D4.4

Report on the Harmonised V&V simulation framework

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	WP4: CCAM V&V framework
Deliverable number	D4.4: Report on the Harmonised V&V simulation framework
Deliverable responsible	Bernhard Hillbrand, Virtual Vehicle Research
Status - Version	Final – V1.0
Submission date	26/09/2024
Keywords	Harmonised V&V simulation framework, data formats, standard interfaces, subsystems, modularity

Authors/Contributors

Name	Organisation
Alperen Kiral	SISW
Anastasia Bolovinou	ICCS
Ashfaq Farooqui	RISE
Bruno Celan	AVL
Bernhard Hillbrand	VIF
Elena Daskalaki	ICCS
Gerhard Benedikt Weiss	VIF
Ilias Panagiotopoulos	ICCS
Jobst Beckmann	ika
Jorge Lorente	TME
Jose Torres	CAF
Georg Stettinger	IFAG
Marcos Nieto Doncel	VICOM
Xizhe Zhang	UoW

Quality Control

	Name	Organisation	Date
Peer review 1	Eloy Collado	IDIADA	24/07/2024
Peer review 2	Stefan de Vries	IDIADA	09/09/2024

Version history

Version	Date	Author	Summary of changes
0.1	06/03/2024	Bernhard Hillbrand (VIF)	First draft of document structure
0.2	19/07/2024	All authors	First draft version
0.3	23/09/2024	All authors	Changes after feedback from reviewers
0.4	26/09/2024	Bernhard Hillbrand (VIF)	Changes after the second feedback from the coordinator
1.0	26/09/2024	Bernhard Hillbrand (VIF)	Final version ready for submission

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
API	Application Programming Interface
ASAM	Association for Standardization of Automation and Measuring Systems
CAM	Cooperative Awareness Message
CCAM	Connected, Cooperative, and Automated Mobility
CD/CI	Continuous Integration/Continuous Delivery
CLHS	Constrained Latin Hypercube Sampling
CMP	Credible Modeling Process
CPM	Collective Perception module
CP	Collective Perception
CSP	Credible Simulation Process
DENM	Decentralized Environmental Notification Message
ECA	Electric Connected and Automated
ETSI	European Telecommunications Standards Institute
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GNSS	Global Navigation Satellite System
GP	Gaussian Process
gRPC	Google Remote Procedure Calls
GUI	Graphical User Interface
HiL	Hardware-in-the-Loop
ISMR	In-Service Monitoring and Reporting

iSO	International Organization for Standardization
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LHS	Latin Hypercube Sampling
NATM	New Assessment/Test Method for Automated Driving
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OSI	Open Simulation Interface
OSMP	Operational and Scientific Monitoring Plan
PDF	Probability Density Function
RHP	Reference Homologation Process
RoMPaC	Robust Motion Planning and Control
ROS	Robot Operating System
RSU	Road Side Unit
RTK	Real Time Kinematics
SAF	Safety Assurance Framework
SCDB	SCenario DataBase
SiL	Software-in-the-Loop
SOTIF	Safety Of The Intended Functionality
SSP	System Structure and Parameterization
SUMO	Simulation of Urban MObility
SUNRISE	Safety assUraNce fRamework for connected, automated mobIlity SystEms
SUT	System Under Test
UC	Use Case
V&V	Verification and Validation
V2X	Vehicle-to-Everything communication
XML	Extensible Markup Language

YOLO	You Only Look Once
ZOH	Zero-Order Hold

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 is about the design of a harmonised V&V simulation framework and the question of which aspects can be harmonised. Since virtual simulation has a key role in the testing of CCAM systems, it makes sense to use a framework that provides interoperability, modularity and scalability. By using standardised interfaces and common data formats it is easier to exchange available tools according to the evolving needs and requirements of the test cases.

This document covers two parts. In the first part, a summary of related projects and their approaches towards harmonisation is introduced (see chapter 2) as well as the individual partner designs of their simulation frameworks (see chapter 3). These frameworks have been designed for the SUNRISE use cases and therefore have different purposes. Some are more focused on connectivity, while others centre around the perception system. The simulation frameworks already show the tools selected for the design. However, the choice of tools is not part of this deliverable but has already been described in the document "D4.3 - Report on CCAM simulation tool landscape" [1]. The harmonised V&V simulation framework is tool agnostic. The aim is not to recommend tools, but to design a simulation framework that allows users to easily switch between different tools without having to redesign the framework.

The second part of the deliverable is about the individual aspects that can be harmonised (see chapter 4). The three aspects that were identified are the subsystems, data formats and interfaces. The subsystems recommended for the harmonised V&V simulation framework are just briefly described since there is a deliverable dedicated to this topic (“D4.1 – Report on relevant subsystems to validate CCAM systems” [2]). To recommend data formats and interfaces, the partner simulation framework designs have been analysed. In the end, the design of a harmonised V&V simulation framework is presented with harmonised subsystems, common data formats and standardised interfaces for these subsystems.

The usage of the proposed harmonised framework is not mandatory for the SUNRISE Safety Assurance Framework but it is highly recommended to create more interoperability, modularity and scalability in the virtual testing process.

1 INTRODUCTION

1.1 SUNRISE project

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: Safety Assurance Framework 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.

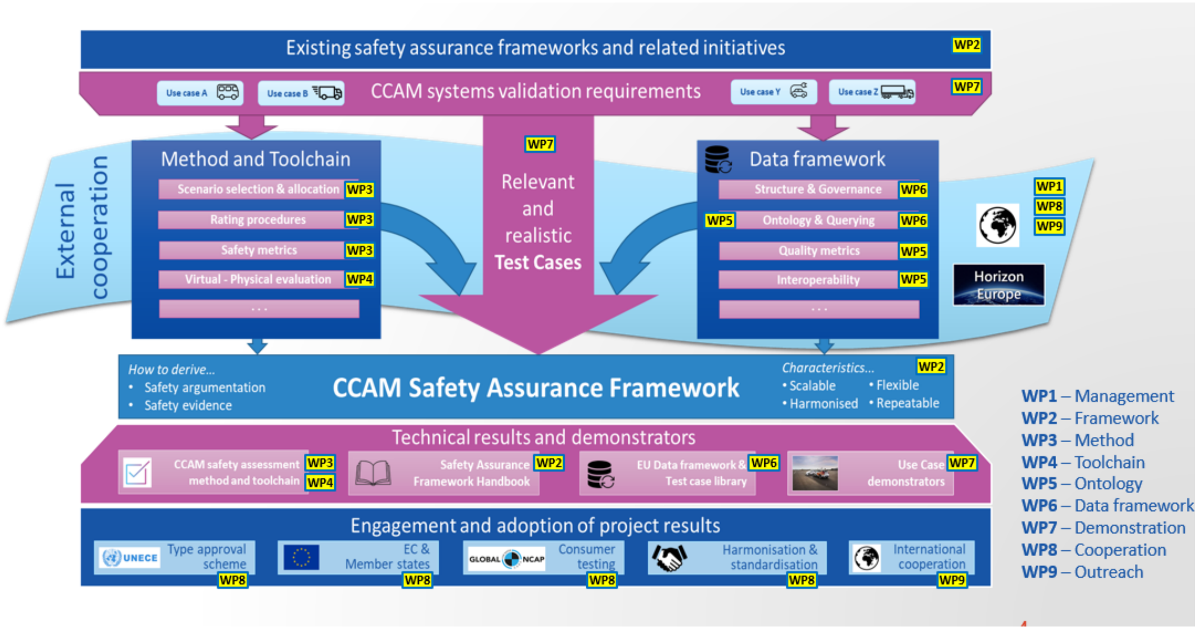

Figure 2: Overview of the SUNRISE Project.

1.2 Purpose of the deliverable

This deliverable is part of work package 4 "CCAM V&V framework", which focuses on the development and validation of a harmonised V&V simulation framework to virtually validate CCAM systems. This task and its deliverable will focus on the design and development of the harmonised V&V simulation framework. The content is based on the work of prior tasks of WP4 but especially on T4.1 in which the subsystems for a harmonised V&V simulation framework were defined. This work only focuses on virtual testing and does not include hybrid or physical testing methods.

This deliverable describes designs of a simulation framework from individual partners, analyses them and proposes a harmonised framework with harmonised interfaces and data formats. This should lead to more modularity and interoperability and thus to a simplification of the simulation process. It is not the aim to make statements or recommendations about simulation tools.

Figure 3 shows the location of this work in the draft version of the overall SUNRISE Safety Assurance Framework (SAF). The simulation framework is part of the virtual testing, which is part of the "Execute" phase of the SAF. The validation of the simulation framework is part of the "Audit" block and will be the main topic of SUNRISE deliverable 4.5.

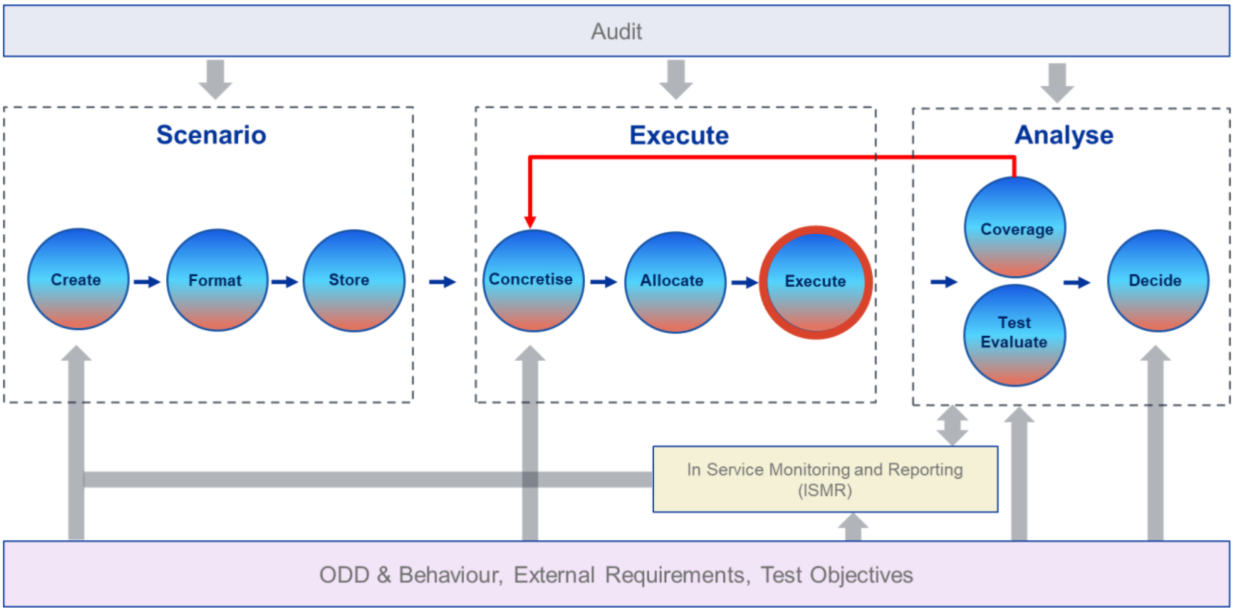

Figure 3: The draft SUNRISE Safety Assurance Framework.

1.3 Intended audience

This deliverable serves multiple stakeholders. One stakeholder is the SUNRISE project itself, as this deliverable presents a harmonised V&V simulation framework that should increase modularity and harmonises the subsystems, interfaces and data formats. The project partners are free to adopt this framework in the design of their use cases. The outcome of this document influences the SUNRISE task 4.5, which is about the validation approaches for the harmonised V&V simulation framework, as well as the remaining tasks of work package 7 that organise the planning, design and execution of the use cases.

This harmonised V&V simulation framework is also recommended for all users of the SUNRISE SAF and provides the same advantages and functionalities to them as well. It serves as a proposal to make the simulation process of the SAF more modular and easier to adjust to different purposes or use cases. It aims to simplify switching between individual simulation tools. It must be emphasised that the use of this harmonised framework is not a prerequisite for the use of the SUNRISE SAF but it is recommended to use it for the reasons of interoperability and modularity.

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

The content of this deliverable is divided in the following chapters:

Chapter 2 introduces the approaches of other projects that work/worked on the harmonisation of simulation frameworks, interfaces or data formats.

Chapter 3 introduces the individual approaches of project partners for a simulation framework. These designs are influenced by the needs of the use cases the partners work on in WP7. Therefore, the focus on certain subsystems differs.

Chapter 4 shows the benefits of a harmonised V&V simulation framework and gives a summary of the outcome from T4.1 (harmonised subsystems) since this is an essential part of the framework. This chapter also presents the outcome of the analysis of the interfaces and data formats that are used in the partner designs. Finally, the harmonised V&V simulation framework is presented as outcome of this deliverable. This will be a input for the task 4.5 that will deal with approaches for the validation of the harmonised V&V simulation framework.

Many of the authors were also involved in tasks of WP5 and WP6 like “T5.2 - Harmonisation of data framework and SCDB content” and “T6.2 – Develop standards output from SCDB” to be aligned on these topics.

2 APPROACHES FROM OTHER PROJECTS

This chapter aims to reflect on the harmonisation approaches of related projects and initiatives. These approaches were considered while working on the SUNRISE harmonised V&V simulation framework.

2.1 SET Level approach

The Set-Level project, a German research project, has the goal to create a credible process for the simulation-based testing of automated vehicles [3]. In doing so it builds upon the work already performed by previous projects within the PEGASUS family. The main output of the project has been the creation of a concept to connect different components of the simulation process together. Additionally, a collection of use cases has been considered to derive requirements necessary to apply the simulation-based approach to real-world problems.

2.1.1 Standards as Common Language

The Set Level Projects uses a variety of different standards to enable the integration of different models and the flow of information between them. Additionally, standards are used to enable the compatibility of models in different simulation toolchains. Common standards used within the Set Level approach include the Open Simulation Interface (OSI), Functional Mock-up Interface (FMI) and System Structure and Parameterization (SSP). Additionally, different simulation processes and approaches were defined within the SET Level project. This includes the Credible Simulation Process (CSP) and Credible Modelling Process, which provides a series of guidelines to ensure traceability, reuse and quality assurance while performing simulation or modelling tasks. The Credible Simulation Process (CSP) and Credible Modeling Process (CMP) are frameworks designed to ensure reliable and structured simulation-based decision-making in product development. In this multi-layered process, decisions regarding design and release often rely on the Simulation-based Decision Process, where the solution approaches for various tasks are determined, potentially involving simulations and real tests. It outlines the steps needed for model development, ensuring the credibility and traceability of results. Another focus is the detailed development of models to meet specified credibility standards. The overall goal of these processes is to minimize risks and ensure that the effort in verification and validation corresponds to the potential impact of a wrong decision. The CSP is adaptable across different companies and applications, offering a structured, generic approach that emphasizes quality assurance and credibility.

<u>OSI</u>

ASAM OSI (Open Simulation Interface) [4] is a standardized interface developed for the automotive industry to facilitate the simulation and testing of automated driving systems. It ensures compatibility between different simulation models, sensors, and automated driving functions. ASAM OSI uses Protocol Buffers, a mechanism for serializing structured data into compact binary formats, for efficient data exchange and includes various interfaces to model environmental perception, traffic participants, and sensor data. It integrates with other standards like OpenDRIVE and OpenSCENARIO to create a comprehensive simulation environment. This standard has been used within the SET Level project to connect a multitude of different simulation models, like sensor-, perception- or driver models together and integrate them in a common framework.

FMI

The Functional Mock-up Interface (FMI) [5] is a standard for exchanging and integrating simulation models across different simulation tools. It ensures that these models can work together seamlessly. A Functional Mock-up Unit (FMU) is a packaged model that adheres to the FMI standard, containing all necessary code, data, and a standardized interface. FMUs can be used in different simulation environments for both model exchange and co-simulation, making it easier to share and integrate models across various platforms.

SSP

System Structure and Parameterization (SSP) [6] is a standard designed to facilitate the modular description and exchange of dynamic systems. It allows for the effective management of system structures and parameters in complex, interconnected subsystems, which is particularly beneficial in fields like automotive and aerospace engineering.

The system structure component of SSP defines how various components of a system are connected and organized in a hierarchical manner. This ensures that the relationships and dependencies between different parts of the system are clearly described. The parameterization aspect involves managing the parameters of these components to ensure that the system can be accurately configured and simulated, reflecting the intended behaviour based on the specified parameters.

SSP uses a standardized packaging format to encapsulate the system structure and parameter data, which enhances compatibility and ease of exchange between different simulation tools. This standard promotes interoperability by allowing different tools to work together seamlessly, supports modularity by enabling the independent development and testing of subsystems before they are integrated, and fosters reusability by allowing system components to be reused in different projects.

Overall, SSP streamlines the process of integrating and simulating complex systems, making it an essential standard in engineering fields where multiple subsystems from various sources need to be efficiently combined.

Simulation Models

Several different simulation models were considered in the SET Level project. These include a Camera Sensor Model, Vehicle Dynamics Model, Highly Automated Driving Function, Motion Control Model, Driver Model and Pedestrian Model.

The Camera Sensor model uses C++ code to process sensor views into detailed sensor data. It can model objects, handle occlusion, and account for uncertainties in field of view, range, and state (like position and velocity).

The Vehicle Dynamics model, built with Simulink and FMU, converts dynamics requests into traffic updates. It uses a 2D kinematic single-track model and considers physical limits of vehicle dynamics, such as those imposed by the powertrain.

The Highly Automated Driving Function employs ROS, Docker, and TCP to integrate sensor data, vehicle communication data, and traffic commands, outputting motion commands. It features capabilities for route planning, sensor fusion, object tracking, and trajectory planning.

The Motion Control model, also based on Simulink and FMU, translates motion commands into dynamics requests. It utilizes nonlinear state control to determine the necessary acceleration and curvature for the vehicle's trajectory.

The Driver Model, leveraging C++ technology, processes sensor views and traffic commands to generate dynamics requests and traffic updates. It includes route planning, right of way management, speed limit adherence, and an internal vehicle dynamics model.

Finally, the Pedestrian Model uses C++ to process ground truth, sensor views, and traffic commands, producing traffic updates. It features path-finding using the Theta* algorithm and a social force model for avoiding pedestrian collisions.

2.1.2 Link to SUNRISE

The SET Level project provides a range of valuable outputs to be considered and worked upon within the SUNRISE project. Especially relevant is the work within SET Level to establish the interconnection between different models and model types within a larger simulation framework and the different standards that make this possible. Additionally, many processes around performing simulations, as well as model selection and distribution have been created in this project and form a valuable basis for SUNRISE.

2.2 P.E.A.R.S / V4SAFETY approach

V4SAFETY is a HORIZON EUROPE funded research project (HORIZON-CL5-2022-D6-01-06 Area A) [7], aiming to develop a method for conducting computer simulations to assess the safety performance of road safety solutions, a so called “Harmonised Prospective Safety Assessment Framework”. The project builds on the experience gained in the P.E.A.R.S. initiative, an open consortium (established in 2012 as Harmonization Group) in which engineers and researchers from over 30 partners from the automotive industry, research institutes and academia join to develop a comprehensible, reliable, transparent, and accepted methodology for quantitative assessment of crash avoidance safety measures by virtual simulation [8]. Both activities are further described below:

2.2.1 P.E.A.R.S. (**P**rospective **E**ffectiveness **A**ssessment for **R**oad **S**afety)

P.E.A.R.S. has been conducting efforts towards harmonising the prospective assessment of safety performance of vehicle integrated pre-crash safety measures based on virtual simulation. Besides an overview of the process to be followed, its content covers in detail the different steps to be taken into account, starting from the definition of the evaluation scope, the baseline selected for the study, the needs of the virtual simulation framework, the assessment of the safety performance of the safety measure under study, as well as the required documentation and the applied validation and verification techniques. All these steps are in both ISO/TR 21934-1:2021 [9] and ISO/CD TS 21934-2 [10]. The experts involved in P.E.A.R.S. have been directly contributing to the preparation of both ISO documents.

The proposed P.E.A.R.S. framework consists of below four main steps:

1) Evaluation Scope: In this step, the target of the study has to be defined by means of a precise research question. The main information that should exist in the research question is:
    - the safety measure to be assessed,
    - the relevant scenarios to be assessed.
    - the metric to assess the safety performance.

Additional information such as considered limitations as well as any spatial or temporal projection of the analysis may be described, depending on the scope of the study.

2) Baseline definition: It defines the situation to be assessed without the safety measure being implemented. Depending on the selected input data as well as how it is treated and processed, four different baseline approaches can be identified:
    - Baseline Approach A: They represent individual real-world scenarios, described in time-series, where a link exists to a real-world situation.
    - Baseline Approach B: They represent modified real-world scenarios, described either as time-series information or by the use of in-simulation models, where a modification to the real-world situation is made.
    - Baseline Approach C: They represent a synthetic case where no direct link exists with a real-world scenario. Two sub-categories may exist, if the trajectories of the participants are pre-defined (approach C1), or if they are derived from in-simulation models (approach C2).
3) Virtual Simulation Framework: A generic simulation framework is proposed which contains all possible required models for running simulations (see Figure 4), where each of the required models is derived from the evaluation scope. The simulations of both baseline (situation without safety measure) and treatment (situation with safety measure) will be simulated following the identified models in the framework.

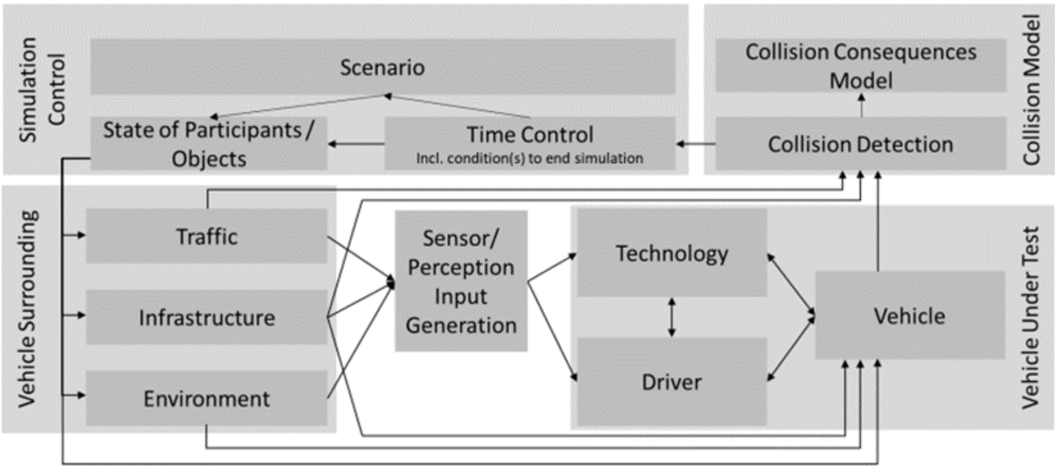

Figure 4: P.E.A.R.S. Virtual Simulation Framework overview *[8]*

4) Assessment of Safety Performance: The final step consists on assessing the safety performance of the safety measure under assessment, based on the results from the simulations of the baseline and treatment, by using the metrics defined in the evaluation scope.

Besides the four steps described above, the ISO activities where P.E.A.R.S. method is described also highlight the need to include:

- A Validation and Verification (V&V) process, to make sure that the overall followed process is correct and that the obtained results can be trusted.
- Documentation of the study (including not only the obtained results, but also the whole process followed).

2.2.2 V4SAFETY (Vehicles and VRU Virtual eValuation of Road Safety)

As mentioned before, the V4SAFETY project [7] builds on the experience gained in P.E.A.R.S. consortium over several years. In order to facilitate the exchange of both activities, all the V4SAFETY project partners are now also P.E.A.R.S. partners. .

V4SAFETYaim to develop a harmonised prospective safety assessment framework (in short V4SAFETY framework) and to demonstrate it in relevant use cases as well as ensuring its acceptance by making sure the approach is standardised (relation to ISO activities is already present via link with P.E.A.R.S.), transparent and holistic. Whereas SUNRISE focuses on a method related to safety assurance for vehicles, V4SAFETY scope is not limited to that and aims to address any type of safety assessment of a road safety measure (in-vehicle solutions, behavioural or infrastructure), by a wide variety of relevant stakeholders (e.g. a researcher who aims to understand the safety benefit of a road infrastructure measure to be developed in a specific area within a city, a developer who aims at understanding the safety performance of a future safety systems in terms of accident reduction...). On the other hand, V4SAFETY scope is strictly related to the use of simulation, while SUNRISE covers also test track and real-world test. These means, that several aspects which are specifically related to the safety assurance framework (e.g. test allocation, analysis coverage, analysis decision, in service monitoring...) are out of the scope for V4SAFETY. It is expected that V4SAFETY approach can support SUNRISE in how virtual testing is addressed (e.g. data and used models, aspects to be considered during simulation execution, as well as assessment of results based on identified metrics).

V4SAFETY project started in October 2022 and will run until September 2025. The project has already delivered a draft of the V4SAFETY framework [11], which is illustrated in Figure 5. The deliverable which will include the final version of the V4SAFETY framework, will be D2.2 (Prospective Safety Assessment Framework – Method) and will be released in September 2025.

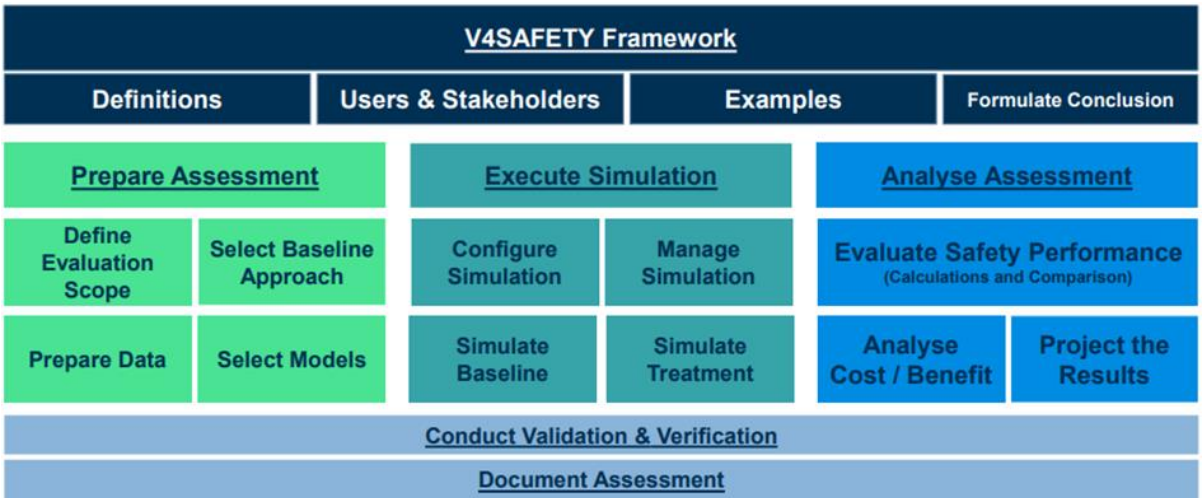

Figure 5: V4SAFETY framework overview

The V4SAFETY framework builds on the steps from the P.E.A.R.S. framework, developing further each of them, as well as including additional cross-functional aspects such as the consideration of used *"Definitions"*, *"Users & Stakeholders"* of the study, provision of *"Examples", "Formulation of Conclusions", "Validation & Verification"* as well as *"Documentation"*.

Besides the cross functional topics, the V4SAFETY framework considers several topics which can be grouped into 3 main pillars, described below:

- *Prepare Assessment*: This part includes the *"Definition of the evaluation scope"*, including the research questions, the *"Selection of the Baseline Approach"*, based on the evaluation scope and the available data, the *"Data preparation"* which may include data selection and data processing steps and the *"Model selection"* which will depend also on the evaluation scope and the selected baseline approach.
- *Execute Simulation*: Two main activities are considered in this pillar, the preparation part, which is described by the *"Configure Simulation"* and *"Manage simulation"* blocks, and the practical part of executing simulations, covered by the running of simulations with and without the safety measure under assessment, represented by *"Simulate Baseline"* and *"Simulate treatment"* respectively.
- *Analysis Assessment*: The last pillar deals with the output of the simulations executed aiming to provide an assessment with regards to the evaluation scope of the study. It should at least contain an assessment of the safety measure under evaluation

"*Evaluate Safety Performance*" and, depending on the evaluation scope, "C*ost/benefit analysis*" as well as "P*rojection of results*".

The three pillars are connected to each other, and in many cases, it requires iterative loops until it can be ensured that the right data is ready to conduct the assessment as defined in the evaluation scope.

2.2.3 Link to SUNRISE

The presented activities cover the relevant aspects that should be considered when using simulations to assess the performance of a safety measure. In SUNRISE, WP4 deals with "CCAM V&V framework", which includes a larger scope than the activities related to the use of simulations, such as physical tests and real-world testing. Additionally, SUNRISE has a clear scope in terms of safety measures, as it refers to CCAM systems, which can be considered as an in-vehicle safety measure, although V4SAFETY also covers safety measures beyond in-vehicle ones (e.g. infrastructure, behavioural...) Despite the above presented differences, this subsection aims to focus on the topics where V4SAFETY can support how the CCAM V&V framework is addressed within SUNRISE.

From the V4SAFETY framework, the pillar Execute Simulation has a direct link with the presented scope of CCAM V&V framework within SUNRISE Task 4.4. The topics *"Simulate Baseline"* and *"Simulate treatment"* correspond to the actual execution of simulations, using a simulation structure that can be generically represented from the P.E.A.R.S approach. Virtual Simulation Framework overview shown in Figure 6.

In addition, the topic "*Configure Simulation*" deals with tasks that need to be done before the execution of simulations, such as assembling the simulation models according to the simulation structure, for which again, a generic one is represented in Figure 6. A visualization of the topic "Configure Simulation" and the actions covered in it, can be seen in Figure 7.

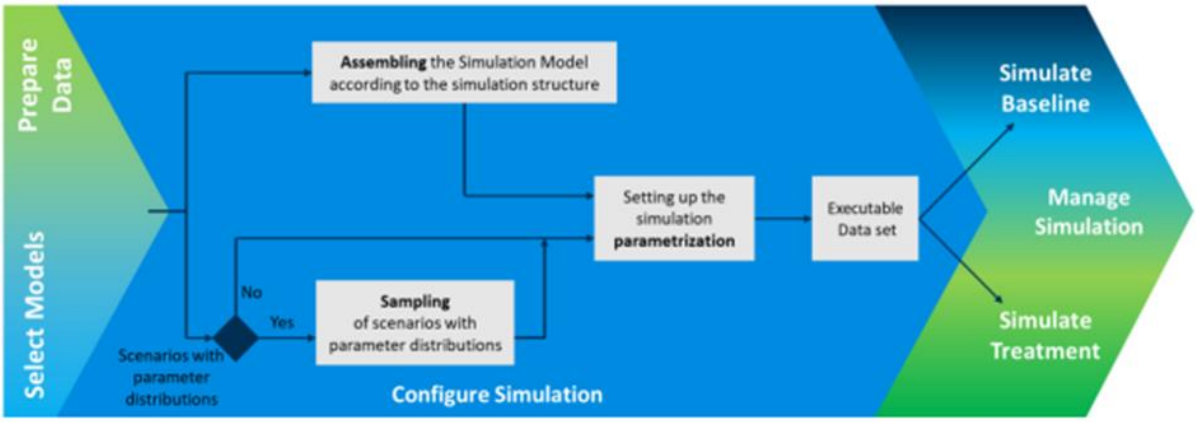

Figure 6: Visualisation of "Configure Simulation" topic in V4SAFETY Framework

The topic "*Manage Simulation*" is also part of the pillar Execution Simulation, and deals with the process associated with how the simulations will be run, executed and the storage and handling of the resulting simulation data. A visualization of the process and actions executed in this topic "*Manage Simulation*" can be seen in Figure 7.

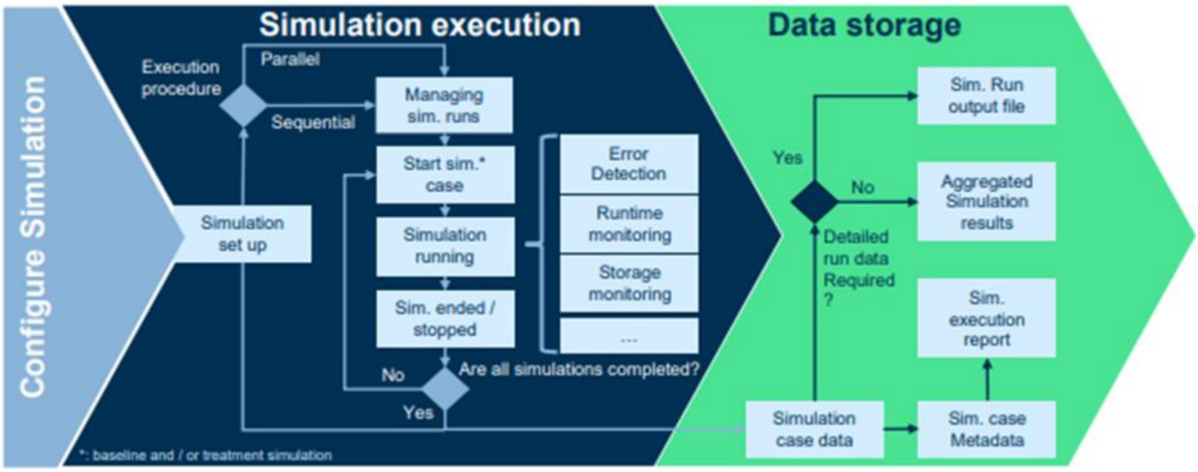

Figure 7: Visualisation of "Manage Simulation" topic in V4SAFETY Framework

Despite the differences in scope of SUNRISE and V4SAFTY which have been presented before, there is a clear link between both project when it comes to the use of simulation. for assessment of a safety measure. The Harmonised V&V Simulation Framework from SUNRISE, might take into account the pillars and topics mentioned in the V4SAFETY framework, as simulations per se, are the result of the execution of the simulation framework, but it also needs to be understood what needs to be done prior to that ("*Prepare Assessment*"), and what to do with the simulations obtained, ("*Analyse Assessment*").

2.3 ArchitectECA2030

In addition to several technology-related activities in the areas of environmental perception, powertrain and connectivity, ArchitectECA2030 focuses on the design and development of a harmonised safety validation framework that can be applied to all targeted demonstrators related to different domains. The idea was to use all the lessons learned and related user stories collected during the design, development, verification and validation of the demonstrators to design and develop the concept of the target validation framework. Together with the related residual risk assessment, see Figure 8, both contribute to the target reference homologation process (RHP). As part of the RHP, the ArchitectECA2030 validation framework must meet the requirements of the entire automotive lifecycle and be ready to address updates to the ADAS/AD functions after their deployment. ArchitectECA2030 describes the validation framework as harmonised as it is applied to several different automotive domains, including perception, powertrain and connectivity, taking into account their domain-specific aspects. In addition, feedback and lessons learnt during the operational and monitoring phase after successful deployment will be incorporated into the RHP and therefore also addressed by the validation framework.

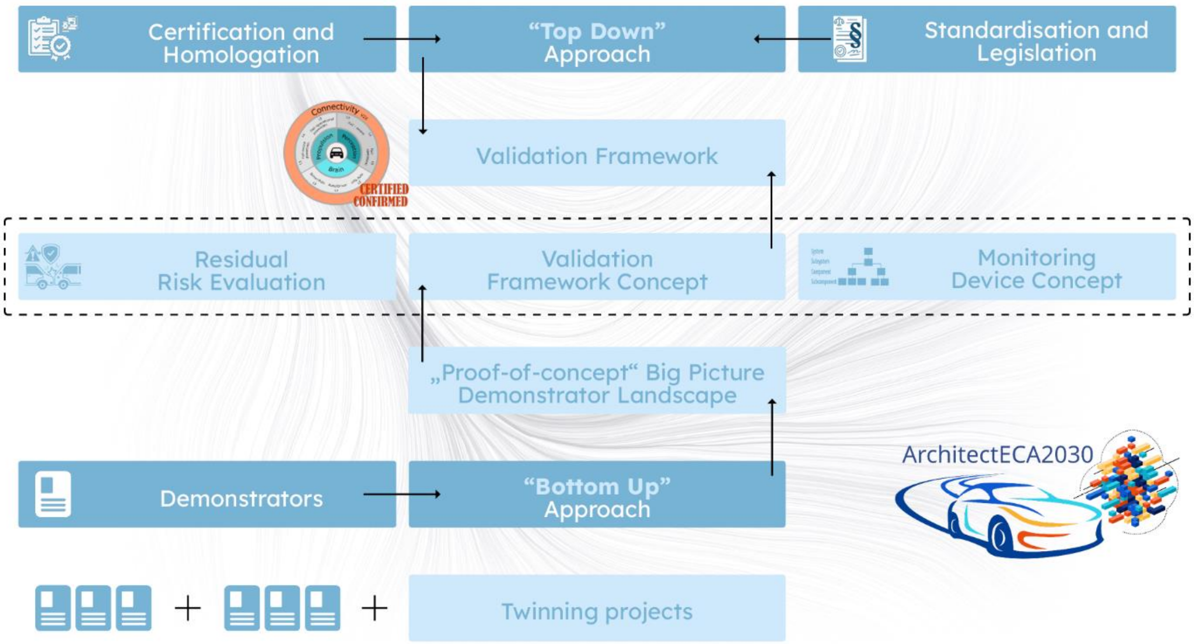

Figure 8: ArchitectECA2030 big picture: top down meets bottom up approach

All demonstrators were structured using the 4-layer concept [12], which represents the whole vehicle in four different layers, starting with the system layer, the sub-system layer, the component layer and ending with the sub-component layer representing the finest granularity, see Figure 9.

In this sense, the requirements flow starts from the main sources including the target ODD for deployment, the target behavioural competencies, certification, homologation, legislation and standardization related aspects feeding the system layer in terms of the Electric Connected and Automated ECA vehicle itself. Based on this collection, requirements are pushed through all layers down to the subcomponent level. Based on the requirements flow, all layers develop their building blocks according to the layer specific requirements.

Each layer implements its monitoring device activities according to the principle of receive, monitor and transmit [13]. During the receive part, all necessary information about the subcomponent / component / subsystem / system health status of the previous layer is collected. The monitor part monitors and classifies the health status of the current layer including the effect of the collected health status of the previous layer. Finally, the transmit part communicates the current health status to the next layer.

The determined health status forms the basis for determining the corresponding residual risk. By connecting all receiving and transmitting units of the individual layers, the fault/residual risk propagation across several layers can be analysed. The main objective is to reduce the residual risk where it occurs, without allowing it to propagate to the system layer. In all cases where this is not possible, the fault and risk propagation are analysed at the system level and approaches are implemented to adequately reduce the operational ODD and/or the corresponding behavioural competencies.

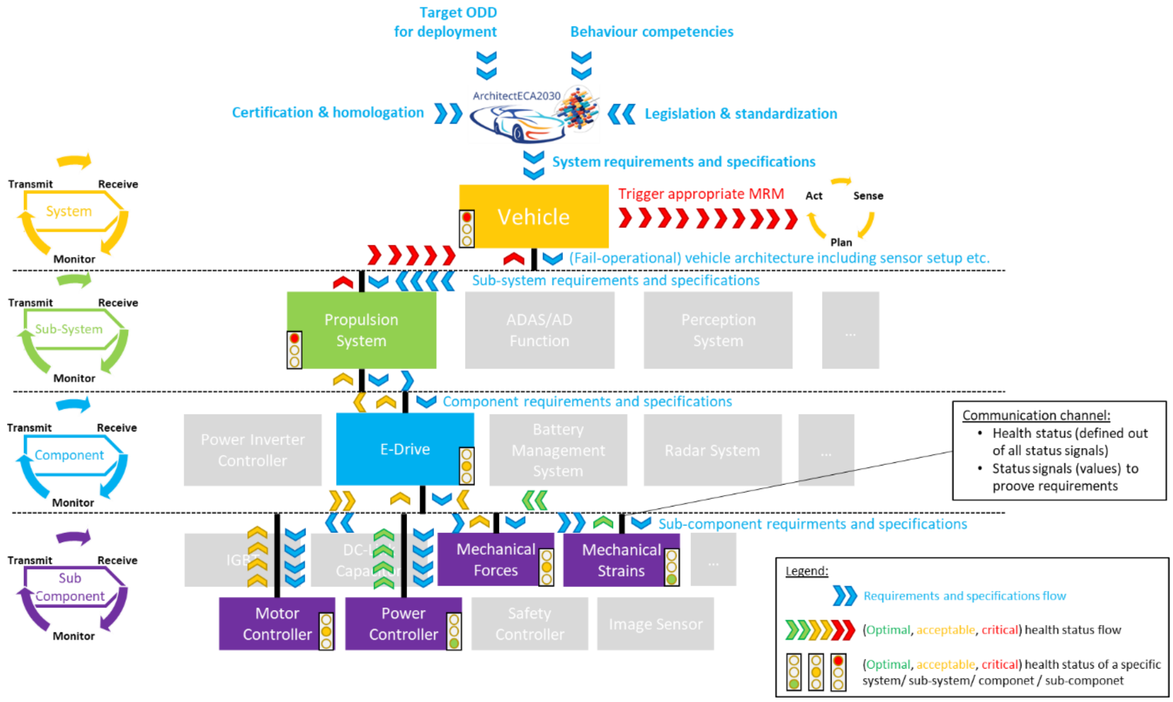

Figure 9: Requirements vs. health status flow across all layers of future ECA vehicles in-line with the hierarchical monitoring device concept.

In order to validate the monitoring activities within all the demonstrators, which are linked to different domains and address several different vehicle layers, the so-called Harmonized Validation Framework concept was created, see Figure 10.

The validation framework consists of a so-called base layer consisting of 4 interconnected subsystems, namely the vehicle, the sensors installed in the vehicle, the behavioural competencies of the vehicle and the environment in which the vehicle operates. In addition to the functional interconnection, a controlled execution is also required, including the targeted test cases and underlying scenarios, together with the KPIs to be evaluated with the validation framework.

In addition to the base layer, the user can extend the framework in 4 dedicated dimensions related to the target ODD, the vehicle sensor setup, the SW architecture and the HW architecture. The ability to extend the base layer with demonstrator specific elements supports the general applicability of the proposed harmonized validation framework. For example, the entire ODD can be built using a few best-in-class environment simulation tools capable of representing specific ODD attributes in all details (high fidelity) and together covering the entire target ODD. The same construction principle can be applied to the sensor setup, SW and HW architecture.

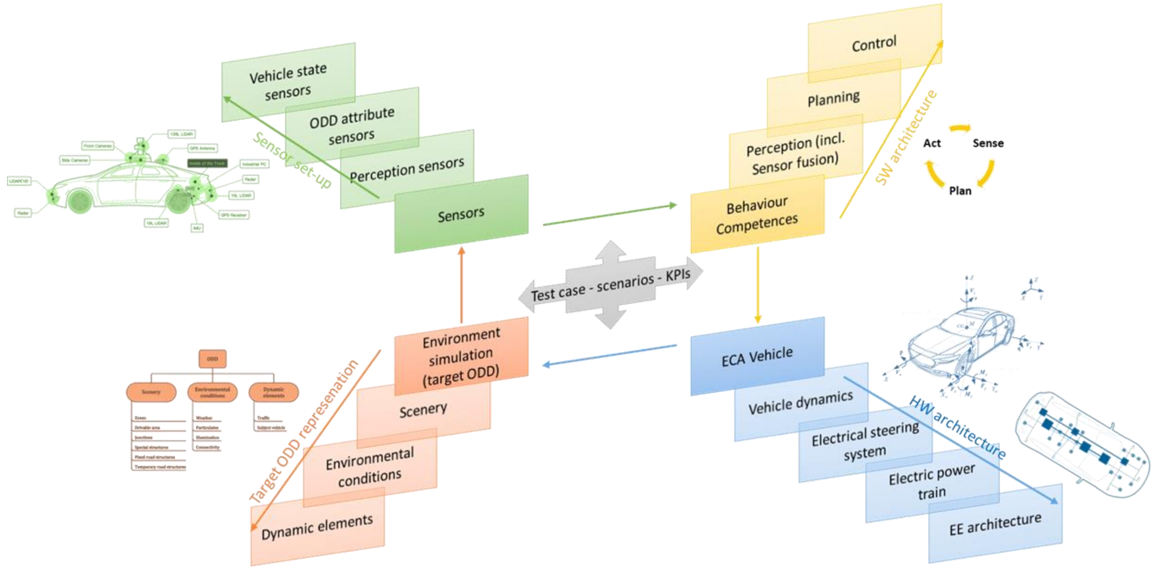

Figure 10: ArchitectECA2030 harmonized safety validation framework.

2.3.1 Link to SUNRISE

The ArchitectECA2030 Harmonised Validation Framework shows a strong link to Work Package 4 (Task 4.4) as both frameworks have Safety Assurance as their main objective. Specifically, the lessons learned and insights gained from several approved demonstrators applying the ArchitectECA2030 Harmonised Validation Framework, corresponding to different domains, can be adopted to be included in the Safety Assurance Framework within SUNRISE. In particular, the proposed structure of the validation framework including the base layer concept together with its extension seems to be a promising approach for the SUNRISE SAF.

2.4 DIVP®

As part of the efforts from Japanese Government to pursue scientific innovation, the Strategic Innovation Program (SIP)-Adus Phase 2 established the DIVP® project, focusing on developing a safety assurance evaluation platform in a virtual environment [14].

The project focused on modelling physical properties of different sensor types such as camera, radar and lidar, in various environment conditions and for different types of scenarios. The focus has been on accurately simulating sensor output so that recognition and detection performance can also be part of the virtual validation. As part of the validation and verification aspect, the project has also included real vehicle test results to verify the comparison of the obtained results. The overall outcome of the project, also named DIVP® product can be seen in Figure 11.

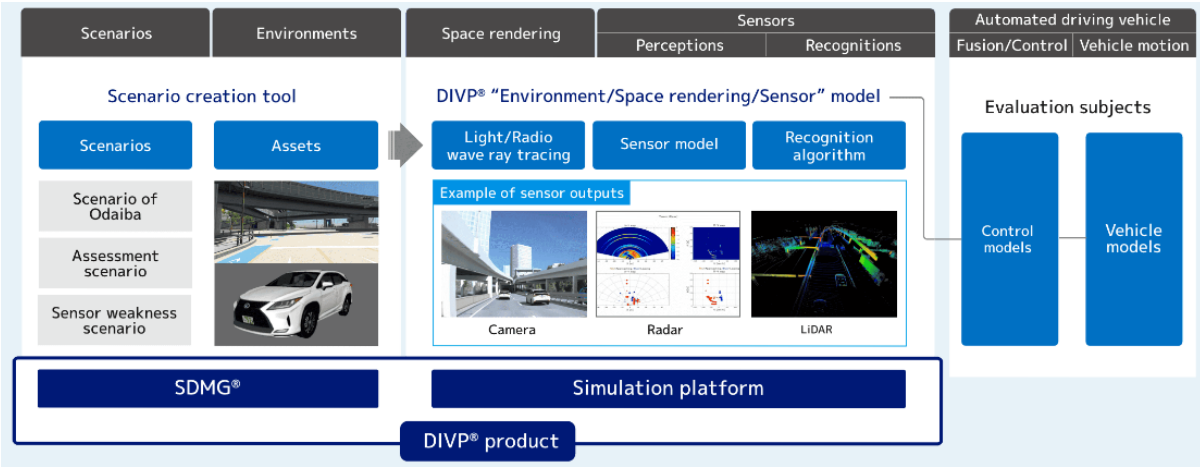

Figure 11: DIVP product

As can be understood from Figure 11, there is a link with the safety assurance methodology, which in Japan is covered by the SAKURA project [15]. The DIVP product covers therefore a practical implementation of a platform that enables scenario generation, recognition performance evaluation and vehicle control verification.

2.4.1 Link to SUNRISE

Although DIVP® project has been dealing with technical challenges of modelling physical phenomena of different sensors in different environment situations, the outcome of the project fits very well with the scope of SUNRISE Task 4.4, as it shows a clear example of how modelling of sensors should be considered towards validation and verification of the results obtained via simulation. Additionally, the DIVP® project and product also shows a clear link of how such a solution can be integrated within a safety assurance framework, as it has been demonstrated via the SAKURA project.

3 DESIGN OF INDIVIDUAL V&V SIMULATION FRAMEWORKS

This chapter gives an overview of the individual simulation framework designs of the partners involved in this task. These simulation frameworks were designed to be used in WP7 (especially T7.3 – Safety Assurance Framework demonstration instances) and the SUNRISE use cases and therefore differ in their focus (e.g. perception, connectivity…). These designs are analysed in chapter 4 and serve as a basis for the harmonisation approach in chapter 4.5.

3.1 AVL simulation framework approach

The simulation framework from AVL, is characterized by its **scalability** and **modularity**, two key features that make it versatile and adaptable to any testing procedure.

AVL simulation framework supports **ASAM OpenSCENARIO**, **OpenDRIVE**, and **OSI** standards. These are open and standardized formats for the description of driving simulation scenarios, road networks, and interfaces for sensor models in virtual environments respectively. The support for these standards ensures that the framework is compatible with a wide range of scenarios, driving environments, and sensor models, thereby further enhancing its versatility and adaptability. AVL framework is created using Model.CONNECT which is developed by AVL. Model.CONNECT is a tool that is used for the integration of different tools from different vendors for seamless co-simulation. Connection between environment and sensors we use OSI data, to not be limited by the number of entities in the simulation, while the rest of the system uses signal data.

- **Scalability –** in this context we are referring to the ability to run the simulation framework locally on high-performance computers or running it in the Cloud environment. With this feature, it can be easily added to existing CI/CD pipelines so that this becomes a standard part of ADAS/AD software development.

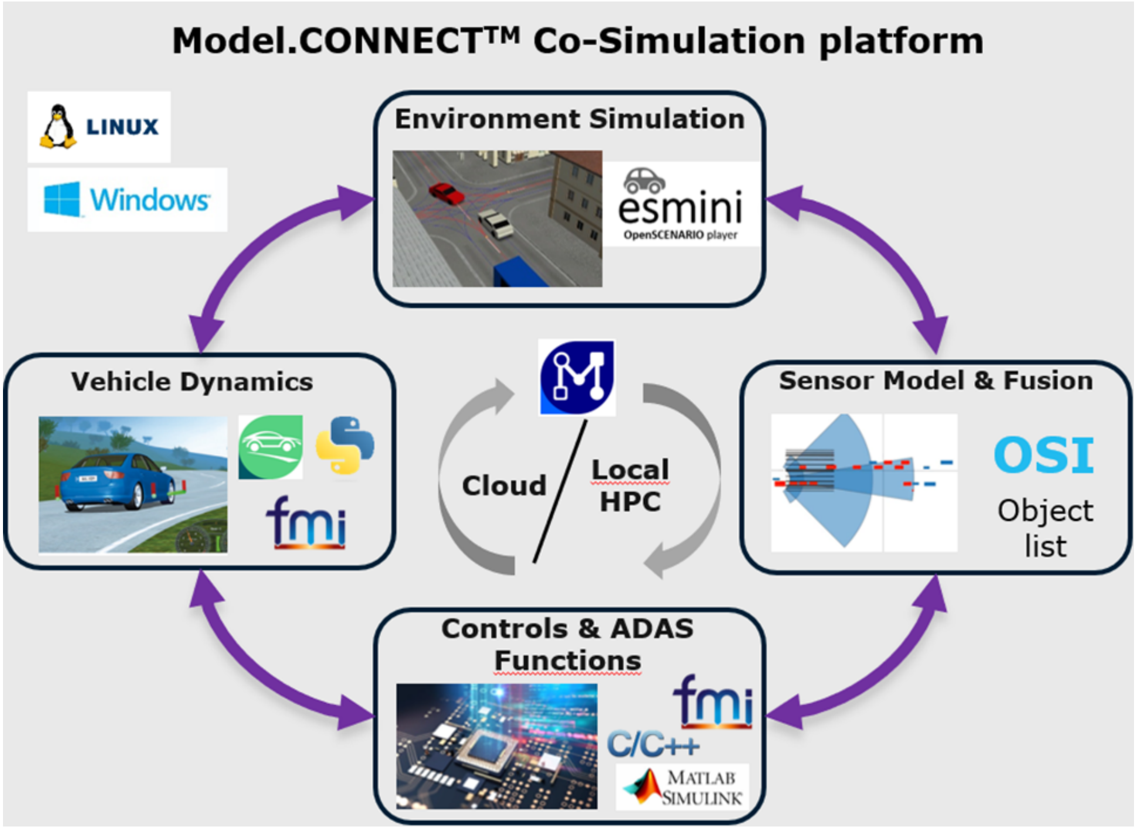

Figure 12: Scalability

- **Modularity** – The fidelity of the models which are used in the Simulation Framework can easily be changed so that we adapt our setup to the testing requirements. This means that components like Vehicle, Sensors, Traffic Manager, etc. can be easily changed, upgraded, or modified without disturbing the overall functionality of the AVL Simulation Framework.

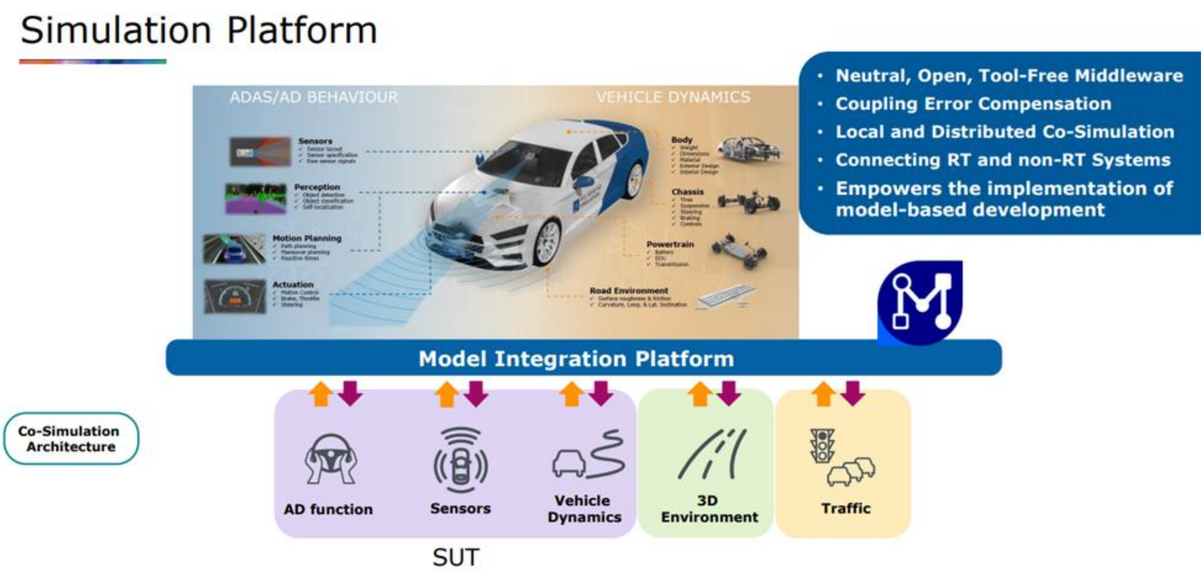

Figure 13: Modularity

3.2 CAF simulation framework approach

The simulation framework of Continental Automotive France (CAF) divides the ODD parametric space based on the BSI PAS 1883 model. This means that the ODD is divided in three main components:

- **Scenery**: Provides context to the test environment including information about special road configurations, structures and elements (both permanent and temporary).
- **Environmental conditions**: Clusters everything with respect to the weather, the illumination, and the connectivity.
- **Dynamic elements**: Gathers all the traffic participants, including the ego-vehicle, other vehicles, vulnerable road users, etc.

A series of parameters are defined belonging to each element of each of those groups, together with their corresponding ranges of values and which probability distribution will be used to sample them. Each parameter is also labelled as either *discrete* or *continuous*. The values of the discrete ones are limited (e.g. the number of vehicles that will be a part of the simulation, the model of a vehicle, the manoeuvre to perform, etc.). The *continuous* ones, however, can have any value from within a defined range (e.g. positions, speeds, rain intensity, etc.).

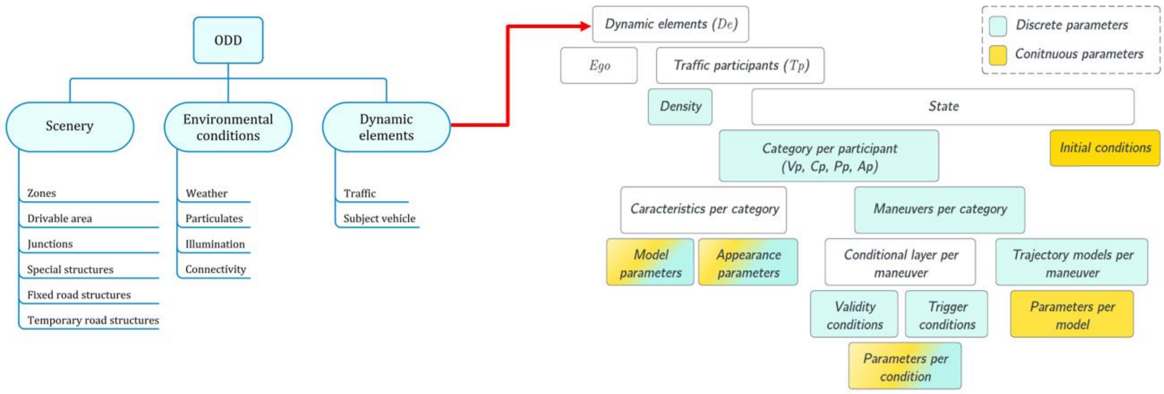

Figure 14: Examples of discrete and continuous parameters in the context of the BSI PAS 1883

It is worth mentioning that some things in the simulation framework are limited by the functionalities of the simulation engine used (CARLA). For example, the way the time of the day is sampled, or the intensity of the weather conditions is completely based on the parameters offered by the simulation tool.

To sample from the variables, the Latin Hypercube Sampling (LHS) is used. In comparison to a completely random sampling, the LHS approach allows to represent the parameter space with a smaller set of samples, at the cost of making a pseudo-random parameter sampling instead of a completely random one.

Due to the physical constraints of the application (a 3D simulation of objects that must not overlap in their initial states), a Constrained LHS (CLHS) is used for the initial position of the traffic participants. The LHS is forced to only output samples that will ensure a certain minimum distance between the traffic agents, ensuring that no initial configuration is generated in which the vehicles would overlap.

In Figure 15, a representation of this is shown. Each graph represents the initial positions of five vehicles. The horizontal axis is their X position (perpendicular to the road) and the vertical axis is their Y position (longitudinal to the road). Each coloured rectangle represents the physical size of the cars that are intended to be spawned. The graphs with red dots indicate a constraint violation (i.e. overlapping vehicles) and the blue ones indicate valid sampling. In case a single overlap occurs, the entire initial scenario is considered invalid.

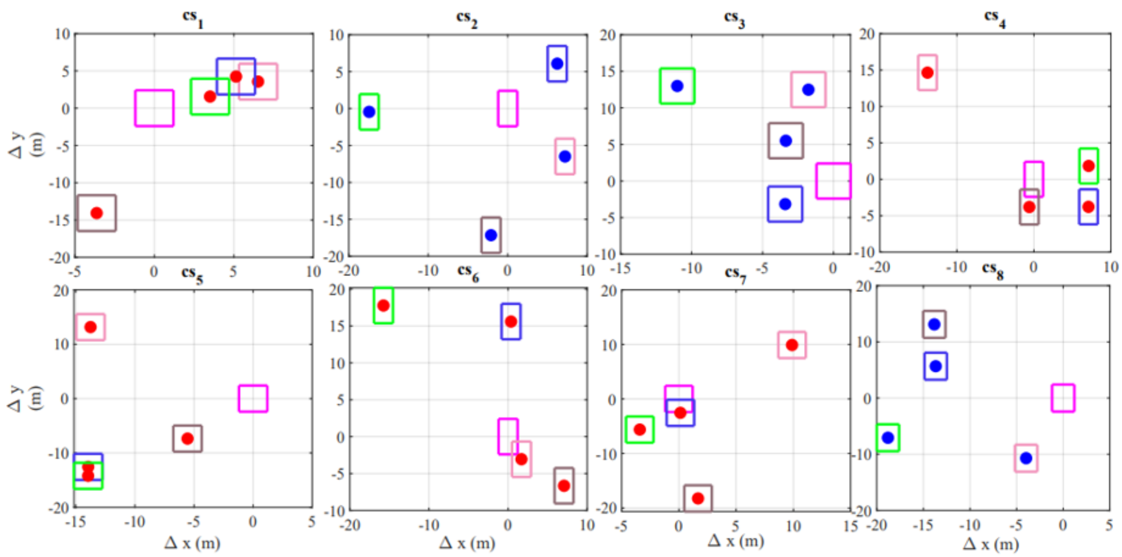

Figure 15: Representation of valid and invalid initial positions of sets of 5 vehicles.

After the sampling, each resulting concrete scenario is stored in an individual file. Our framework supports both CSV and OpenSCENARIO as output formats. Those files are the input to the *scenario manager*, which is able to parse the information contained in them and generate and run 3D simulations that replicate the described conditions and behaviours. That means that the behaviour (or performed maneouver) is always the same, what changes are the sampled variables, such as the initial speeds/positions of road users, weather, vehicle models/colours, etc.

Finally, a criticality metric for each of the generated concrete scenarios is generated. Said metric is based on how close a collision was encountered in a given scenario.

3.3 ICCS simulation framework approach

Connectivity and cybersecurity aspects form part of the harmonized V&V simulation framework. The simulation framework of ICCS will assess sub-UCs 1.3 and 4.2, where ICCS is leading and are mainly related to virtually validate Collective Perception (CP) systems (UC1.3) including a CP system under cyber-attack (UC4.2).

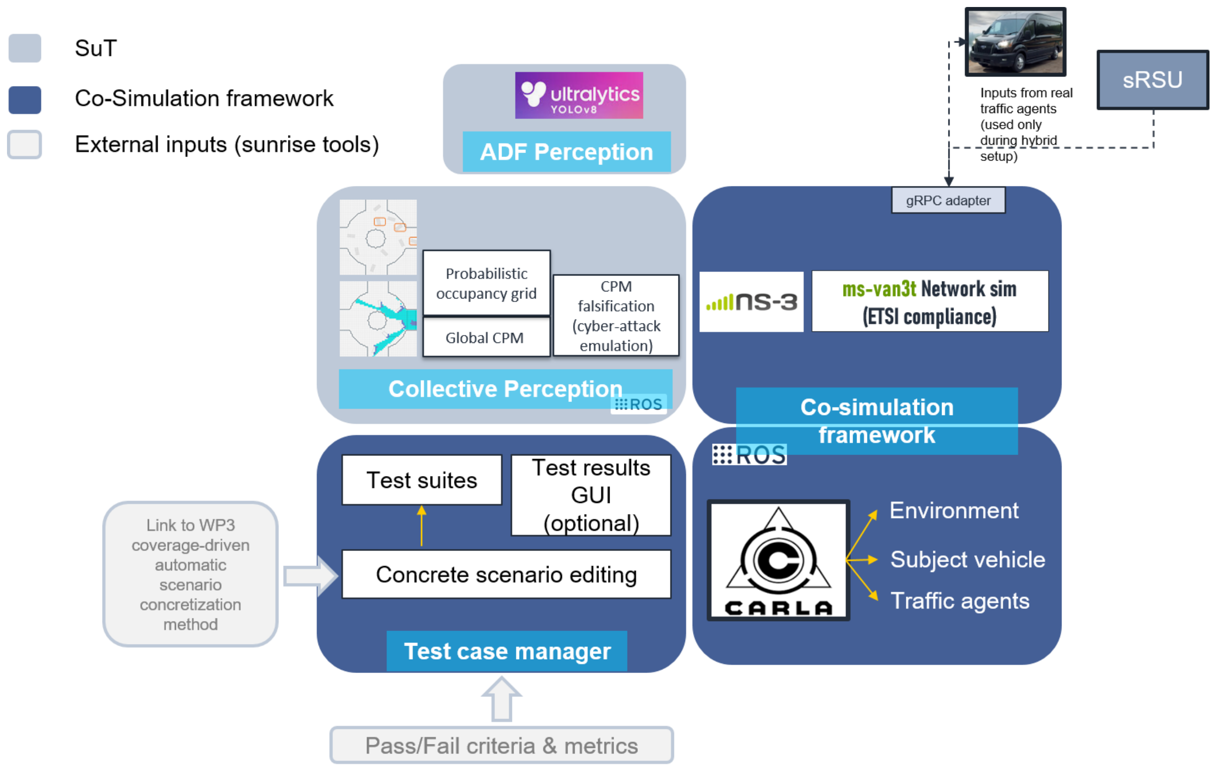

Figure 16: Representation of ICCS simulation framework.

ICCS simulation framework contains some main subsystems as described below and shown in Figure 16.

- **Test case manager:** It feeds the simulation with test scenarios and records the outcomes of each test scenario using predefined perception-related safety metrics. Since such metrics cannot be easily linked with pass/fail criteria there is always the option that these pass/fail criteria will be applied to a perception and control end-to-end system. During the initial use of the system, this module is also used for scenario concretization purposes under coverage constraints.

- **Co-simulation platform that includes:**
  - **CARLA** that handles Sensors, Environment, Subject vehicle, and Traffic agents**:** CARLA sensors (camera, lidar, GPS), environment and agent assets will be used. While other traffic agents (vehicles and pedestrians) will be controlled by CARLA, the subject vehicle will be controlled by an external ICCS controller. For the connected vehicular agents which transmit Collective Perception local Messages (local CPMs), a perception stack is assumed which is developed by ICCS and interfaced with CARLA via CARLA-ROS bridge. For the virtual RSU node, this is implemented as a static sensor spawned in an elevated position overlooking an urban area.
  - **Connectivity**: ns3 simulator extended to support ETSI compliant CPM messaging.
  - MsVan3t open-source **middleware** which is built upon OpenCDA and is responsible for the communication of CARLA with ns3 [16] that uses CARLA

for mobility and sensor perception simulation and ns3 for network simulation [17] [18].

- **ADF Perception:** Each connected vehicle subsystem includes a (pre-existing) perception component deployed outside CARLA environment and implemented in Python. It takes sensor loggings as input from CARLA and outputs object bounding boxes and lanes. It can also be the SuT.

- **RSU collective perception**: The main SuT in all SUNRISE experiments also deployed outside CARLA environment. ICCS (pre-existing) module that fuses local CPMs received from connected traffic agents and produces a globally aligned CPM based on a Bayesian fusion algorithm. It also includes functionality for falsifying CP messages which is used in UC4.2 (this can also be achieved via other means, e.g. by spoofing CARLA camera sensor).

Technical note: SuT can be tested via data replay, where it is fed by logged simulated data. Suppose the SuT and co-simulation framework concurrent execution is desired. In that case, this can be also supported where ROS and gRPC are used as interfaces among the framework main subsystems and the SuT.

As also shown in Figure 16, the framework expects inputs from WP3 regarding the test objectives and the concrete scenarios respectively. For the latter, a probabilistic method based on Gaussian Processes (GP) will be employed for the generation of critical scenarios (implemented in T3.3) for perception testing. It will be evaluated by sub-UC 1.3 led by ICCS. The advantage of this method is that it can provide pass/fail space coverage information hence playing a vital role in SAF's safety argumentation.

3.4 IFAG simulation framework approach

The utilization of virtual testing as a principal methodology for the assurance of ADS safety necessitates the implementation of an adequate simulation framework that encompasses all pertinent subsystems. In essence, there exists a de facto multitude of simulation frameworks, each of which is typically designed to fulfil a specific function. The IFAG simulation framework employed within the SUNRISE project is defined following the overarching framework proposed by [19].

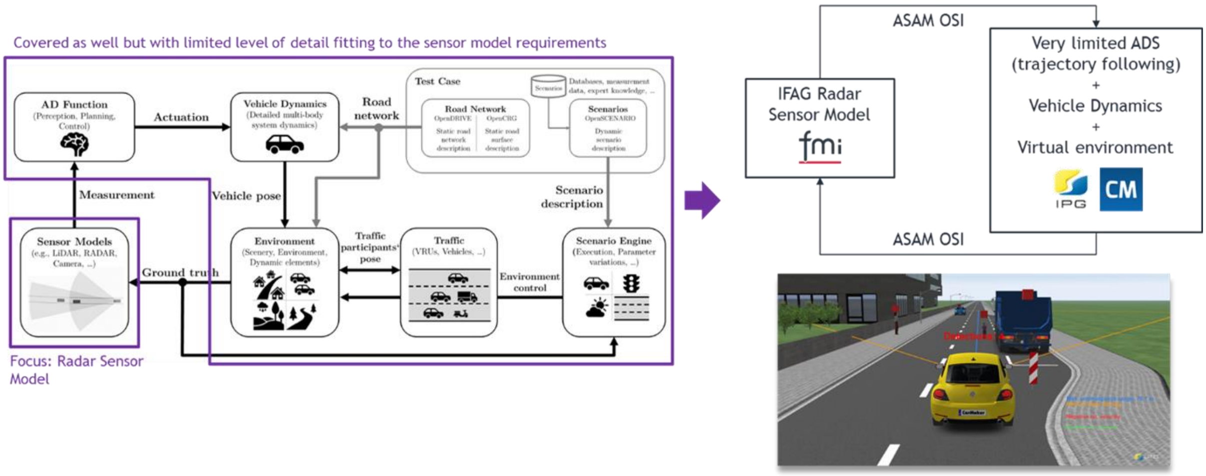

Figure 17: IFAG simulation framework.

As can be observed in Figure 17, the four principal blocks, which are fundamental and represent the foundation of every architecture for the virtual validation of ADS, are

1. the ADS itself (AD function),
2. the vehicle dynamics of the ego vehicle,
3. the virtual environment (representing the defined ODD),
4. and the sensor models.

Sensor models utilize the ideal ground truth input from the virtual environment and calculate the output based on the modelled behaviour of actual sensors. The output may be either data that has already been processed (e.g., object lists for detected traffic participants, lane markings) or raw sensor data.

It is also important to consider the fidelity of the individual models involved in the simulation environment for virtual validation purposes. This is because the fidelity of the models must match in order to create meaningful simulation results. To illustrate, if one is to simulate the behaviour of emergency evasion ADS functionalities under wet conditions on suburban roads, it is essential to ensure that the environment is capable of modelling rain and wet roads accurately. Furthermore, it is crucial to ensure that the correct information can be passed to the involved sensor models, considering the effects of wet conditions. This information must then be passed to the emergency evasion system, which in turn triggers the corresponding evasion manoeuvre to be modelled correctly by the implemented vehicle dynamics models. The interconnected four main blocks demonstrate the significance of ensuring that the fidelity of the models is aligned. In the event that there is a discrepancy between the fidelity of the models, for instance, if the sensor model is unable to consider rain and wet conditions accurately, the simulation results will be rendered meaningless.

The high-fidelity radar sensor model is based on standardized interfaces such as ASAM OSI and FMI. All other three main subsystems, namely the environment, the vehicle dynamics and the ADS functionalities are modelled within the IPG CarMaker simulation software. Furthermore, IPG CarMaker serves as a test case manager, providing the targeted test cases related to EuroNCAP benchmark scenarios and evaluation functionalities to analyse the set metrics and KPIs. Moreover, it serves as an FMU-Master for the radar sensor FMU. Figure 17 depicts the simulation framework implemented by IFAG, which comprises two principal subsystems, the radar sensor model and the environment model including a limited ADS and vehicle dynamics model.

Furthermore, Figure 18 illustrates the utilization of standardized interfaces, including ASAM OSI and the FMI 2.0 standard, within the interconnected radar sensor model (FMU) and the IPG CarMaker simulation environment.

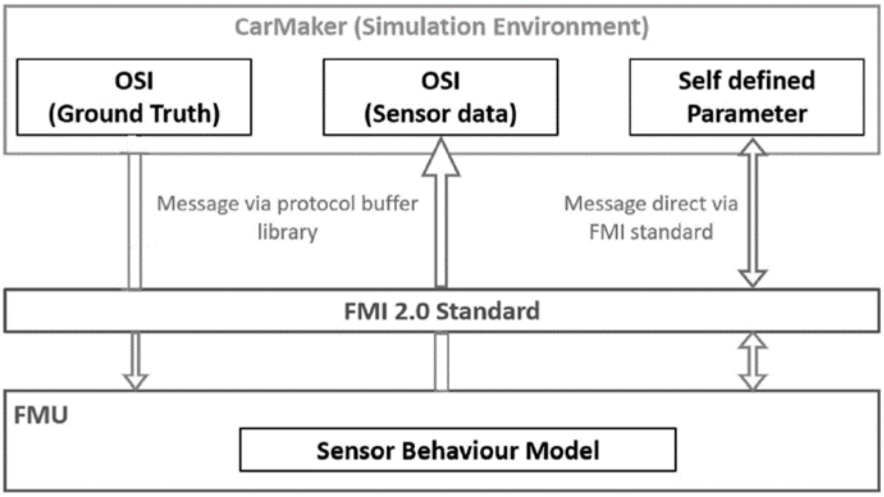

Figure 18: Standardized interfaces used within the interconnected subsystems

3.5 SISW simulation framework approach

As part of the overall SUNRISE SAF, the simulation framework of Siemens enables two safety workflows (processes):

1. Test Execution of Scenarios from SCDB
2. Exploration of unknown-unsafe scenarios

These workflows share many common simulation blocks and therefore, the core part of their designs is identical as shown in the figure below.

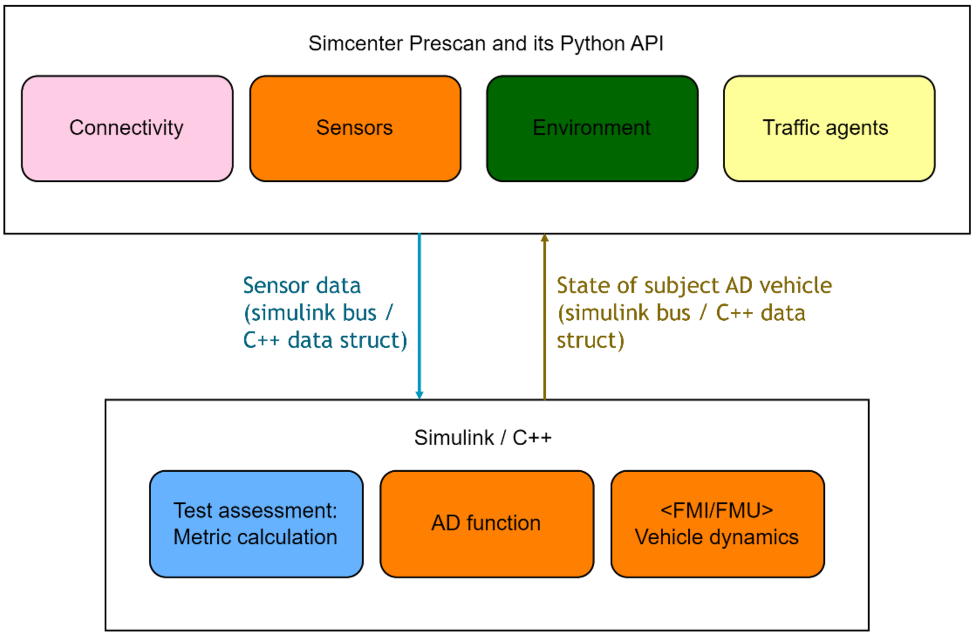

Figure 19: Design of "Test Execution of Scenarios from SCDB"

- Simcenter Prescan – Handling visualization, environment modelling, traffic agent behaviour, & sensor output
- Simulink / C++ – The environment where co-simulation with Prescan and implementation of user algorithms are enabled & metrics for safety critical assessment are being calculated.
- Interfacing with Simulink / C++ allows further data exchange for capabilities of systems such as Connectivity, Traffic Agents, and AD Function. As an example, a ROS based AV stack could be coupled with Simulink or C++ to perform AD Function.
- Model validation (sensor validation, vehicle dynamics model validation, etc.) and switching & configuring SUT are the steps which are performed during preparation phase of the models manually.
- For the orchestration and distributed execution of simulation blocks, Simcenter HEEDS is being used, which is a tool from Siemens with functionality including process automation, optimization studies, and data visualization.

The initial difference between the workflows occurs on the input levels, where the first workflow expects a set of concrete scenarios defined by standardized scenario definition (.xosc, .xodr) and HEEDs manages these inputs to run consecutive Prescan simulations. Outputs and KPIs are later fed back to HEEDs to evaluate simulation outputs. For example, some tests in C++ have been conducted where a logical scenario defined in OpenScenario file is being parsed and concretized to get a set of concrete scenarios, which are then ran by adjusting the speed of ego vehicle, initial position of any actor, lightning condition, or intensity of precipitation in a defined respective parameter range.

Contrary to the first workflow, the second workflow needs to explore for unexpected, unknown, and unsafe scenarios based on a defined ODD & recorded data as can be seen in the architectural diagram below. This requires an additional SOTIF assessment block. Therefore, the second workflow can take ODD, recorded data / OpenScenario, OpenDrive as inputs to extract a search space for exploration of scenarios.

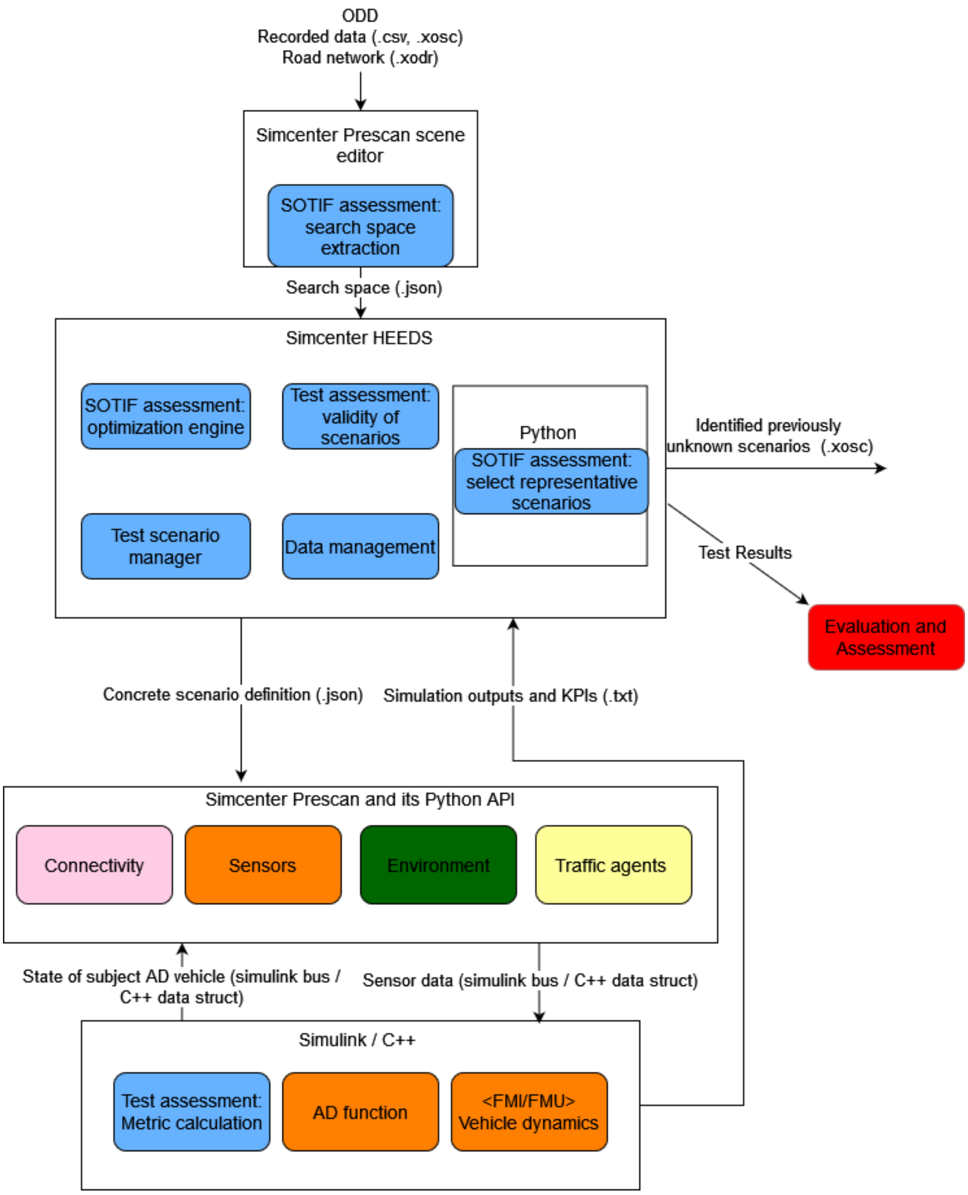

Figure 20: Design of "Exploration of Unknown-Unsafe Scenarios"

Following that, multi-objective optimization function is defined in HEEDs, which searches for unknown-unsafe scenarios within the search space starting from some default conditions. The optimization study aims at efficiently finding scenarios that are highly unknown-unsafe in large search spaces. In each step of the optimization study, a new scenario from the search space is extracted, executed, and the results are used to guide the choice of the next scenario. In the end, the selection of the newly discovered unknown-unsafe scenarios can complement the SCDB for future testing of the system under test.

To test the developed second workflow, sensor data has been collected on a specific intersection by which the overall path and speed profile of vehicles could be extracted. Next, road network in OpenDrive form and ODD are being provided to HEEDs, which then successfully found many potentially unsafe situations for the selected intersection.

3.6 UoW simulation framework approach

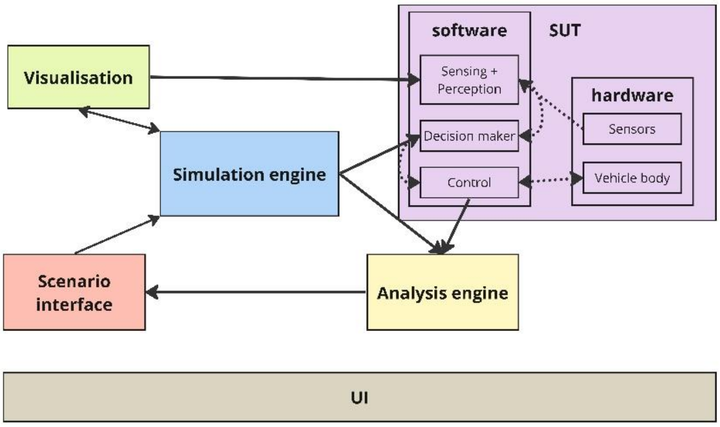

Figure 21: Modularised simulation framework for scenario-based safety assurance workflow in UoW

As seen in Figure 21, the simulation framework at UoW utilises a modularised approach, incorporating interchangeable components within the framework. The primary objective of developing this modular virtual testing framework is to enable the implementation of the SAF detailed earlier with a focus on virtual testing aspects (both purely virtual and XiL).

To align with the flow order as indicated in Figure 21, the **Scenario interface** block is the first block. Upon retrieving scenarios, the Scenario interface layer interprets the scenario semantics embedded within the scenario format and converts them into execution logic. This execution logic is then passed to the **Simulation engine**, which focuses on accurately simulating both the scenery (e.g., road layout as defined in the ASAM OpenDRIVE) and the behaviours (e.g., vehicle manoeuvres as defined in the ASAM OpenSCENARIO XML). When incorporating the **Visualisation** element, the ground truth data can be passed from the Simulation engine to render high-fidelity graphics.

Based on the level of system/sub-systems in the testing target, the simulated sensor data from the Visualisation can be used by the **perception and sensing** layer of the **SUT** to make decisions and actuations. Alternatively, the ground truth data from the Simulation engine can be used directly by the decision-making layer, or the simulated sensor data from the visualisation can be used for sensor emulation technologies to interact with the SUT **hardware**. The **Analysis engine** block then receives data from both the SUT and the simulation engine to optimize the scenario parameter values with the goal of driving the system towards failure. Lastly, the optional **UI** layer allows end users to interface with the testing framework in a more intuitive manner.

Figure 22 below shows the instantiation of the simulation framework at different levels, from left to right: pure planning level, SiL, HiL (hardware and human in the loop).

Figure 22. instantiation of the UoW simulation framework at different system/subsystem testing level

3.7 VIF simulation framework approach

Virtual Vehicle is part of UC1.1 “Urban AD Validation (Perception testing)”. The V&V framework from ViF follows a modular structure and is based on the subsystems defined in D4.1 “Report on relevant subsystems to validate CCAM systems” [2]. It consists of the following items:

1) Test case manager

2) Subject Vehicle – AD-Function

3) Subject Vehicle – Vehicle Dynamics

4) Subject Vehicle - Sensor

5) Environment

However, in contrast to D4.1 [2] the Environment simulation also contains the traffic simulation (see below). Moreover, the subsystem connectivity is not included as the focus of UC1.1 is perception testing. The neutral co-simulation platform Model.CONNECT^{TM} is used as Test case manager. It offers the possibility to integrate different tools and to define test cases for automatic execution of Scenarios. The Software has been described in detail in D4.3 “Report on CCAM simulation tool landscape” [1]. Figure 23 shows the co-simulation topology of the ViF framework in Model.CONNECT^{TM}

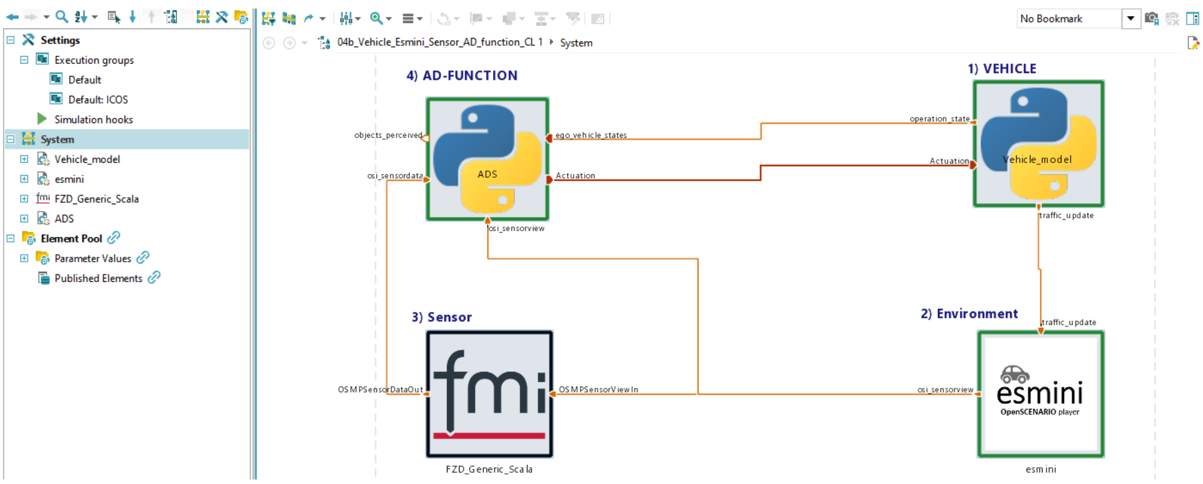

Figure 23: Co-Simulation topology in Model.CONNECT

The modular structure allows both the testing of AD functions and sensor models. Thus, it can be easily used for perception testing. OpenScenario files, including OpenDrive files, are used to execute specific scenarios. The scenarios focus on left and right turn manoeuvres interacting with an opponent car. Figure 24 shows a left turn manoeuvre simulated with the framework. The red car in this figure is the subject vehicle controlled by the AD function, the gray car is controlled by esmini based on the OpenScenario file.

Figure 24: Simulation of a left turn maneuver.

AD-Function and Vehicle:

A python module named RoMPaC (Robust Motion Planning and Control) has been developed by ViF to simulate and develop ADS-Functions together with a vehicle model. For the sake of modularity, RoMPaC has been split into AD-function simulation and vehicle dynamics simulation.

The vehicle subsystem receives from the AD function actuation signals: gas pedal, brake pedal, and steering angle while the AD-function receives the operation state of the vehicle. The current velocity of the ego vehicle is taken from the environment simulation (*SensorView*). However, the AD-function algorithm uses the imperfect object list *SensorData* from the sensor model.

Environment:

esmini is used as a minimalistic environment simulator as there is no need for a photorealistic environment simulation. Moreover, esmini includes SUMO (Simulation of Urban MObility) which is used for the traffic agent simulation. esmini supports ASAM OpenSCENARIO to simulate the dynamic content. However, the vehicle subsystem determines the ego vehicle's actual position. It is packed into the *traffic_update* signal which is received by esmini. In return esmini has as output the ideal object list of the environment *SensorView*.

Sensor:

A highly parameterizable generic perception sensor and tracking model is used for the sensor simulation. The outer layer of the model is the Modular OSMP (OSI Sensor Model Packaging) Framework by FZD (Institute of Automotive Engineering of TU Darmstadt). It specifies ways in which models using the Open Simulation Interface (OSI) are to be packaged for their use in simulation environments using FMI 2.0. The sensor model is coupled with the environment simulation and AD function.

Co-Simulation Settings:

Table 1: Co-Simulation Settings UC1.1 ViF

subsystem	Coupling Step Size	Trigger Sequence	Extrapolation
Vehicle	**0.02s**	**1**	**ZoH**
Environment	**0.02s**	**2**	**ZoH**
Sensor	**0.02s**	**3**	**ZoH**
Vehicle dynamics	**0.02s**	**4**	**ZoH**

Table 1 shows the chosen co-simulation settings for the framework. All subsystems have the same coupling step size and sequential scheduling is used to reduce coupling errors. The execution order of the subystems is defined by the Trigger Sequence. It starts with the vehicle dynamics subsystem and then calculates sequentially in clockwise order. Furthermore, zero-order hold (ZOH) is chosen as a simple coupling algorithm for extrapolating the input from the last to the first subsystem.

4 DESIGN OF THE HARMONISED V&V SIMULATION FRAMEWORK

This chapter deals with the design of the harmonised V&V simulation framework after the individual partner designs have been presented in the previous chapter. The chapter starts with the motivation for a harmonised solution in chapter 4.1 and then recaps the subsystems that were proposed in D4.1 [2] (see chapter 4.2). After that, the partner designs have been analysed in terms of their data formats (chapter 4.3) and interfaces (chapter 4.4). All this leads to an approach for a harmonised V&V simulation framework in chapter 4.5.

4.1 Purpose of the harmonised V&V simulation framework

When designing a system like SUNRISE **Safety Assurance Framework** (SAF), it's essential to divide the system into multiple modules. Our focus here is on the V&V Simulation Framework. There are several key features that contribute to the overall effectiveness and efficiency of the V&V Simulation Framework.

These features are **interoperability**, **modularity**, **scalability**, and **user-friendliness**. Simulation plays a crucial role in this framework, with its main advantage being repeatability. This ensures that the V&V Simulation Framework can consistently reproduce the same results across multiple runs.

Interoperability is another vital component. It involves the use of standardized interfaces, such as ASAM **OpenSCENARIO**, **OpenDRIVE**, **OSI**, Modelica association **FMI**, and standardized control signals. These interfaces ensure seamless interaction between different subsystems, promoting efficient communication and data exchange.

Modularity refers to the design technique that separates the functionality of the V&V Simulation Framework into independent, interchangeable subsystems. Each subsystem represents one unique part of the V&V Simulation Framework, e.g. System Under Test (SUT), Environment, etc. Modularity allows for easier coverage of more use cases. This means that each subsystem fidelity level can be adapted to specific test needs. It also enables comparability between different subsystems, comparing Low fidelity models to High fidelity models for certain tasks.

Scalability is the V&V Simulation Frameworks ability to handle increased workloads effectively. It involves improving the speed, efficiency, and throughput of the simulation. A scalable system can cover a huge search space, leading to more comprehensive testing and validation. This also means that the simulation should be able to run on local High-Performance Computer or be deployed on a Cloud environment. Scalability leads to quicker adaptations due to faster deployments, it seamlessly integrates into existing processes or pipelines, making it easy to incorporate.

Lastly, **user-friendliness** is a defining feature of any successful system. A user-friendly system is easy to use, set up, and integrate. This not only enhances the overall user experience but also promotes wider adaption of the system.

In conclusion, the effectiveness and efficiency of the V&V Simulation Framework are determined by its simulation capabilities, interoperability, modularity, scalability, and user-friendliness. By focusing on these features, we can design robust and user-friendly systems that meet the evolving needs of the technological landscape.

4.2 Harmonised subsystems

A fundamental part of the SUNRISE SAF is the simulation framework. Figure 25 shows an illustration of the SUNRISE Harmonized V&V Simulation Framework with the identified relevant subsystems. According to deliverable D4.1 “Report on relevant subsystems to validate CCAM systems” [2] the simulation framework contains some main subsystems as presented below:

- **Test case manager:** The “test case manager” subsystem’s main function is to orchestrate the execution of test scenarios in the simulation framework.

- **Environment:** The “environment” subsystem’s main function is to describe the surrounding environment in which the CCAM system under test operates.

- **Subject vehicle:** The “subject vehicle” subsystem includes the “Sensors”, the “AD function”, and the “Vehicle Dynamics”. The “Sensors” is a key element in enabling automated driving systems to provide both reliable vehicle localisation and robust environmental perception of the vehicle's surroundings within its ODD. The “AD function” controls the vehicle's response to achieve desired outcomes. The “Vehicle Dynamics” describes the motion of a vehicle based on specific inputs (e.g., external, and internal forces).

- **Traffic agents:** The “traffic agents” subsystem simulates the behaviour of various types of dynamic elements except for the subject vehicle (SuT).

- **Connectivity:** The “connectivity” subsystem enables communication between vehicles and other actors (pedestrians, cyclists, infrastructure elements in the surroundings).

- **Simulation model validation:** This subsystem is necessary to approve the quality and correlation to the reality of a simulation model.

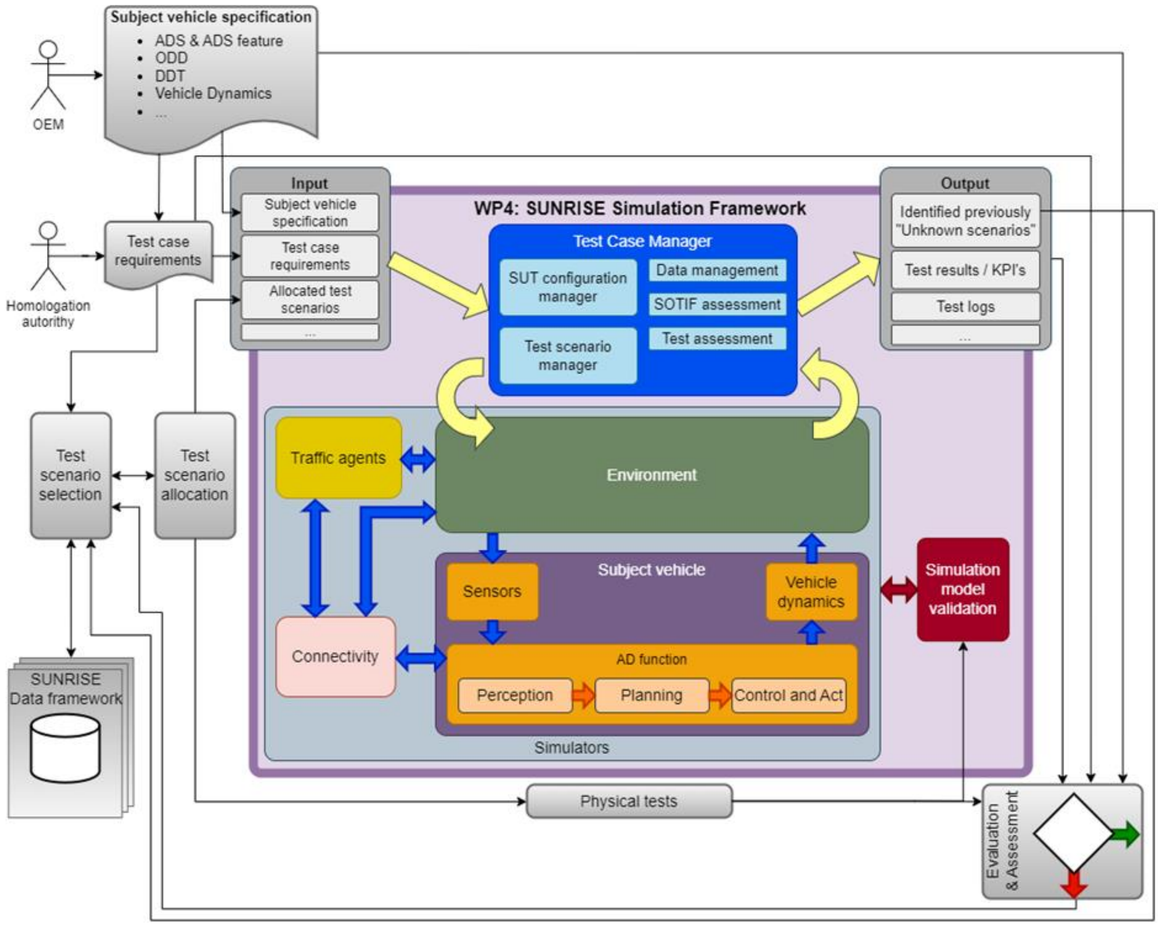

Figure 25: The SUNRISE simulation framework with its subsystems.

4.3 Harmonised data formats

The analysis of the partner frameworks (see chapter 3) provides some **insights that guide the definition of the proposed harmonised data format**. The summary of these insights is presented as follows:

- Simulation frameworks use, in most cases, at least one component as **proprietary tools**. This implies certain steps of the processes are vendor-conditioned, and most likely force users to have some **proprietary data formats** along the chain, especially in **configuration files**, data interfaces, or other metadata. Examples: Simcenter Prescan, Simcenter HEEDS, IPG CarMaker, etc. Note: this does not imply harmonisation is not possible, as the core data contents are standardised and most tools and frameworks already make use of them.

- The most important data contents follow, in most cases, **international standards**, especially from the simulation branch of **ASAM e.V.**, such as ASAM OpenSCENARIO XML, ASAM OSI, ASAM OpenLABEL, ASAM OpenDRIVE.

- **Open source simulators** that are used to execute simulations and to validate data formats, are compliant with open standards. In particular, CARLA, **esmini** and **SUMO** are widely used, given that both CARLA and esmini **are** simulators that offer good compliance with ASAM OpenSCENARIO 1.x scenario and ASAM OpenDrive data formats, **while** SUMO can extend simulations to traffic management level.

- **Co-simulation** is implemented using standard interfaces, such as the FMI (Functional Mock-up Interface), or ASAM OSI (Open Simulation Interface). The co-simulation is necessary to interconnect different simulation models (e.g., different virtual sensors). For connectivity aspects, there is no currently supported standardized interface, however different open source frameworks built on top of VEINS or ns3, such as msvan3t, are used which offer co-simulation capabilities either with SUMO or CARLA or both via protocol buffers or ROS messages.

Next, subsections provide more details on the utilisation of ASAM OpenScenario XML and ASAM OpenLabel as main standardised data formats among the different partner's tools and frameworks.

ASAM OpenSCENARIO XML - Multiple abstraction scenario formats

After generating the scenario content, an adequate scenario description format is used to represent the content and enable its sharing and execution. Several requirements have been identified regarding the scenario format, such as the need for multiple abstraction levels, common and formalised in a single data format, following the specifications of an international standard (to enable effective data exchange and compatibility across different toolchains).

Three different levels of scenario abstractions were originally proposed by Menzel et al. [20] as part of the PEGASUS project [21]. These levels are functional scenarios, logical scenarios, and concrete scenarios. A functional scenario describes the scenario at a semantic level, using linguistic notation to detail the entities and their relationships. Logical scenarios target the scenario content at a state space level, representing entities and their relationships using parameter ranges. Concrete scenarios also target scenario content at a state space level but use specific or concrete values to represent the entities and their relationships.

Additionally, Neurohr et al. [22] recently extended these three levels by introducing a fourth level: abstract scenarios. An abstract scenario sits between the functional and logical scenarios, using a formalized format to organize scenario descriptions, typically adopting a natural language-based expression [23] [22] [20]. Under the new definitions, the functional scenario abstraction is free-form and lacks a formalized format, while the abstract scenario level incorporates a formalized grammar. The BSI Flex 1889 standard sits at the natural language-based abstract scenario description format. Whereas the ASAM OpenSCENARIO XML and OpenDRIVE sit at the concrete and logical scenario level, with the ability to describe also parameter ranges and distributions. Figure 26 illustrates the definitions of different scenario abstraction levels based on Neurohr et al.

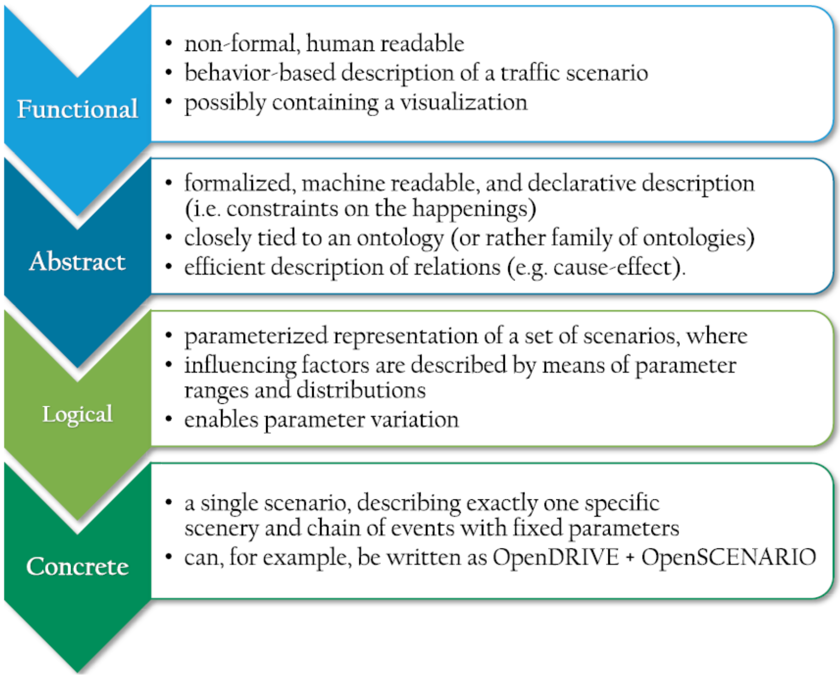

Figure 26: Different scenario abstraction levels with their characteristics

OpenSCENARIO XML and OpenDRIVE are low level xml-based formats that can be executed across a variety of simulators. The drawbacks are that: 1) they are not human readable, and therefore cannot satisfy certain requirements alone, 2) OpenSCENARIO XML covers the dynamic and environmental aspects, and OpenDRIVE covers the road network, this leaves a part of the ODD elements uncovered under neither standard, such as trees and buildings. Although a user could add such additional ODD elements within the object description in OpenDRIVE, such descriptions, from a language point of view, could benefit from further development and alignment with ODD requirements.

ASAM OpenLABEL 1.0

Amongst the data formats required to interoperate a V&V simulation framework, ASAM OpenLabel 1.0.0 plays an important role, especially when part of the simulation is connected to real world sensor data. This is the case when scenario content is derived or inferred from real world data.

When sensor data is collected with instrumented vehicles, processing stages may produce object-level information, such as the position of the ego-vehicle through time (e.g., using RTK-GNSS systems), the position of other dynamic elements like vehicles or pedestrians (e.g., using perception or V2X systems that inform about the presence, type, speed or these objects), or other static elements like lanes, curbs, traffic signs, etc.

From this object-level information, it is then possible to obtain higher-level information useful for V&V processes. In particular, object-level information can be understood as the abstraction of a certain scenario that has occurred in real life. Via semantic processing or scenario extraction techniques, it is possible to recover such scenarios, and then reproduce the same object-level description of the (recorded) reality but executed or reproduced in a virtual world.

OpenLabel 1.0 is an ASAM standard that provides a data model and format for multi-sensor labelling, with a wide range of expressions to represent object-level information (including odometry, calibration, synchronization, complex properties, etc.). Data processing campaigns can produce OpenLabel 1.0 content attached to real world recordings, either automatically or manually. The OpenLabel 1.0 JSON schema provides the reference data model and determines the data format to represent this object-level information that can be fed to scenario extraction processes. In some cases, a direct mapping between OpenLabel 1.0 and OpenScenario is foreseeable, especially when the OpenScenario file represents the instantaneous positions (coordinates) of all involved objects, and thus it is equivalent to OpenLabel 1.0 object-level information.

In addition, OpenLabel 1.0 has another utility: scenario tagging. Using the same JSON schema and underlying data model, OpenLabel 1.0 provides the ability to define tags, either as simple text forms or as complex structures with data properties. Such tagging functionality can be used to add metadata to existing scenario files, and thus incorporate additional searching capabilities to a scenario database. A V&V process may benefit from using OpenLabel 1.0 tags also because the standard defines a taxonomy in the form of an ontology which maps with the BSI/PAS 1883 ODD standard.

In conclusion, the landscape of tools and frameworks for simulation used by the partners offers a heterogeneous panorama with a variety of data formats used along the pipelines. This is natural, as different partners have different needs and purposes for the usage of these frameworks, and thus, the tooling is consequently not the same. Despite the presence of many proprietary applications, all partners have reported the intention and ability to utilise standardised data formats for certain critical data contents (scenarios, maps, labels, sensors). In particular, the utilisation of ASAM e.V. standards seem to be widely adopted, among users, but also among the proprietary tools, which offer export/import capabilities into standardised formats.

This implies that despite the tools and frameworks being proprietary (in some cases) and with varied purposes, the data formats of critical content are harmonized, and effective data exchange and sharing may happen without significant data re-formatting difficulties. Potentially, the work left on harmonization is on the different interpretations of the standard data formats for different use cases, i.e., unfortunately, no standard can completely define to the last detail how users shall define or use the data formats, and some user-defined content is unavoidable.

4.4 Harmonised interfaces

After analysing the interfaces chosen by the consortium partners between each of the subsystems, the following tendencies were observed:

- **Environment**: The majority of the partners choose to either let simulators such as CARLA, Prescan handle it or use ASAM OSI 3.x to directly handle what the sensors will perceive. The first approach requires less configuration, making it simpler to use and enabling faster development times. OSI, on the other hand, implies full access to simulation framework subsystems and allows a lower-level interface, enabling the possibility to have more tailored and customized final products that have the potential to be lighter and more efficient. Similar to CARLA, Prescan also requires less configuration effort with various 3D models in its GUI. Many commercial simulators provide interfaces via Python, C++, or Matlab APIs to set up environment models, which will be then processed internally inside Prescan to actuate and update environment models for sensors and visualization.

- **Sensors**: Many partners picked ASAM OSI 3.x, to go from the *SensorView* provided by the *environment* subsystem to *SensorData.* However, some rely on ROS2 custom messages or C++ data types. Since following the OSI standard in either ROS2 messages or C++ structs or classes is possible, doing that could be a potential way of harmonizing the *sensors* interfaces while keeping the advantages that prompted each partner to choose one or another interface option. However, the limitations of each choice shall be kept in mind to make the final choice:
    - ASAM OSI: It defines the data that can be sent and how it shall be done. This standardization allows to profit from the existing ASAM documentation and its wide-spread knowledge and it may ease the integration of sensors and features, but it may constraint the implementation of certain sensors whose data can't be perfectly represented with the standard.
    - ROS2 and C++: Allow for full flexibility to represent all kinds of data in whichever way is desired (parameter names, types, precision, etc.). This enables a certain degree of future-proofness, since the imposed constraints are minimal. On the other hand, this increased flexibility may make the development process more error-prone and very vulnerable to suboptimal coding standards and badly designed data types.

- **AD Functions**: Most of the partners use ASAM OSI 3.x, while some opted for either ROS2 or C++. Again, the custom ROS2 messages and the C++ approaches could follow the ASAM OSI standard to benefit from its documentation and wide-spread knowledge. However, following a standard reduces the flexibility of the solution. This may be an issue if some AD function uses uncommon inputs or outputs that may not have been part of the use cases considered during the standard development.Co-simulation approaches are also possible here, e.g. integrating a sophisticated AD stack from Autoware with CARLA or running a custom AD stack module and interfacing this with CARLA via ROS bridge.

- **Vehicle Dynamics**: The consortium is divided between those who prefer to rely on CARLA to handle this and those who opt to use their own tools. When using CARLA as the simulation engine, it is not possible to modify the way it interacts with the other subsystems without modifying its source code. The CARLA API takes certain inputs and can provide certain information. When implementing a specific vehicle dynamics module, using OSI's TrafficUpdate is the preferred option.

- **Traffic Agents**: Once again, using CARLA is a wide-spread decision, which handles the traffic agents on its own. To control other traffic agents, it is also possible to override CARLA's traffic manager controller and to implement instead an external controller of your choice Some partners, however, handle them on their own, taking the actor type, position and trajectory as their inputs. For example, Prescan enables API calls to automatically generate traffic agents for a scenario in Matlab or C++ environment. For such an API request, the input option can be OpenScenario or raw trajectory data. For even more complex testing setups, where all traffic agents need to have a re-active behaviour to the behaviour of the ego-vehicle, simulators that allow closed-loop testing like NuPlan have recently been proposed.

- **Connectivity**: Integrating connectivity aspects requires a co-simulation framework where scenarios also include networking elements. The current consensus on the consortium relies on a standardised V2X communication protocols for CCAM applications as these are standardized by ETSI . Custom messages that are not ETSI-compliant can also be used for research purposes as this is one of the goals when testing things in simulation. Open source simulation frameworks allowing co-simulation of communication network and the rest of the system that constitutes a driving simulation system have been reviewed and used in UC1.3. A plugin for a commercial simulator is the Prescan V2X Plugin Toolbox which models communication interaction between different actors in simulation. It is capable of defining effective or statistical ranges for signals and users can choose various types of messages such as ETSI CAM, ETSI DENM and these are then interfaced through C++ or Matlab.

As an additional note, given the widely extended reliance on the CARLA simulator, it is worth mentioning that regardless of its main limitation of reduced customizability in comparison to other lower-level options for the interfaces, it is open source. This means that its source code can be downloaded, edited, and recompiled. This option, however, requires a very in-depth knowledge of CARLA (and sometimes of Unreal Engine) and will most likely require a big investment in terms of development time, so even if it's an option, it may not be viable without expert knowledge and available development and time resources.

To sum up, regardless of the choice of some partners to use either their own solutions built on Python, C++, or Matlab, most decided to build upon already existing tools and standards, such as the CARLA simulator and the ASAM OSI standard. The need of a photorealistic simulator as CARLA may be questionable depending on the needs of the simulations (e.g. validate image-based perception algorithms or not), and given its computational requirements, the use of that kind of tool may or may not be recommended depending on the specific use cases of the framework. It is important however that open source tools typically offer interfaces to the external world (e.g. CARLA-ROS bridge) and hence can easily form a part of a bigger CCAM system.

Regarding interfaces among simulation subsystems, after analysing the needs and decisions of each partner regarding the interfaces between their sub-systems, we determined that the usage of **ASAM OSI standard** would help standardise the approaches without adding any major restriction other than having to follow a standard instead of choose freely the format of the data structures.

To enable efficient interaction, the ASAM Open Simulation Interface (OSI) provides standardised interfaces and message definitions for integration between different systems, subsystems, and simulation environments. Depending on the subsystems used, the OSI ground truth data can be used for the planning aspect, or raw sensor data can also be used when the sensing layer is involved.

In the context of scenario-based testing, a reliable communication interface is crucial to guarantee the accuracy of simulation results. Executing scenarios for a specific ODD, whilst utilising OSI, allows for comprehensive and scalable development and testing of ADSs.

The ASM OSI standard describes how the interface can be used to exchange data between separate models, as depicted in Figure 27.

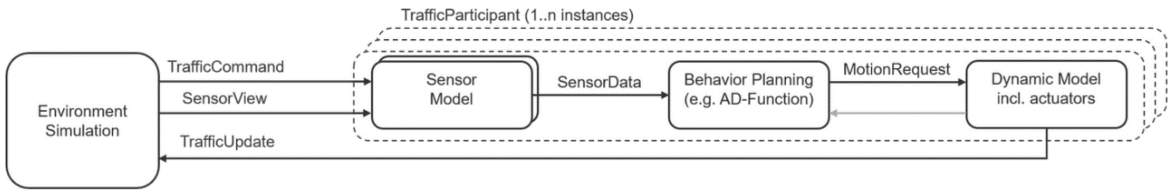

Figure 27: Traffic participant with sensor models, AD function, and dynamic model *[24]*

Table 2 shows an overview of the signals defined in the standard [4].

Table 2: Overview OSI messages

OSI message	Description
SensorView	all information regarding the environment concerning the virtual sensor coordinate system
SensorData	messages that imitate the output of real sensors
TrafficUpdate	updates on the position, state, and future trajectory of a traffic participant
Motion Request	OSI currently provides only limited support for that: the interface can be used to communicate the results of the behaviour planning to the dynamic model, e.g.: gas pedal, (desired) steering angle, …
HostVehicleData	OSI currently provides only limited support for that: the interface describes the measured internal states of a traffic participant. e.g.: actual velocity

4.5 Final harmonised V&V simulation framework

As described in the previous chapter, the consortia are divided into those who rely on a (single) simulator such as Carla or PreScan and those who opt to use their own tools. In the first case, the simulator handles the interfaces and data formats which requires less configuration and makes it easy to use. However, the drawback is the flexibility of the individual subsystems with respect to the signal exchange and implementation. To define a harmonised V&V simulation framework that covers the subsystems defined in D4.1 [2] a single simulator approach is not applicable due to a lack of modularity. Moreover, the harmonised V&V Simulation Framework should be applied by a wide variety of users. Therefore, it must be based on (open) standards and it should be possible to use non-commercial tools. Section 4.4 shows that ASAM OSI is already widely used for the sensor model. Thus, it has been decided that the harmonised V&V Simulation Framework will be built on the following standards:

- ASAM OSI [4]
- ASAM OpenScenario [25]

Figure 28 shows the architecture of the harmonised V&V simulation framework with the subsystems defined in the SUNRISE deliverable D4.1 [2]. It consists of a so-called base layer consisting of 4 interconnected subsystems, namely the **Subject Vehicle – Vehicle Dynamics**, the sensors installed in the vehicle (**Subject Vehicle - Sensors**), the behavioral competencies of the vehicle (**Subject Vehicle – AD function**) and the **Environment** in which the vehicle operates. In this approach, the base layer is the core element that can be harmonized, because these four subsystems are essential for all simulations. That is the reason why it is possible to use standardised interfaces between these subsystems (like ASAM OSI). For the sake of simplicity, the environment block covers the subsystem's Traffic Agents and the Connectivity as there is no standard available for the interface. In addition to the functional interconnection, a Test Case Manager is required for the controlled execution of the targeted test cases and underlying scenarios. Moreover, the KPIs can be calculated by this component.

The framework can be extended by the user in 4 dedicated dimensions related to the **Target Operational Design Domain**, the vehicle **Sensor set-up**, the **Software architecture** and the **Hardware architecture**. The target ODD dimension can be extended to include subsystems that focus more on specific parts like connectivity or traffic agents. The sensor set-up can be extended to represent all sensors used in the subject vehicle from perception sensors to vehicle state sensors. The software architecture follows the classical approach of autonomous driving of “Sense”, “Plan” and “Act” and the hardware architecture can be extended by all necessary vehicle dynamic components to represent the simulated subject vehicle in all details.

The ability to extend the base layer with demonstrator specific elements, supports the general applicability of the proposed harmonised V&V Simulation Framework. This flexibility in the design gives the user the possibility to adjust the harmonised V&V Simulation Framework in a way that it can be used for a variety of use cases. For example, the entire ODD can be built using a few best-in-class environment simulation tools capable of representing specific ODD attributes in all details (high fidelity) and together covering the entire target ODD. This can be useful to reduce the number of simulation tools in the toolchain and therefore, reduce the complexity of the architecture. The same construction principle can be applied to the sensor setup, SW and HW architecture.

It can be seen that the base layer corresponds well to the ASAM OSI standard in Figure 27. The lower layers describe the implementation of each subsystem in the base layer. The model fidelity and the used tools are up to the user allowing a flexible framework, as long as the interfaces of the base layer are respected.

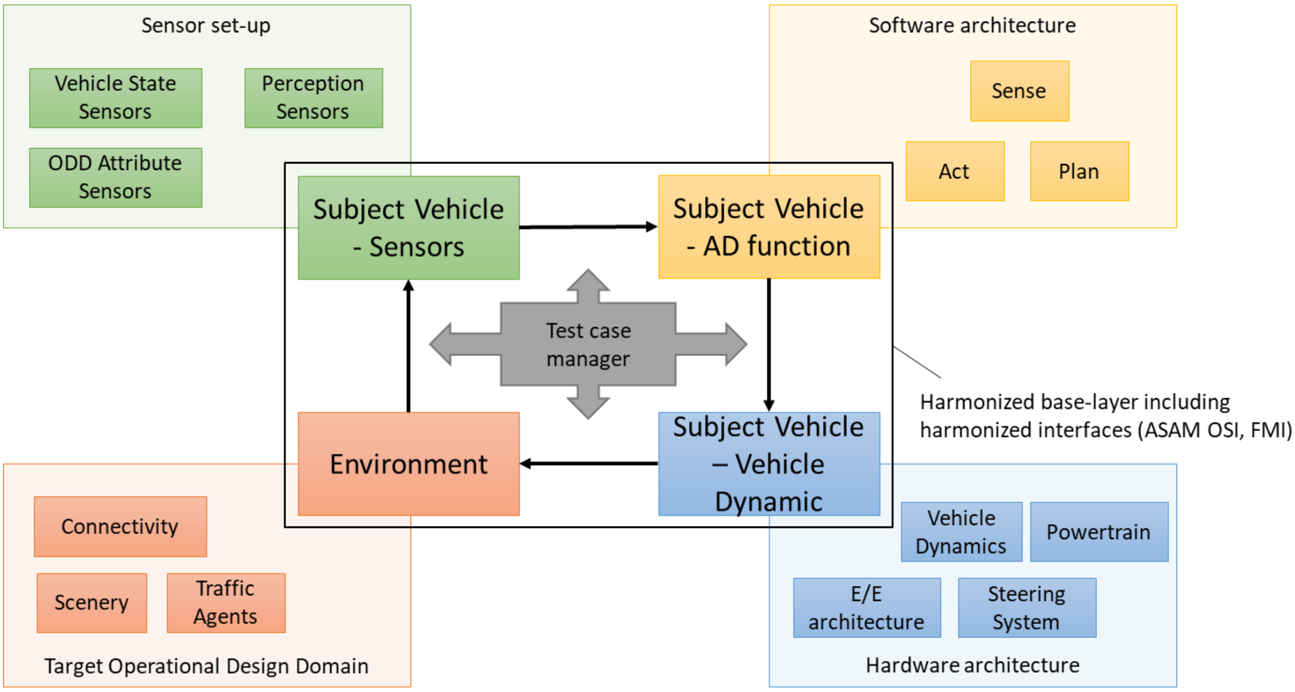

Figure 28: Architecture of the harmonised V&V Simulation Framework

Figure 28 shows an overview of the interface of each subsystem in the base layer. The signals themselves are described in Table 3.

Table 3: Interfaces of the harmonized framework based on the ASAM OSI standard

	Subject Vehicle - Sensor	Subject Vehicle – AD Function	Subject Vehicle - Vehicle Dynamics	Environment
Input(s)	SensorView	SensorData HostVehicleData	MotionRequest	TrafficUpdate
Output(s)	SensorData	MotionRequest	TrafficUpdate HostVehicleData	SensorView

It should be mentioned that the harmonised V&V Simulation Framework can only serve as an initial starting point and may need to be tailored for the specific needs of the use case.

The next step will be developing a validation approach for the harmonised V&V Simulation Framework, which will be the content of the follow-up deliverable D4.5.

4.5.1 Application example of the harmonised V&V Simulation Framework

A possible implementation of the harmonised V&V Simulation Framework using Functional Mock-up Units (FMUs) for each subsystem and the co-simulation platform Model.CONNECT is shown in Figure 29. Model.CONNECT serves both as a co-simulation master and as a test case manager. However, any other tool that can import and execute FMU's could be used as a simulation master, e.g.: Mathwork Simulink. An open-source alternative to run FMU's is OpenMCx [26]. It's a tool-neutral co-simulation framework based on Modelica standards. OpenMCx is a pure command line tool without a GUI, therefore, ModelCONNECT has been used here for better illustration purposes.

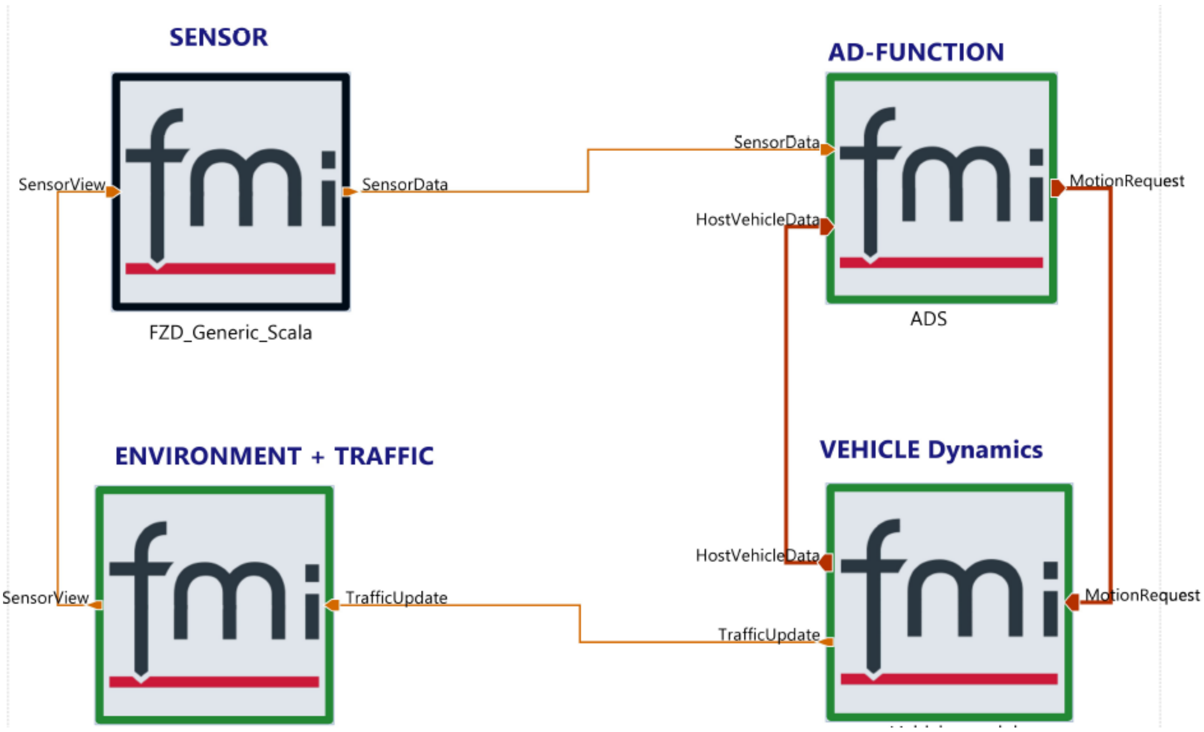

Figure 29: Implementation of the harmonised V&V simulation framework in Model.CONNECT

Another aspect of the framework is the co-simulation settings, which are essentially the following:

- coupling step size/communication time of each subsystem
- scheduling of the subsystems, trigger sequence (execution order)
- extrapolation method of the inputs of each subsystem

In general, sequential scheduling gives better results than parallel scheduling because fewer inputs must be extrapolated. The drawback is that sequential scheduling needs more computation time than parallel scheduling. Since real-time capability is not considered here it is recommended to use sequential scheduling anyway. Of course, sequential scheduling raises the question of the execution order (or better said trigger sequence). As the extrapolation of object-based signals should be avoided (in contrast to physical signals like velocity, temperature, ...) the following trigger sequence is recommended:

1) Subject Vehicle – Vehicle Dynamics
2) Environment
3) Subject Vehicle – Sensor
4) Subject Vehicle – AD Function

For the extrapolation technique, the following rule can be followed:

- discrete signals (e.g.: gear signal) and status signals (on/off, …) →Zero Order Hold
- continuous, non-noisy signals (e.g.: velocity) → First Order Hold

For the coupling step size, the following rule of thumb can be applied:

- the coupling step size is a multiple integer of the internal step size
- the frequency of the coupling step should be 10x bigger than the frequency of the fastest relevant coupling signal, e.g.: if the frequency of a coupling signal is 100Hz, then the coupling step size should be 1ms

More information regarding the co-simulation settings can be found in [19].

5 CONCLUSIONS

This deliverable is about the design of a harmonised V&V simulation framework and the question of which aspects can be harmonised. Since virtual simulation plays a key role in the testing of CCAM systems, it makes sense to use a framework that provides interoperability, modularity and scalability. By using standardised interfaces and common data formats it is easier to exchange available tools according to the current needs and requirements of test cases.

This document covers two parts. In the first part, a summary of related projects and their approaches towards harmonisation was introduced (see chapter 2) as well as the individual partner designs of their simulation frameworks (see chapter 3). These frameworks have been designed within the SUNRISE use cases and have different purposes. For example, some focus more on connectivity, while others center around the perception system. The partner simulation frameworks already indicate the tools that they selected for the design of their simulation framework. However, the choice of tools is not part of this deliverable but has already been described in the document "D4.3 - Report on CCAM simulation tool landscape". This report does not focus on or recommend specific tools, as this is not part of the harmonisation of the V&V Simulation Framework. The aim is not to recommend tools, but to design a harmonised V&V simulation framework that allows users to easily switch between different tools without having to redesign the framework.

The second part of the deliverable (see chapter 4) is about the individual aspects that can be harmonised. The three aspects that were identified are the **subsystems, data formats and interfaces**. The subsystems recommended for the harmonised V&V simulation framework are just briefly described (see section 4.2) since there is a deliverable dedicated to this topic ("D4.1 – Report on relevant subsystems to validate CCAM systems"). To recommend data formats and interfaces the partner simulation framework designs have been analysed. With regard to the **data formats**, partners are following **ASAM formats**. These formats are in line with the approaches of the other projects analysed in chapter 2. This leads to the decision that these formats should be used in the proposed harmonised V&V simulation framework (e.g.: OpenScenario, OpenDrive, ...). In terms of interfaces, it is clear that OSI is the most widely used standard and is therefore recommended for a harmonised framework.

The proposed design of the harmonised V&V Simulation Framework is based on a base layer with the components **Subject Vehicle – Vehicle Dynamics**, the sensors installed in the vehicle (**Subject Vehicle - Sensors**), the behavioral competencies of the vehicle (**Subject Vehicle – AD function**) and the **Environment** in which the vehicle operates. The framework can be extended by the user in 4 dedicated dimensions related to the **Target Operational Design Domain**, the vehicle **Sensor set-up**, the **Software architecture** and the **Hardware architecture**. Possible extensions of the environment would be, for example, 'Traffic Agents' and 'Connectivity'. This modular structure offers flexibility to be used in a wide variety of use cases. The interfaces of the components within the base layer can be harmonised using the **ASAM OSI** standard (see chapter 4.5).

The usage of the proposed harmonised V&V Simulation Framework is not mandatory for the SUNRISE Safety Assurance Framework, but it is highly recommended to create more interoperability, modularity and scalability in the virtual testing process. It is also an important input for the WP7 tasks 7.2 and 7.3 which deal with the design and the demonstration of the SUNRISE use cases that aim to validate the SUNRISE Safety Assurance Framework.

The next step will be developing a validation approach for the harmonised V&V Simulation Framework, which will be the content of the follow-up deliverable D4.5

REFERENCES

[1] SUNRISE, “D4.3 - Report on CCAM simulation tool landscape,” 2024.

[2] SUNRISE, “D4.1 - Report on relevant subsystems to validate CCAM systems,” 2023.

[3] H.-M. Heinkel and K. Steinkirchner, “Credible simulation process – SET Level - Simulation-based Engineering and Testing of Automated Driving,” Vols. (Version 1.3, 19.08.2022), 2022.

[4] [Online]. Available: https://report.asam.net/asam-open-simulation-interface-osi. [Accessed 15 07 2024].

[5] [Online]. Available: https://fmi-standard.org/. [Accessed 15 07 2024].

[6] “https://ssp-standard.org/,” [Online].

[7] V. Project, “https://cordis.europa.eu/project/id/101075068,” [Online].

[8] “https://pearsinitiative.com/,” [Online].

[9] “ISO/TR 21934-1:2021 (Road vehicles — Prospective safety performance assessment of pre-crash technology by simulation — Part 1: State-of-the-art and general method overview”.

[10] “ISO/CD TS 21934-2 (Road vehicles — Prospective safety performance assessment of pre-crash technology by simulation — Part 2: Guidelines and requirements for application)”.

[11] V. Framework, “https://v4safetyproject.eu/media/pages/outputs/dba34ee736-1713785034/ms10_draft_prospective_safety_assessment_framework_method_v12.pdf,” [Online].

[12] S. Solmaz, G. Stettinger and F. Wotawa, “Residual risk management strategies at system level presented for acc/lka behavioural competencies,” in *2023 IEEE International Automated Vehicle Validation Conference (IAVVC)*, 2023.

[13] G. Stettinger and W. F., “Smart monitoring for safety-assurance in autonomous driving,” in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, 2023.

[14] D. Project, “https://divp.net/,” [Online].

[15] S. Project, “https://www.sakura-prj.go.jp/project_info/,” [Online].

[16] “https://github.com/ms-van3t-devs/ms-van3t,” [Online].

[17] “https://github.com/ucla-mobility/OpenCDA,” [Online].

[18] “https://github.com/veins/veins_carla,” [Online].

[19] P. Weissensteiner, G. Stettinger, J. Rumetshofer and D. Watzenig, “Virtual Validation of an Automated Lane-Keeping System with an Extended Operational Design Domain,” *Electronics 2022.*

[20] T. Menzel, G. Bagschik and M. Maurer, “Scenarios for Development, Test and Validation of Automated Vehicles,” in *IEEE Intelligent Vehicles Symposium*, 2018.

[21] “https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf,” PEGASUS Method An Overview, 07 2024. [Online].

[22] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle and R. Galbas, “Criticality Analysis for the Verification and Validation of Automated Vehicles,” *IEEE Access,* vol. vol. 9, pp. pp. 18016-18041, 2021.

[23] G. Bagschik, A. Reschka, T. Stolte and M. Maurer, “Identification of potential hazardous events for an unmanned protective vehicle,” *2016 IEEE Intelligent Vehicles Symposium (IV),* pp. pp. 691-697, June 2026.

[24] [Online]. Available: https://opensimulationinterface.github.io/osi-antora-generator/asamosi/latest/interface/architecture/architecture_overview.html. [Accessed 15 07 2024].

[25] [Online]. Available: https://www.asam.net/standards/detail/openscenario/. [Accessed 15 07 2024].

[26] [Online]. Available: https://github.com/eclipse/openmcx. [Accessed 17 07 2024].