunny, cloudy, rainy, snowy}, or a numerical range for the temperature or humidity. A discrete domain is a finite or countable set of values, such as {red, green, blue} or {0, 1, 2, ...}. A continuous domain is an interval or a union of intervals of real numbers, such as [0, 100] or (-∞, -10) ∪ (10, ∞). A combination domain is a mix of discrete and continuous values, such as {low, medium, high} ∪ [0.1, 0.9].

5.2.3 Requirement E.3

Requirement: For scenario parameter distributions, it shall be defined the source from which the distributions are derived.

Description: Scenario parameter distributions describe the likelihoods of the values of parameters (such as speed, position, and orientation) that may affect the outcome of a scenario. These distributions can be derived from two sources: real-world data or simulation. Real-world data are collected from actual driving situations, such as traffic cameras, sensors, or surveys. Simulation data are generated by computer models that mimic real-world conditions, such as traffic flow, weather, or road geometry. The source of the scenario parameter distributions has implications for the validity and reliability of the scenario-based testing. Therefore, it shall be clearly defined and documented whether the distributions are derived from real-world data or from simulation, and what are the advantages and limitations of each source.

5.2.4 Requirement E.4

Requirement: A numerical scenario parameter should generally use SI units, but it can also accommodate counting parameters that do not have specific units.

Description: When describing quantities like distances, speeds, or time using numbers, it is essential to use a standard set of units (SI units). The purpose of this requirement is to promote clarity and consistency, minimize errors, and enhance the accuracy and reliability of safety assessments by establishing a common understanding of numerical values across the stakeholders.

5.2.5 Requirement E.5

Requirement: A scenario parameter shall use the same unit for similar parameters within a scenario.

Description: This requirement states that when describing different parameters of a scenario, such as distances, speeds, or time intervals, the units should be the same for similar parameters. For example, if distance is measured in meters for one parameter, it should be consistently measured in meters for all parameters related to distance within that scenario. This ensures uniformity and avoids confusion for developers and stakeholders to understand and work with the scenario parameters.

Note that when complying with requirement E.4, one also complies with this requirement. However, whereas requirement E.4 is a strong recommendation (hence the word “should”), this requirement is mandatory (hence the word “shall”).

5.2.6 Requirement E.6

Requirement: A scenario parameter shall be measurable.

Description: This requirement states that any parameter used as part of a scenario description must be something that can be measured. Defining the characteristic of the scenario, like the distance a vehicle travels or the time it takes to complete an action, it should be something that can be quantified or measured with a specific value. This ensures that the scenario parameters are clear, concrete, and can be objectively assessed during testing.

5.2.7 Requirement E.7

Requirement: Scenario parameters should be either numerical or textual.

Description: This requirement specifies that scenario parameters must fall into one of two categories: numerical or textual. Numerical parameters involve quantifiable values, such as distances, speeds, or time intervals, providing a precise and measurable description. Textual parameters, on the other hand, involve non-numeric descriptors, offering a more qualitative representation of elements like road conditions or human behaviours. This distinction ensures clarity and precision in defining the attributes of scenario parameter properties.

5.2.8 Requirement E.8

Requirement: The scenario parameter value shall be values that can occur in the real world.

Description: The main purpose of a SAF is to ensure that the system is tested against scenarios it might encounter in real-life situations. The scenario parameters, such as speed, distance, or time, should only have values that make sense and are possible in real-world situations.

5.2.9 Requirement E.9

Requirement: Categorical scenario parameters shall be logically ordered.

Description: This requirement states that, when feasible, categorical variables should be carefully ordered within the framework of scenario-based CCAM safety testing. These variables, influenced by factors such as environmental conditions, traffic scenarios, and automation levels, necessitate a deliberate arrangement that mirrors their inherent logic or hierarchy. For instance, categorical parameters like severity levels, degrees of autonomy, or lane positions should be methodically ordered to align with their practical or theoretical significance when possible. This ordering can take the form of a progression from least to most severe, lower to higher autonomy levels, or from the nearest to the farthest lane from the road edge. This logical organization contributes to a more intuitive comprehension of scenarios and enhances the overall clarity and systematic evaluation of the testing process.

5.3 Requirements for scenario parameter origin (cluster F)

5.3.1 Requirement F.1

Requirement: A scenario parameter shall be defined by an ontology, specifying its meaning and unit of measurement.

Description: It is crucial to define a scenario parameter in a way its interpretation is unambiguous for humans and machines. In practice, projects may obtain scenarios from multiple sources often providing heterogeneous definitions of scenario parameters. Therefore, unambiguity and interoperability shall be achieved by precisely defining the meaning and unit of measurement. On the one hand, the unit of measurement shall follow international standards, which already contain the semantics of the values, possible numerical or categorical equivalences and conversions across standards. On the other hand, the meaning of the parameter shall be as well documented to avoid semantic miss-interpretations which may lead to confusion or errors when the parameter is consumed at testing environments. Ontologies can provide a solid basis to hold precise, unambiguous, and linked data representations for units and meaning, as a tool to guarantee descriptions exist, are explicit and can be consumed both by humans and machines.

5.4 Requirements for scenario parameter application (cluster G)

5.4.1 Requirement G.1

Requirement: It shall be possible to cover environmental conditions using scenario parameters.

Description: There is a necessity to have scenario parameters that accurately reflect environmental conditions within the ODD (e.g., ISO 34503 [4]) impacting the performance of the chosen sensors and actuators.

5.4.2 Requirement G.2

Requirement: It shall be possible to cover connectivity using scenario parameters.

Description: Effective connectivity parameter assessment requires evaluating factors that capture and describe the connectivity that is used in the ADS feature and the reliance on these in the ODD.

5.4.3 Requirement G.3

Requirement: It shall be possible to cover the actions and physical properties of the actors of the scenario using scenario parameters.

Description: Scenario parameters must be connected to the characteristics of the actors within the scenario, including aspects like vehicle count, lane widths, starting velocities, and pedestrian positions. In addition, the scenario parameter must be able to capture the condition of systems. For example, in case of sensor degradation (due to aging, wear, or prolonger exposure to heat or water) or obstruction (due to dust, mud, leaves, etc.), it must be possible to capture or quantify that using the scenario parameters. They must also correspond to factors influencing the scenario's progression, encompassing timing and actions taken by the actors, and affecting the safety performance of CCAM systems.

5.4.4 Requirement G.4

Requirement: It shall be possible to cover scenery using scenario parameters.

Description: The scenario parameter should cover various sceneries including various traffic environment components like street furniture and landscapes.

5.5 Requirements for scenario parameter usage (cluster H)

5.5.1 Requirement H.1

Requirement: A naming convention for the scenario parameter shall be agreed on.

Description: The requirement for agreeing on a naming convention for scenario parameters establishes the need for a standardized and consistent approach to naming the various elements used to define scenarios within a SAF. A naming convention ensures that all stakeholders involved in the development and testing processes use a common language when referring to scenario parameters.

5.5.2 Requirement H.2

Requirement: The list of scenario parameters shall be extensible.

Description: It should be possible to expand and add new parameters to the parameter list in a scenario if necessary. This feature will be essential when a new standard or protocol (ISO, NCAP) or even a new technology is released. Then, the extensible scenario parameters can be adapted and enhanced to capture any newly introduced attributes. Note that this is contributing to SUNRISE's first objective, which is to develop a future-proof SAF for a continuously evolving number of CCAM UCs.

6 REQUIREMENTS FOR PARAMETER SPACES (CLUSTER I)

The requirements for parameter spaces are not further grouped into clusters. The following sections contain the requirements and their descriptions.

6.1 Requirement I.1

Requirement: The parameter space shall define the scenario parameters that it addresses.

Description: The parameter space is the set of all possible values that the input parameters of a scenario can take. The parameter space shall define the scenario parameters that it addresses.

6.2 Requirement I.2

Requirement: The parameter space shall allow to define constraints on the (combination of) parameter values.

Description: The parameter space is the set of all possible values that the input parameters of a scenario can take. For example, the speed, position, and orientation of a vehicle are parameters that affect the outcome of a scenario. The parameter space shall allow the tester to define constraints on the (combination of) parameter values, so that only valid and relevant scenarios are generated and executed. For example, a constraint could be that the speed of a vehicle must be positive ($x \geq 0$), or that the distance between two vehicles must be less than their speed difference ($0 < y < x$). Constraints can help to reduce the size of the parameter space and to focus on the critical scenarios that pose the highest risk for safety assurance of CCAM systems.

6.3 Requirement I.3

Requirement: A parameter space can contain a (possibly joint) Probability Density Function (PDF) of the parameters.

Description: The parameter space is a mathematical representation of the possible values and combinations of the scenario parameter values that affect the behaviour and outcome of the CCAM System Under Test (SUT). The parameter space should include a PDF that assigns a probability density to each point or region in the space. The PDF can be based on expert knowledge, historical data, or other sources of information. The PDF is used to guide the sampling strategy for generating test scenarios, such that scenarios with higher probability are more likely to be selected. The pdf does not have to reflect the real-world frequency or likelihood of the scenario parameters, as long as it covers the relevant range and variation of the parameter space.

6.4 Requirement I.4

Requirement: A parameter space shall facilitate rapid determination of equidistant sampling efficacy.

Description: The parameter space must be structured in such a way that allows for the rapid assessment of the effectiveness of equidistant sampling strategies. For the safety assurance of CCAM (Cooperative, Connected, Automated, and Autonomous Mobility) systems, it is crucial to ensure that the distributions of parameters are statistically evaluated and validated, confirming their appropriateness for equidistant sampling methods. This approach, which is similar to Latin Hypercube Sampling, is designed to improve the efficiency and accuracy of scenario-based testing by systematically covering the Operational Design Domain (ODD). This facilitates a thorough yet feasible testing regimen, addressing all pertinent scenarios and parameter combinations without necessitating an excessively large number of tests. Nonetheless, this method depends on precise distribution estimation using data marked by dependable performance metrics such as recall and precision rates. Furthermore, it is essential to effectively manage correlations among dependent variables to preserve the representativeness of the samples.

6.5 Requirement I.5

Requirement: The context of the given parameter space of the given parameter space must be clarified.

Description: The parameter space of a scenario-based testing for CCAM safety assurance is the set of values that define the input variables of the test cases. The parameter space can be influenced by various factors, such as the environment, traffic, communication, automation level, and human factors. Therefore, it is important to clarify the background of the given parameter space, i.e., how it was derived, what assumptions were made, and what limitations were considered. For example, if the parameter space was defined by a protocol, such as a standard or a regulation, then the source and the rationale of the protocol should be explained. If the parameter space was measured in real life, such as by collecting data from field tests or simulations, then the methods and the quality of the data should be described. If the parameter space was based on physical constraints, such as by considering the maximum and minimum values of the scenario parameters, then the validity and the completeness of the range should be justified. For instance, certain constraints might seem obvious at times, and at other times, they are not. Moreover, when scenarios are generated through a distribution or a simulation, and simply due to perception issues, more unusual values can appear. By clarifying the background of the given parameter space, the test engineer can better understand and evaluate the coverage of the scenario-based testing for CCAM safety assurance.

6.6 Requirement I.6

Requirement: A parameter space shall have the possibility to contain a discretisation strategy. The discretisation strategy can be used to derive concrete scenarios from a given scenario spaces (generated from the individual parameter spaces).

Description: The parameter space is a representation of the possible values and ranges of the (multiple) parameters that define a scenario. For example, the parameter space for weather conditions could include values such as sunny, cloudy, rainy, snowy, etc. The discretisation strategy is a method to divide the parameter space into discrete and manageable subsets. For example, the discretisation strategy for weather conditions could group the values into clear, wet, and icy categories. The discretisation strategy can help to reduce the complexity and size of the scenario space, which is the set of all possible scenarios that can be generated from the parameter space. The discretisation strategy can also help to ensure that the scenarios cover the relevant aspects and variations of the parameter space. The requirement states that a parameter space shall have the possibility to contain a discretisation strategy, meaning that it is not mandatory but optional to define one. The requirement also states that the discretisation strategy can be used to derive concrete scenarios from a given scenario space. This means that the discretisation strategy can guide the selection and instantiation of specific values for the parameters that define a scenario. For example, if the discretisation strategy for weather conditions is clear, wet, and icy, then a concrete scenario could be instantiated with a specific value such as sunny for the clear category. The purpose of deriving concrete scenarios from a given scenario space is to perform scenario-based testing, which is a technique to evaluate the safety and performance of CCAM systems under different situations and conditions.

Note that the actual discretisation strategy is an important topic of Task 3.3 of the SUNRISE project.

6.7 Requirement I.7

Requirement: The parameter space shall be bounded.

Description: As previously explained in requirement I.2, by defining clear boundaries and constraints for each parameter, one creates a framework that helps to maintain consistency and feasibility within different scenarios. This approach is fundamental for testing and validating the behaviour of CCAM systems under various conditions. Additionally, it facilitates the comparison of results across different testing environments and scenarios and enable a comprehensive assessment of system performance. Moreover, the parameter space needs to be bounded for practical reasons, while still allowing for the detection of edge or corner cases. In many scenarios, logical reasoning leads to boundaries that signify impossibility beyond a certain limit. For example, in time-to-collision (TTC) calculations, the TTC must be positive, and the ego vehicle's speed should not exceed a certain value (e.g., 300 km/h). Beyond these bounds, methods like Kernel Density Estimation (KDE) or Gaussian Mixture Models (GMM) might generate nonsensical cases. Erroneous perception measurements can also lead to such impossible occurrences. These would be mistaken for legitimate corner cases (as they would be within the parameter range), even though they are physically impossible to achieve. Bounding the parameter space prevents such unrealistic scenarios, ensuring more accurate and meaningful analysis.

6.8 Requirement I.8

Requirement: Both a closed and an open parameter space shall be possible.

Description: This requirement means that the system or framework being described should allow for two types of parameter spaces: closed and open. First, a closed parameter space refers to a scenario where the set of possible parameter values has well-defined boundaries, and this boundary is included within the set itself (e.g., $0 \leq x \leq 1$). In contrast, an open parameter space represents a setting with unclear and undefined boundaries (e.g., $0 < x < 1$), making it uncertain what the minimum and maximum values should be when used for simulation. In open parameter spaces, it is challenging to establish clear boundaries, often allowing for adaptability to evolving conditions or a broader range of potential inputs. The distinction between closed and open parameter spaces is pivotal for configuring systems and algorithms in testing AV systems, enabling them to either operate within clearly defined boundaries or set limit constraints without specifying boundaries values.

6.9 Requirement I.9

Requirement: The parameter space shall be described using well-defined semantics.

Description: Describing the parameter space using well-defined semantics is crucial for ensuring clarity, precision, unambiguity, and consistency in the definition of scenarios and promotes ease of implementation across different tools and testing environments. For this purpose, we need to specify the meaning and interpretation of terms used in defining the parameter space and ensure a shared understanding among stakeholders involved in designing, testing, and validating CCAM systems. Also, we need to describe types of parameter space and to categorize them based on relevant criteria (e.g., continuous, discrete, fixed, variable, dependent, independent). Semantics are often derived from industry standards, regulations, and specifications for CCAM systems. Standards may provide guidelines on how to describe and interpret parameter spaces in a consistent manner.

6.10 Requirement I.10

Requirement: Parameter spaces shall be defined in a way it is possible to apply set operations like intersect and union.

Description: Defining parameter spaces in a way that allows for set operations, such as intersection, union, and difference, is a crucial aspect of scenario design. Set of operations provide flexibility in combining and comparing different parameter spaces. The necessity to define parameter spaces in such a way that set operations like intersection and union can be applied is integral to ensuring that we can effectively select and manage heterogeneous scenarios. By establishing these spaces and variables in a homogeneous manner, we ensure a common ground for these operations. This is not just about the interface or tool we might use later; it is about setting a foundational framework that allows for these scenarios to be compared and combined effectively.

7 DEFINITIONS OF THE SAF INTERFACES

As this chapter defines the interfaces between components of the SAF, first, the different SAF components are described in Section 7.1. Next, the interfaces that are considered in this deliverable are listed in Section 7.2. The remaining sections provide more details regarding the information that is exchanged at the different interfaces.

7.1 SAF components

The SAF, as illustrated in Figure 4, comprises five key components, including the pillars of the multi-pillar approach of NATM [3]: Scenario, Execute, Analyse, In-Service Monitoring and Reporting (ISMR), and Audit. Inputs to this framework are sourced from various entities, including Original Equipment Manufacturers (OEMs), homologation authorities, consumer testing entities, SCDB hosts, research institutes, and others. These inputs consist of ODD and Behaviour Descriptions, External Requirements, and Test Objectives, see the grey block at the bottom of Figure 4. For the sake of brevity, these inputs are collectively called the "SAF inputs” in the remainder of this chapter.

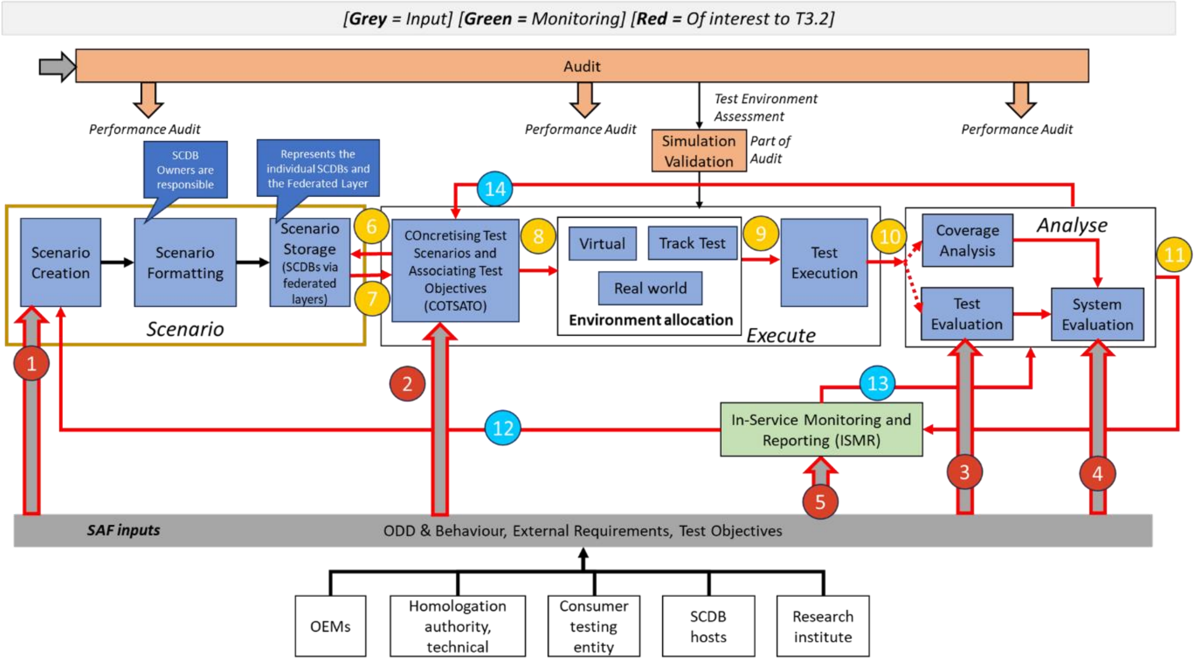

Figure 4: High-level schematic overview of the SAF and the interfaces that are considered in this deliverable.

Key component “Scenario”

In the “Scenario” component, three distinct steps are involved: *Scenario Creation*, *Scenario Formatting*, and *Scenario Storage* on a platform from which the scenarios can be accessed.

Scenario Creation entails acquiring the necessary data and knowledge to create a scenario, while *Scenario Formatting* involves structuring the scenario using, e.g., a scenario description language such as ASAM OpenSCENARIO XML in combination with ASAM OpenDRIVE, or a schema based on the StreetWise domain model [7] or MetaScenario [8]. Finally, *Scenario Storage* entails storing the formatted scenario in a searchable SCDB. This component also includes the SUNRISE federated layer for accessing individual SCDBs. However, those wanting to access a scenario from the SCDB provider may do so directly also.

Key component "Execute"

The "Execute" component encompasses *COncretizing Test Scenarios and Associating Test Objectives (COTSATO)*, *Environment Allocation*, and *Test Execution*.

The process of *COTSATO* involves creating a concrete test scenario from a chosen scenario from the *Scenario Storage* and associating it with relevant test objectives, resulting in one or more test cases. This may require the creation of new concrete scenarios to address coverage concerns or if the SUT failed its pass/fail criteria. *Environment Allocation* pertains to assigning the test scenario in consideration of the scenario content and test objectives to a specific environment platform (virtual, test track, real-world, or combinations thereof) for execution, followed by the actual *Test Execution* on the chosen platform.

Key component "Analyse"

In the "Analyse" component, the results of executed tests are evaluated through *Coverage Analysis*, *Test Evaluation*, and *System Evaluation*.

Coverage Analysis involves assessing test results against coverage goals at both the test suite and logical scenario levels to determine the need for additional testing. *Test Evaluation* involves reviewing results from individual test cases to ascertain the level of success of an individual test, while *System Evaluation* assesses the overall system safety based on a collection of tests cases (which are representative of the ODD for the SUT evaluation).

Key component "ISMR"

The "ISMR" component occurs during SUT operation, with continuous monitoring by the manufacturer and, if applicable, the fleet operator for purposes of continual safety assessment and improvement. The "ISMR" component should also check whether the assessment before the operation is done adequately.

Key component "Audit"

The "Audit" component ensures that the manufacturer has proper processes in place for operational and functional safety throughout the system's development and lifecycle, confirming the safety of the system's design and sufficient validation before market introduction.

7.2 SAF component interfaces

For this deliverable, 14 different interfaces are considered. The "SAF inputs" provide inputs to several components of the SAF (red dots in Figure 4):

1. *Scenario creation* (i.e., first part of the key component “Scenario”);
2. *COTSATO* (i.e., first part of the key component “Execute”);
3. *Test evaluation* (part of the key component “Analyse”);
4. *System evaluation* (part of the key component “Analyse”);
5. *ISMR*

Next, there are multiple interfaces (6 to 11) that follow the process of the SAF, as shown in Figure 4 from left to right (yellow dots in Figure 4):

6. From *COTSATO* to *scenario storage*;
7. From *scenario storage* to *COTSATO, i.e., the reverse direction of interface 6*;
8. From *COTSATO* to *environment allocation*;
9. From *environment allocation* to *test execution*;
10. From *test execution* to *analyse* (both *coverage analysis* and *test evaluation*);
11. From *analyse* to *ISMR*

Finally, there are feedback interfaces (12 to 14), wherein SAF components provide input to previously addressed components (blue dots in Figure 4):

12. From *ISMR* to *scenario creation*;
13. From *ISMR* to the key component *analyse*;
14. From *analyse* to *COTSATO*.

Note that the interfaces are not limited to the ones listed above. For example, there is also an interface from *scenario creation* to *scenario formatting* and from *scenario formatting* to *scenario storage*. These two interfaces, however, are the responsibility of the SCDB owner and, therefore, not further considered in this document. The objective is to harmonize the inputs to the scenario creation and the output from the SCDB storage; how the interfaces in between are handled is up to the respective SCDB owner. Also, interfaces with the audit are not further considered in this document as it is currently unknown what information is required for performing the audit. Further detailing the role of the audit remains future work that could (partially) be addressed in Task 2.2 of the SUNRISE project and may also be considered in future regulatory guidelines such as [3].

7.3 Interfaces 1 to 5: from SAF inputs

As mentioned in Section 7.2, there are five SAF components considered that receive input from the *SAF inputs*. Table 2 shows the information that is exchanged at the different interfaces with *SAF inputs*. In this section, the information that is exchanged at each of the five interfaces is further discussed.

Table 2: Information that is considered at the interfaces from SAF inputs (which contain ODD, behaviour, external requirements, and test objectives) to (1) scenario creation, (2) COTSATO, (3) test evaluation, (4) system evaluation, and (5) ISMR.

Information	**1**	**2**	**3**	**4**	**5**
ODD description	✓	✓	✓	✓	✓
Requirements	✓	✓		✓	✓
SUT	✓	✓			
Variables to be measured during test execution		✓			
Pass/fail criteria for successful test execution		✓			
Monitoring requirements					✓

At all five interfaces, the description of the ODD of the ADS or CCAM system is considered. The ODD description includes ranges of relevant attributes, such as the vehicle speed and rainfall intensity. The ODD description may adopt an inclusive approach, i.e., describing what is in the ODD, an exclusive approach, i.e., describing what is not part of the ODD, or a combination of the two. It would be highly beneficial if the ODD description is standardized and machine readable, e.g., using the JSON file format. For formatting the ODD, it is further suggested to follow the norms related to the ODD definition format listed in ISO 34503 [4]. The *scenario creation* can utilize the ODD description to create scenarios that are part of the described ODD. The COTSATO component uses the ODD description to generate the test scenarios that are needed for the safety assurance of the ADS or CCAM system, including the description of needed output to analyse the results. The ODD description is used by the *test evaluation* to determine whether the ADS or CCAM system operated safely within its ODD in the specific test. The same evaluation is done in *system evaluation*, but then on a system level. The *ISMR* component employs the ODD description to verify whether the system is operating in its ODD.

Next to the ODD description, the *scenario creation*, *COTSATO*, *system evaluation*, and *ISMR* processes also consider the requirements of the ADS or CCAM system. These requirements should reflect the required behavioural competences, (external) regulations, rules of the road, safety objectives, and standard/best practices. The requirements can be a source for creating scenarios, which is why the requirements are part of the first interface. Furthermore, it is important that the process of *COTSATO* considers the requirements, as the goal of the SAF is to assure that the requirements are met. Note that this process also establishes the means to measure compliance with the requirements for the test cases, which are then communicated further in the SAF through the interfaces 8 to 10, see Section 7.5. Not all requirements can be formulated using test validation criteria, which is why the requirements can also be communicated to the *system evaluation*. For example, requirements like "get 5* Euro NCAP", or "fail maximum x% of all cases", is not scenario / test case specific, but only available on the overall level for use in the system evaluation. Lastly, interface 5 needs the requirements to check whether system-level requirements are satisfied over the lifetime of the system. Note that the requirements can be very different from system to system, so formalizing this might be challenging. At least, a standardized format would be preferred.

The SUT is the main subject of the test process, thus this needs to be provided to the *COTSATO* process. The SUT can also be a source for creating scenarios, e.g., scenarios created using knowledge of the system architecture and fault analysis techniques such as systems-theoretic process analysis (also known as STPA). The SUT can be, e.g., a physical prototype or mathematical model of the actual system.

In case some variables need to be measured during the test execution – additional to the variables that are needed to verify the test objectives – this information can be provided to the *COTSATO* process.

Additionally, pass/fail criteria for successful test execution must be available in the *COTSATO* process to identify when tests have or have not successfully been executed. For example, if there are certain tolerances on speed values or lateral path deviations, these shall be included. When the test is executed outside of the pass/fail criteria, then its execution would have to be deemed as unsuccessful from an execution point of view. These examples are related to both real-world and test-track test allocation, but other examples may apply to virtual test environment (e.g., when simulation output contains error)

Through interface 5, monitoring requirements can be provided to the *ISMR*. For instance, these monitoring requirements can relate to the operation during the in-service monitoring or details that must be reported during the in-service monitoring.

For the interfaces 1 to 5, it is assumed that simulation models (other than the SUT) and the simulation platform are part of the “Execute” component and, therefore, not provided externally. Hence, simulation models (other than the SUT) and the simulation platforms are not part of the listed interfaces.

7.4 Interfaces 6 and 7: interfaces between scenario storage and COTSATO

The information exchange between the *scenario storage* and *COTSATO* processes is bidirectional, which is why two interfaces are considered. These interfaces form the federation layer, which is the main subject of Work Package 6 of the SUNRISE project.

Interface 6 considers the input to *scenario storage* from *COTSATO*. This interface contains the query that is used to fetch scenarios from the *scenario storage*. It is the task of the federation layer to process the query, e.g., by passing the query to (some of) the underlying SCDBs. Based on the query and the availability of scenarios that satisfy the query, scenarios may be returned from the *scenario storage*, which are then passed to the *COTSATO* process through interface 7. Note that the scenario concept requirements (Section 4), scenario parameter requirements (Section 5), and parameter space requirements (Section 6) apply to the scenarios exchanged at interface 7. Also note that it is strongly recommended to enable the efficient exchange of multiple scenarios, rather than exchanging all scenarios one by one. The information exchange at the interfaces 6 and 7 is summarized in Table 3.

Table 3: Information that is exchanged at interfaces 6 and 7.

Interface 6: From COTSATO to scenario storage	**Interface 7: From scenario storage to COTSATO**
Query for fetching scenarios from the scenario storage	Scenarios fetched from the scenario storage based on a query

7.5 Interfaces 8 to 11: from COTSATO to ISMR

The information exchange at the interfaces 8 to 10, which are the interfaces from *COTSATO* to *analyse* – with *test environment allocation* and *test execution* in between – contain substantial overlap. Table 4 shows an overview of the information that is exchanged at these interfaces as well as the interface from *analyse* to *ISMR*.

Table 4: Information that is exchanged at interfaces 8 to 11, which are the interfaces (8) from COTSATO to test environment allocation, (9) from test environment allocation to test execution, (10) from test execution to analyse, and (11) from analyse to ISMR.

Information	**8**	**9**	**10**	**11**
SUT	✓	✓		
Test cases: test scenarios, metrics, and validation criteria, and pass/fail criteria for successful test execution	✓	✓	✓	
Metrics	✓	✓	✓	
Validation criteria	✓	✓	✓	
Mapping from requirements to test cases	✓	✓	✓	

Metrics on collection of test scenarios	✓	✓	✓	
Allocated test environment (per test scenario)		✓		
Test execution results			✓	
Overall decision				✓
Assessment result				✓

The SUT is considered at the interfaces 8 and 9, in addition to interface 2 as discussed in Section 7.3. As mentioned earlier, the SUT can be, e.g., a physical prototype or mathematical model of the actual system.

One of the main objectives of the COTSATO is to define the test cases that are used for the safety assurance. Hence, this information is passed to the subsequent SAF components. Note that a test case constitutes of a test objective, the conditions in which the test must be performed (i.e., test scenario), metrics to measure the performance of the SUT in the test, validation criteria that are used to evaluate whether the SUT complies with the requirements and pass/fail criteria to deem whether the test case has been executed successfully. The *test environment allocation* needs this information to decide on the test environment. The test cases are required by the *test execution* to perform the tests. Finally, the test cases are needed for the analysis of the coverage of the tests. Some further notes regarding the test cases:

- It is strongly recommended to implement a harmonized method for formally describing the test scenarios that are part of the test cases. ASAM OpenSCENARIO XML may be a good candidate for this.
- Metrics could include the following:
    - Safety metrics, such as metrics related to the emergency response (e.g., TTC, speed difference), component failure handling, and traffic law compliance;
    - Functional performance metrics that are related to vehicle control, sensor accuracy, and software decision-making;
    - HMI (Human machine Interface) metrics related to perceived safety, interface usability, and system feedback clarity;
    - Metrics for measuring the operational performance, such as the performance in diverse conditions, energy efficiency, maintenance, and interoperability needs;
    - Reliability metrics, such as the system uptime, backup system effectiveness, component longevity, and, according to ISO 8800, AI Robustness, AI Generalization capability, and AI Resilience; and

    - Scenario validation metrics that are used to determine whether the scenario has been executed correctly.
- The requirements of the ADS or CCAM system are also used to define the validation criteria. Most notably, the validation criteria are used by the *Analyse* component to determine whether the ADS or CCAM system complies with the requirements.

In addition to the validation criteria, the mapping of the requirements to the test cases is required. Hence, both the test cases and this mapping are passed from *COTSATO* to *analyse* through the interfaces 8 to 10. The mapping of the requirements to the test cases may be, e.g., a table that specifies which test cases are used for the verification of which requirements.

The last type of information that is passed from *COTSATO* to its subsequent components is information about the collection of test scenarios. The “metrics on collection of test scenarios” include, but are not limited to, representativity (e.g., to what degree are the scenario representing the real world) and source of the test scenarios (e.g., based on what information are these test scenarios generated).

Interface 9 considers all information at interface 8 with the addition of the *allocation test environment(s)*. Note that there may be one test environment allocated to all test cases, but it may also be possible that for each test case, a different test environment is assigned, or even multiple ones. Note that within Task 3.4, methodology for test environment allocation is developed.

Interface 10 considers all information at interface 8, except the SUT information, with the addition of the test execution results. The test execution results include the measurement data, such as vehicle positions, speeds, and accelerations, as well as measures derived from this, such as inter-vehicle distances. It is recommended to use a harmonized means for describing the test execution results. ASAM OSI may be a candidate for describing the test execution results.

The information at interface 11 is different than for the previously-discussed interfaces. First, the overall decision of the analysis of the test execution results is communicated. Based on this decision, the safety assurance can proceed with the in-service monitoring. In case of a positive decision, the assessment results are communicated such that the performance of the ADS or CCAM system during the in-service monitoring can be compared with these results.

7.6 Interfaces 12 to 14: Feedback interfaces

There are three feedback interfaces considered in Figure 4. Note, however, that the SAF may contain more feedback loops during the process of safety assurance. In fact, at each component, it might be required to revisit an earlier process. For example, if the test cases are incompatible with the available test environments, it might be needed to revisit the *COTSATO* component during the *test environment allocation*. In other words, the feedback interfaces are not limited to the ones listed here. To not clutter the SAF of Figure 4 and for the sake of brevity, not all (potential) feedback interfaces are listed here.

Table 5 summarizes the information exchange considered at the interfaces 12 to 14. The *scenario creation* and *coverage analysis* receive observation data during the in-service monitoring. The *scenario creation* could utilize the in-service data to create (new) scenarios, based on the scenarios encountered during the in-service monitoring. This feedback is essential, among others, to capture changes in the traffic system, e.g., due to the introduction of more ADS or CCAM systems. The *coverage analysis* could use the data to revisit the analysis of the coverage based on new insights obtained during the in-service monitoring. This feedback is essential, among others, to account for test scenarios that may have been overlooked during the initial coverage analysis.

Table 5: Information that is exchanged at interfaces 12 to 14, which are the interfaces from ISMR to (12) scenario creation and (13) coverage analysis, and (14) from coverage analysis to COTSATO.

Information	12	13	14
Observed data during in-service monitoring	✓	✓	
Coverage analysis			✓

An open issue is how the feedback from the in-service monitoring is operationalized. In any case, it is recommended that the data complies with the European data strategy and the five related legislative proposals: the Data Governance Act, the Digital Markets Act, the Digital Services Act, the Artificial Intelligence Act, and the Data Act (together known as the “Big Five”). In addition, the General Data Protection Regulation (GDPR) might be considered.

Another open issue is if the feedback for the coverage analysis is done in a direct manner, i.e., through interface 13, or indirectly through interface 12 and the subsequent SAF components. With the alternative approach, new scenarios are added to the data storage and – following the subsequent component of the SAF – might become part of the test scenarios (if deemed relevant) and/or metrics on the collection of test scenarios that are used for the safety assurance.

At interface 14, the results from the coverage analysis are exchanged. This interface accommodates the use of iterative testing, such that new test cases could be generated or selected based on results of earlier tests cases.

8 IMPLICATIONS

This chapter discusses the (possible) implications of the requirements that are presented in this deliverable. First, the requirements for the scenario concept are discussed. Next, Sections 8.2 and 8.3 discuss the implications of the requirements for the scenario parameters and parameter spaces, respectively. This chapter ends with a discussion of the SAF component interfaces.

8.1 Scenario concept

This section aims to discuss the implications of the requirements for the scenario concept which are grouped into four clusters. In the following subsections, a summary and the resulting implications is given for each cluster. Within SUNRISE, the implications mostly concern the development of the scenario concept in Task 5.2 and the communication of scenarios in Task 6.2.

8.1.1 Attributes (cluster A)

This cluster focuses on what elements or objects the scenario concept must possess.

Implications: Utilizing the requirements outlined in the attributes cluster of the scenario concept, we can accurately gather the information needed to define scenarios that emerge from this concept in more detail. The requirements within this cluster address various aspects essential for applying the scenario concept within the SUNRISE SAF framework.

One critical aspect is the prohibition of including any metrics in the scenario definitions, which helps prevent biases from being introduced at the creation stage and subsequently permeating the SUNRISE SAF process. For the SUNRISE project itself, this means that adequate metrics need to be defined within the project to ensure that scenarios, as well as databases containing scenarios and tests being conducted with scenarios, can be compared and quantified.

Another important requirement is the mandate to cite the original source of each scenario, providing insights into how the scenarios were formulated and allowing the filtering of relevant scenarios through the SUNRISE Federated Layer. To facilitate this requirement, it will become necessary to define relevant characteristics of a source and a formalized process of storing this information within scenarios. Additionally, methods for searching and filtering scenarios within databases using this source information will need to be derived.

Lastly, the requirement for a detailed definition of the entities within the SUT's environment ensures that all scenarios within the concept can fulfil their intended roles in the testing process and that crucial information influencing the SUT's behaviour is preserved. This requirement demands for scenario ontology and format to enable the description of any relevant entity within the environment, as well as to capture any important information relevant to such an entity.

8.1.2 Description (cluster B)

Requirements in this cluster pertain to the way scenarios are described to function effectively within the SUNRISE SAF framework.

Implications: For the scenario descriptions, the requirements mandate the use of a standardized format. This necessity arises from several implications. Firstly, the format must be compatible with existing SCDBs utilized within the SUNRISE SAF, ensuring seamless integration. Additionally, it should conform to standards commonly agreed upon by the project's stakeholders. Another implication stems from the need for a formal syntax and semantics to describe scenarios across abstract, logical, and concrete levels. To meet this, the scenario descriptions must adhere to a specifically defined ontology within the SUNRISE project, capable of capturing the nuances of scenarios at varying levels of detail. For the description of logical and concrete scenarios, the format must support parameter definition, requiring these parameters to be articulated within a shared ontology.

Furthermore, the requirement for scenarios to be human interpretable necessitates a precise definition of "human interpretable" based on existing standards within the realm of scenario description. This definition, along with any derived requirements, must be incorporated into the scenario description format selected for the SUNRISE project. Additionally, the SUNRISE Federated Layer must accommodate supplementary materials such as images and videos to enhance the human interpretability of scenarios and enabling the sharing of such data with compatible databases.

Moreover, scenarios must be interpretable by both computers and humans, demanding a format that parsers can easily process. This may involve adopting a widely recognized structure like JSON or XML, for which parsers already exist, or developing a new parser specifically for the SUNRISE project. This requirement similarly applies to the necessary ontology, ensuring that scenarios are accessible and interpretable across the project's various components.

8.1.3 Content (cluster C)

This cluster deals with the actual content within scenario descriptions, distinct from how it is described or attributed.

Implications: The requirements specified in this cluster delineate the types of content that must be supported within a scenario concept for the SUNRISE SAF framework. Scenarios are expected to incorporate dynamic and static elements, environmental conditions, V2X (Vehicle-to-Everything) communications, specifications of unknown factors, and human elements. This necessitates an ontology that encompasses these elements comprehensively, ensuring the format employed can adequately describe them. Additionally, the methodology and the implementation of test frameworks within the SUNRISE SAF must be equipped to manage scenarios incorporating these diverse elements.

Furthermore, these requirements extend to the parameters of each scenario. Every parameter within a logical scenario allows for an established valid parameter range and an associated probability distribution, both of which must be expressible in a unified format. This has significant implications for the SUNRISE methodology, as these parameter ranges and probability distributions form the basis of the parameter spaces analysed by the methodology. It must also factor in the probability distribution of each parameter when segmenting parameter spaces and determining representative parameter sets for testing, ensuring a comprehensive and effective testing process.

8.1.4 Test Case Attributes (cluster D)

Although not directly for the scenario concept, these requirements are crucial for the overall SAF and specifically relate to the attributes of a test case.

Implications: To set up a test, it needs to be known what signals are to be recorded. In addition, the possible inclusion of pass/fail criteria enables the actual evaluation of a CCAM system based on its performance in the test.

8.2 Scenario parameter

Logical and concrete scenario descriptions for the assessment of CCAM systems are parameterized, i.e., parameters are defined that characterize a scenario. Choosing the parameters that describe a scenario is not trivial. By choosing too few parameters might lead to an oversimplification of the actual scenarios, and as a result, not all possible variations of a scenario are modelled. On the other hand, too many parameters lead to problems due to the curse of dimensionality [9]. This section aims to discuss the implications of the requirements for the scenario parameter which are grouped into four clusters. In the following subsections, a summary and the resulting implications are given for each cluster. Within SUNRISE, these implications are mostly relevant for Task 3.3 regarding the parameter space creation methodology and Task 6.2 because the scenario parameters are part of the scenarios that are obtained from the SCDBs.

8.2.1 Attributes (cluster E)

This cluster focuses on what elements or objects the scenario parameter must possess.

Implications: These requirements underscore the need for handling and simulating realistic situations with various parameters when the scenario-based testing is applied for the safety validation of CCAM systems. By incorporating human-readable descriptions for the scenario parameters, specified domains for the permissible set or range of parameter values, and sources of the scenario parameter distributions, these requirements aim to enhance the reliability of the scenario-based testing. The above approach helps to ensure the validity, completeness, and consistency of the scenario-based testing for the safety assurance of CCAM systems within the SUNRISE project.

8.2.2 Origin (cluster F)

The requirements state how scenario parameters should be obtained and/or derived.

Implications: These requirements underscore the need for defining ontologies, by specifying the meaning and unit of measurement for the scenario parameters. Overall, these requirements provide precise, unambiguous, and linked data representations, which can be objectively assessed during the scenario-based testing for the safety assurance of CCAM systems within the SUNRISE project.

8.2.3 Application (cluster G)

The requirements relate to the way the scenario parameters could be used.

Implications: First, scenario parameters shall be accurately reflecting environmental conditions within the ODD. Second, scenario parameters shall cover connectivity issues if the ADS under test has such features. Third, scenario parameters shall be able to connect to the characteristics of the actors and cover various sceneries within the test scenarios. Overall, these requirements ensure the accurate reflection of all the conditions within the ODD that are necessary for the scenario-based testing of the safety assurance of CCAM systems within the SUNRISE project.

8.2.4 Usage (cluster H)

The requirements relate to the usage of scenario parameters.

Implications: First, a naming convention for the scenario parameter shall be agreed on. Second, it is quite important (if necessary) to expand and add new parameters to the parameter list in a scenario. Overall, these requirements ensure that all stakeholders involved in the scenario-based testing of the safety assurance of CCAM systems use a common language when referring to scenario parameters and can be adapted to capture any newly introduced attributes.

8.3 Parameter space (cluster I)

This section discusses implications of the parameter space requirements (see Section 6). In general, the main implications are on the risk to produce ambiguous parameter space definitions when leaving freedom to the user to define the ranges, PDFs, and sampling strategies. Within SUNRISE, the implications are relevant for Task 3.3, which is about the creation of scenario spaces.

Requirement I.2 implies the user can produce constraints formulated as equations. This has several implications:

- The user is responsible to produce constraints that do not result in an empty set.
- The level of complexity of the mathematical constraints is not specified in the requirement and may imply a solver tool to find the valid parameter values.
- Multiple or complex constraints, when combined (e.g., with set operations as intersection as defined in requirement I.10), might yield parameter spaces which are either empty or with lower dimensionality than the full parameter space. That means a constrained-sampling strategy might be needed to produce valid samples.

Requirement I.7 points out that ranges shall be defined with maximum and minimum values, and as open or close boundaries (requirement I.8), whereas Requirement I.9 also considers ranges as discrete sets of enumerables. However, additional considerations for the sampling strategy might be needed if the minimum or maximum values are undefined, or explicitly defined as infinite.

It is important to identify the parameter type of parameters (e.g., natural number, integers, real number, subset of real number, complex number, or higher-dimensional structs), beyond its unit (e.g., m/s for "speed"). In most cases, the unit may already contain the semantics to understand if a parameter can be expressed with a certain parameter type. For instance, if the unit of measurement is "count", "cardinality" or "number of" (e.g., for a parameter "number of lanes"), or even unitless (aka dimensionless) for enumerables (see requirement I.9), then a human may naturally understand this unit is obviously defined as a natural number or a no-unit. However, machines, if not explicitly told about the parameter type, may use floating numbers when averaging or sampling ranges. To avoid such incorrect numerical expressions, parameter types should be either defined within the unit of measurement of the parameter, or implicitly agreed between the parts, implementing tools that truncate, convert, or validate parameter types. Requirement I.6 also addresses this as the need to permit a discretisation strategy, focused on reducing the parameter space into discrete and manageable subsets.

Requirement I.3 opens the door to having PDFs associated to parameters. This way, the "ranges" can be understood as simplification of such pdfs, which just define the so-called support of the pdfs, i.e., the domain where the probability density is nonzero. PDFs may be defined either analytically (with parameterized equations, e.g., normal distribution or exponential distribution), or numerically (with histograms, or sets of samples). One important implication is the potential correlation between the pdfs of different inter-related parameters. From the perspective of probability and sampling, a parameter space is a set of random variables, which might be independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent. Being independent means that the knowledge of the value of one variable gives no information about the value of the others. However, correlation between variables often exists, as different values of certain parameters might have impact on the values of other variables. Such complexities might be addressed using covariance or correlation matrices to represent the linear relationship between pairs of variables.

It is important to recognize that scenarios may be sourced from data or knowledge from a domain expert. When scenarios are informed by domain experts, there is often no associated PDF with the scenario parameters. The focus of testing lies in identifying the most critical scenarios for the safe and correct function of the ADS. To achieve this, optimization techniques like Bayesian Optimization come into play. Unlike relying solely on PDFs, these techniques explore and identify parameter spaces that allow us to understand how the ADS behaves across various conditions critical to the safety or performance criteria defined.

The definition of the parameter space can be seen as a three-step procedure: first, the parameters are defined precisely, with meaning, unit of measurement (optionally with explicit parameter types) and ranges. A second step may add detail to the space by defining constraints between variables, PDFs (with or without covariance matrices) and even discretization strategies to discretize the parameter space. The third step consists of a sampling strategy (requirement I.6) that generates valid samples from the defined parameter space, according to some user-defined sampling mechanism, which may include additional parameters such as number of samples, minimum step between samples (in one of many parameters), etc.

8.4 SAF component interfaces

This section discusses the interfaces between the SAF components. Section 8.4.1 relates to the interfaces 1 to 5 discussed in Section 7.3. The interfaces 6 to 11, presented in Sections 7.4 and 7.5, are discussed in Section 8.4.2. The feedback interfaces 12 to 14 of Section 7.6 are addressed in Section 8.4.3. This chapter ends with a few comments regarding other interfaces (Section 8.4.4) and some open points (Section 8.4.5).

8.4.1 Interfaces 1 to 5

The SAF is heavily dependent on the quality of inputs it receives. These inputs, sourced from a variety of entities such as OEMs, homologation authorities, consumer testing entities, SCDB hosts, research institutes, and others, form the backbone of the SAF. They consist of ODD and Behaviour Descriptions, External Requirements, and Test Objectives.

The accuracy and comprehensiveness of these inputs directly impact the effectiveness of the SAF. For instance, an inaccurate ODD description could lead to the creation of irrelevant or misleading scenarios. Similarly, outdated Behaviour Descriptions or External Requirements could result in tests that do not accurately reflect the current capabilities or constraints of the system. Therefore, it is crucial to have mechanisms in place to ensure the accuracy, relevance, and timeliness of these inputs. It is specifically important that the format used to communicate inputs follows standard formats, for example, and ODD description aligned with the ODD definition format guidelines in ISO 34503 [4].

8.4.2 Interfaces 6 to 11

Interfaces 6 to 11 represent the core operational flow of the SAF, facilitating the exchange of information between different components such as scenario storage, test scenario concretization, environment allocation, test execution, and analysis. The efficiency and accuracy of these interfaces are critical for the overall performance of the SAF.

The use of standard formats is considered important for the effective operation of the SAF.

- In the “Scenario” component of the SAF, it is important to use a scenario description language such as ASAM OpenSCENARIO XML in combination with ASAM OpenDRIVE, or a schema based on the StreetWise domain model [7] or MetaScenario [8] for structuring the scenarios.
- There must be a harmonized means for describing the test case results. ASAM OSI is suggested as a candidate for this purpose.

It is also acknowledged that formalizing these standards might be challenging due to the variability of system requirements. Nonetheless, there is the need for a standardized format to address these requirements. This is captured within Task 2.2 and Task 5.1. Various aspects are addressed within other tasks within the SUNRISE project. For instance, Task 5.2 addresses this with the need for a common ontology and labelling format. Tasks within Work Package 6 address the scenario language exchange formats to be used with the SUNRISE Federated Layer.

Any miscommunication or error in these interfaces could lead to inaccurate test results or missed safety issues. For example, if the scenario storage interface fails to accurately retrieve the required scenarios, the subsequent test execution could be based on incorrect or incomplete test scenarios. Therefore, it is essential to have robust and reliable mechanisms in place for these interfaces, possibly incorporating error-checking and validation steps to ensure the integrity of the information exchange.

8.4.3 Feedback Interfaces 12 - 14

Feedback interfaces 12 to 14 provide a mechanism for continuous improvement in the SAF. They allow for the incorporation of insights gained from in-service monitoring into scenario creation and coverage analysis. This feedback loop is vital for capturing changes in the traffic system and addressing scenarios that may have been overlooked during the initial coverage analysis.

However, operationalizing this feedback remains an open issue and requires careful consideration of data privacy regulations. For instance, how can the SAF ensure that the data collected during in-service monitoring is anonymized and does not infringe on the privacy of individuals? How can this data be securely stored and accessed for the purpose of improving the SAF? These are questions that would be further discussed within Task 2.2, an ongoing task, running through to the end of the project.

8.4.4 Other Interfaces

The “Scenario” component interfaces of the SAF are managed by individual SCDB providers. These interfaces, which include those from scenario creation to scenario formatting and from scenario formatting to scenario storage, are crucial for the operation of the SAF. While the aim is to harmonize the inputs to the scenario creation and the output from the SCDB storage, the handling of the interfaces in between is up to the respective SCDB owner.

Interfaces with the audit component are also not considered in this document due to the current unknowns regarding the information required for performing the audit. This highlights the need for further exploration and standardization in the SAF’s operation. This would be a point of further discussion within Task 2.2, an ongoing task, running through to the end of the project.

8.4.5 Open Points

There are several open points that need further exploration:

- *Standardization of the ODD description:* The ODD description forms a crucial part of the inputs to the SAF. A standardized format for the ODD description could improve the consistency and accuracy of the scenarios created. It is proposed that this format is aligned with the ISO 34503 standard [4].

- *Formalization of system requirements:* System requirements guide the creation of test scenarios and the evaluation of test cases. Formalizing these requirements could help ensure that they are clear, measurable, and relevant.
- *Harmonization of test scenario and test case result descriptions:* A harmonized method for describing test scenarios and test case results could improve the consistency and comparability of tests, making it easier to evaluate and improve the SAF. Note that this is an important objective of Task 5.2.
- *Operationalization of in-service monitoring feedback:* The feedback from in-service monitoring is a valuable resource for improving the SAF. However, it’s unclear how this feedback should be operationalized, especially considering data privacy regulations.
- *Identification of additional feedback interfaces:* The SAF may contain more feedback loops than currently considered. Identifying and examining these potential feedback interfaces could help improve the responsiveness and adaptability of the SAF.

These open points provide a roadmap for future research and development efforts aimed at improving the SAF. Addressing these points could lead to a more effective and reliable SAF, ultimately contributing to the safety of ADSs or CCAM systems. Within the remainder of the SUNRISE project, these open points will be addressed, but it is also expected that continuous research is required to address these.

9 CONCLUSIONS

For the successful adoption of connected, cooperative and automated mobility (CCAM) systems, their safety assurance is crucial. The Safety assUraNce fRamework for connected, automated mobIlity SystEms (SUNRISE) project develops a Safety Assurance Framework (SAF) that enables the safety assurance of CCAM systems. Due to the infeasibility and impracticality of assuring safety solely through test drives, scenario-based testing forms a substantial part of the SAF.

The contribution of this deliverable is to provide requirements for the scenario concept that forms the backbone of the scenario-based approach of the SUNRISE SAF. Requirements are provided for the scenario concept, the scenario parameters, and the parameter spaces. In addition, this deliverable proposes the major interfaces between the different components of the SAF and the information that should be available at these interfaces.

This deliverable presents 22 requirements for the scenario concept, which are grouped into three different clusters. While the requirements state that the concept of scenario should be broad enough to consider both abstract scenario descriptions as well as concrete scenario descriptions, the requirements strive for an unambiguous description of a scenario. Furthermore, the scenario concept should not limit the inclusion of relevant attributes, such as different types of actors, different environments, and different environmental conditions. Note that a clear distinction is made between a scenario and a test case, where the latter includes a (test) scenario as well as test objective and test metrics. To clarify this, a few requirements for test case are listed in this deliverable. These requirements for scenario concept and test case are relevant for all stakeholders involved in the implementation of scenario-based assessment methods for CCAM systems. More specifically, within the SUNRISE project, these requirements are relevant for Task 5.2 and Work Package 6 because these deal with the development of the data framework of which the scenario concept is part of.

It has been proven to be useful to parameterize scenarios, such that different scenarios can be created by only altering the values of the scenario parameters. Thus, scenario parameters are widely adopted. To promote consistent usage of scenario parameters, 16 requirements for scenario parameters are presented, which are also grouped into four clusters. The requirements for the scenario parameters describe what information should be provided, such as a clear description of the meaning of the scenario parameters. Other requirements are focussing on the consistent usage of scenario parameters. These requirements are relevant for stakeholders that use (parametrized) scenarios for the assessment of CCAM systems. Within the SUNRISE project, these requirements are relevant for Task 6.4 when creating parametrized test scenarios and Tasks 7.2 and 7.3 when utilizing parametrized scenarios for demonstration purposes.

To describe the range of values that - possibly dependent - scenario parameters can take, parameter spaces must be defined. For these parameter spaces, requirements are presented in this deliverable, which contain 10 requirements in total. These requirements are relevant for Task 3.3 for the parameter space creation methodology and the Tasks 6.4, 7.2, and 7.3 in case these Tasks utilize logical scenarios.

Additionally, to the requirements of the scenario concept and the associated scenario parameter and parameter spaces, this deliverable describes the **interfaces** between the different components of the SAF. The definition of these interfaces should foster the collaboration between different stakeholders that are responsible for the different components as the interface definitions provide clarity on the required information. Although the development of the SAF itself is part of Task 2.2 and will be described in deliverable 2.3 of the SUNRISE project, the interface definitions provide a substantial step towards the development of the SAF. These interfaces are a valuable input to SUNRISE Task 2.2, which deals with the definition of the SAF. Furthermore, the defined interfaces should be considered during the definition of the data framework (SUNRISE Work Package 6) which materializes some of the interfaces.

The provided requirements and SAF interface definitions form a substantial part of enabling the safety assurance of CCAM systems. However, to fully materialize the SAF, future work is required. One important Task is to turn the requirements into an actual concept that can be used (Tasks 2.2, 6.2, and 6.3 of SUNRISE) and will be adopted by the different stakeholders (e.g., in Tasks 7.2 and 6.3 of SUNRISE). Another Task is to provide clarity on how the information at the interfaces is realized or formalized (e.g., Task 5.2 of SUNRISE).

10 REFERENCES

[1] H. Weber, J. Bock, J. Klimke, C. Roesener, J. Hiller, R. Krajewski, A. Zlocki and L. Eckstein, “A Framework for Definition of Logical Scenarios for Safety Assurance of Automated Driving,” *Traffic Injury Prevention,* vol. 20, pp. S65--S70, 2019.

[2] ISO 34501, *Road Vehicles - Test Scenarios for Automated Driving Systems - Vocabulary,* International Organization for Standardization, 2022.

[3] ECE/TRANS/WP.29/2021/61, “New Assessment/Test Method for Automated Driving (NATM) - Master Document,” World Forum for Harmonization of Vehicle Regulations, 2021.

[4] ISO 34503, *Road Vehicles - Test Scenarios for Automated Driving Systems - Taxonomy for Operational Design Domain for Automated Driving Systems,* International Organization for Standardization, 2023.

[5] ISO 34504, *Road Vehicles - Test Scenarios for Automated Driving Systems - Scenario categorization,* International Organization for Standardization, 2024.

[6] E. de Gelder, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper and B. De Schutter, “Risk Quantification for Automated Driving Systems in Real-World Driving Scenarios,” *IEEE Access,* vol. 9, pp. 168953-168970, 2021.

[7] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg and B. De Schutter, “Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework,” *IEEE Transactions on Intelligent Vehicles,* vol. 7, pp. 300-314, 2022.

[8] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng and L. Li, “MetaScenario: A Framework for Driving Scenario Data Description, Storage and Indexing,” *IEEE Transactions on Intelligent Vehicles,* vol. 8, pp. 1156-1175, 2022.

[9] E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg and B. De Schutter, “Scenario Parameter Generation Method and Scenario Representativeness Metric for Scenario-Based Assessment of Automated Vehicles,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, pp. 18794--18807, 2022.