PRISSMA Project
Plateforme de Recherche et d'Investissement pour la Sûreté
et la Sécurité de la Mobilité Autonome
04/2021 - 04/2024

[L6.6] WP6.3: EVIDENCE TO BE PROVIDED ON THE DEVELOPMENT PROCESS, WITH IDENTIFICATION OF POSSIBLE COVERAGE OF APPROVAL REQUIREMENTS

WP6.3 : Preuves à fournir concernant le processus de développement, avec identification de la couverture possible d'exigences d'homologation

Main authors: J-F. MARLIERE (UTAC), L. HABIB (AIRBUS PROTECT), E. ARBARETIER (AIRBUS PROTECT), M. GARNIER (ATC), K. Quintero (IRT SystemX), C. Bohn (IRT SystemX).

Keywords: Audit, requirements, AI-based systems, risk

Abstract.
The purpose of this document is to provide evidence in the development process with an identification of possible coverage of the registration requirements. We will therefore discuss the audit system, the coverage issues, and the information that the manufacturer will have to provide in the context of mobility system including autonomous or automated vehicle with AI. Subsequently, we will give some examples of evidence according to the tests carried out and will address the case of the life cycle of AI-based systems.

Résumé.
Ce document a pour but de fournir des preuves dans le processus de développement avec une identification de la couverture possible des exigences d'homologations. Nous aborderons donc le système d'audit, les enjeux de couvertures et les informations que le constructeur devra fournir dans le cadre d'un système de mobilité comportant des véhicules autonomes ou automatisés avec de l'IA. Par la suite nous donnerons quelques exemples de preuves selon les tests effectués et aborderons le cas du cycle de vie des systèmes à base d'IA.

1 GENERAL SPECIFICATIONS

Given the complexity of artificial intelligence-based systems, it is necessary to supplement the performance requirements and regulatory tests by manufacturer documentation demonstrating that the artificial intelligence-based systems are free of unreasonable safety risks to vehicle occupants and other road users in the relevant scenarios and during the systems lifetime. In this respect, it is necessary to lay down the safety management system to be put in place by the manufacturers, to set for manufacturers and authorities the parameters to be used for the traffic scenarios relevant for artificial intelligence-based systems, to lay down criteria to assess whether the safety concept of the manufacturer addresses the relevant traffic scenarios, hazard and risks, and to set out criteria to assess the validation results from the manufacturer in particular validation results from virtual toolchains. Finally, it is necessary to specify the relevant in-use data that shall be reported by the manufacturer to the authorities.

2 DEFINITIONS

‘fault’ means an abnormal condition that can cause a failure. This can concern hardware or software.

‘failure’ means the termination of an intended behaviour of a component or a system of the artificial intelligence-based system due to a fault manifestation.

‘lifetime of the artificial intelligence-based systems’ means the period of time during which the artificial intelligence-based system is available on the vehicle.

‘minimal risk manoeuvre (‘MRM’)’ means a manoeuvre aimed at minimising risks in traffic by stopping the vehicle in a safe condition (i.e. minimal risk conditions).

‘minimal risk condition (‘MRC’)’ means stable and stopped state of the vehicle that reduces the risk of a crash.

‘operational design Domain (‘ODD’)’ means operating conditions under which a given artificial intelligence-based system is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.

‘object and event detection and response (‘OEDR’)’ means subtasks of the dynamic driving task that include monitoring the driving environment and executing an appropriate response. It includes detecting, recognizing, and classifying objects and events and preparing and executing responses as needed.

‘unreasonable risk’ means the overall level of risk for the vehicle occupants and other road users which is increased compared to a manually driven vehicle in comparable transportation services and situations within the operational design domain.

‘scenario’ means a sequence or combination of situations used to assess the safety requirements for an ADS.

‘nominal traffic scenarios’ means reasonably foreseeable situations encountered by the ADS when operating within its ODD. These scenarios represent the non-critical interactions of the ADS with other traffic participants and generate normal operation of the ADS.

‘critical scenarios’ means scenarios related to edge-cases (e.g. unexpected conditions with an exceptionally low probability of occurrence) and operational insufficiencies, not limited to traffic conditions but also including environmental conditions (e.g. heavy rain or low sunlight glaring cameras), human factors, connectivity and miscommunication leading to emergency operation of the ADS

‘functional safety’: absence of unreasonable risks under the occurrence of hazards caused by malfunctioning behaviour.

‘operational safety’ means the absence of unreasonable risk under the occurrence of hazards resulting from functional insufficiencies of the intended functionality (e.g. false/missed detection), operational disturbances (e.g. environmental conditions like fog, rain, shadows, sunlight, infrastructure) or by foreseeable misuse/errors by the vehicle occupants and other road users (i.e. safety hazards — without system faults).

‘CoC’ means Certificate of Conformity

‘CPNCO’ means Car to Pedestrian Nearside Child Obstructed

‘CBLA’ means Car to Bicycle Longitudinal Adult

‘CPFA’ means Car to Pedestrian Farside Adult

3 SYSTEM AUDITING

3.1 Introduction

Within the scope of describing the evidence that should be provided when justifying AI-based systems approval, the following subsections tackle the aspect of auditing for a vehicle: it’s scope, the mandatory elements as well as the limitations of the auditing process.

It is fundamental to state that the auditing process certifies the proofing elements that the system provider documents. In this sense, it is mandatory for the auditing process to cover the design process and the safety concepts considered and applied therein in order to verify that the design covers factually the risks and hazards that the system could encounter.

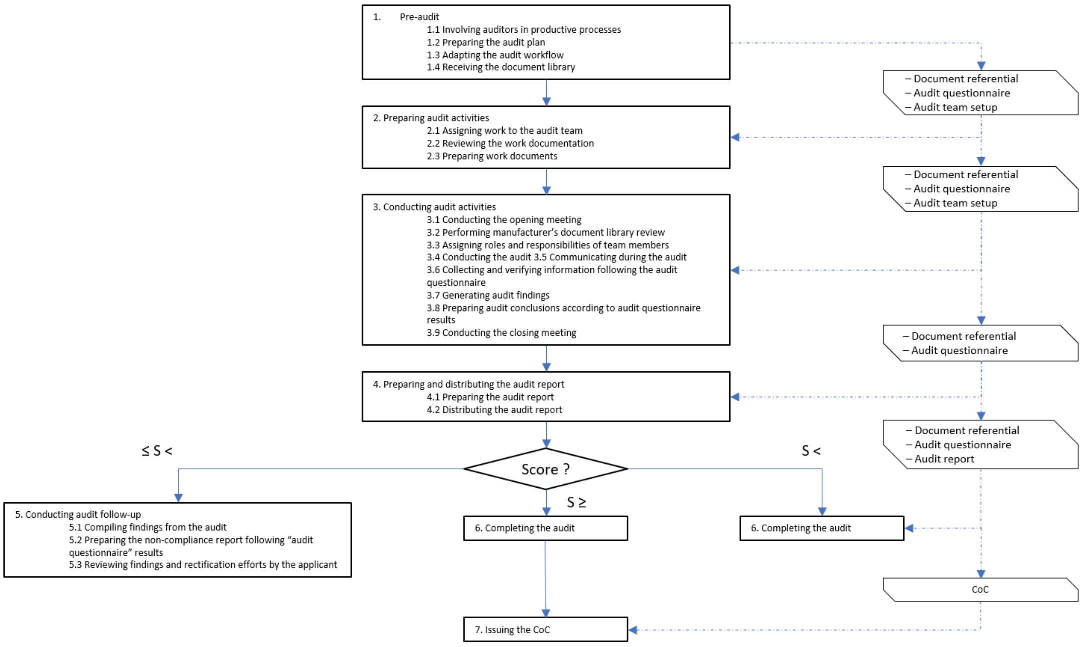
Further in this section performance validation is addressed which is necessary but only pertinent when having considered the auditing process within the scope of system design.
Through the safety lens, performance evaluation does not constitute a standalone proof in itself for the system’s safety. Performance validation will however show, as stated by the system provider or a pertinent third party, that the system’s behavior is as expected given its specifications.

3.2 Auditing process

The auditing process shall be carried out by a designated technical service under the supervision of the authorities (stakeholders, certified bodies…).

The authority may, at its discretion, request to attend the audit, and may request any information from the technical service regarding the audit process, evaluation, findings, proceedings, etc. When the confidential nature of some information generates specific clauses in the audit plan, the authority shall be subject to the same clauses.

A visual description of the auditing process is given by the flow chart below:



The chart below gives the requirements of the audit process steps and resources:

Audit phase	Start time	Resources requirements (man-days)	Remarks
1. Pre-audit	Day 1	1 day 1 person: Lead auditor	
2. Preparing audit activities	Day 1	1 to 2 days 2 people: Lead auditor / Management systems expert	Time determined by the audit plan made with the man ufacturer's agreement.
3. Conducting audit activities	After 1. and 2. After 5. when applicable	3 to 4 days 2 people: Lead auditor / Management systems expert	Time determined by the audit plan. If the audit is a follow-up audit, 2 to 3 days may be sufficient
4. Preparing and distributing the audit report	After 3.	21 days 1 person: Lead auditor	21 days is the longest delay allowed for the technical service to distribute the audit report. Fewer than 21 days may be required in practice to produce the reports (audit questionnary, partner report, audit report..)
5. Conducting the audit follow-up	After 4. Only when applicable	3 month delay 2 people: Lead auditor / Management systems expert	May take more than 3 months depending on the manufacturer's rectification process.
6. Completing the audit	After 4. Only when no follow-up	1 day 1 person: Lead auditor	If the final score is lesser than 80, the process ends here and any new attempt at homologation shall begin from phase 1.
7. Issuing the CoC	After 6. Only if the audit was passed	1 day Lead approver	If the audit is successful, issuance of a certificate.

3.3 Audit requirements

Operational Design Domain (Speed, road type, country, Environment, Road conditions, etc), Boundary conditions/ Main conditions for Minimum risk manoeuvres and transition demands Basic Performance (e.g. Object and Event Detection and Response (OEDR)…)

Supervision centre (if relevant))
The means to activate, override or deactivate the system by the human supervision centre (if relevant), passengers (if relevant) or other road users (if relevant)

For this part, a general description of the system must be added.
A meticulous breakdown of the system's functions, including detailed specifics of the variables involved, the limits of its operation, as well as the Human-Machine Interface (HMI) concept, especially at the boundaries of ODD. This also includes an explanation of the data processing required for learning algorithms.

The exposition of the system architecture, highlighting its components and their respective roles, internal and external connections, signals exchanged between them, as well as elements not physically present on board but necessary for its operations.

A detailed presentation of the manufacturer's security concept, including preventive and corrective measures put in place to ensure safe and reliable operation, as well as the validation procedure for these measures.

3.4 Coverage stakes

The subject of coverage is crucial when providing evidence of proper guidelines being followed in the development process of AI-based systems.

In the case of systems in the mobility sector, the subject of coverage is tightly related to *pathway management*, *use cases*, as well as *environment*. These 3 angles inherently impose requirements that need to be addressed in the design phase as well as the associated risks. Within this context, parameter coverage is crucial in the process of safety demonstration, i.e. to ensure that all situations to be encountered are properly handled by the system.

In order to address the issue of coverage, the main challenges and stages for this include:

- Ensuring that the constraints linked to pathway management, use cases and environment are properly identified.
  This entails that justification of the *pathways choice strategy* must be provided in coherence with the system’s capabilities and the strategies implemented if such pathways were to be managed by the system and potential impact to nominal conditions for road users.
  Choice strategy for use cases should also be documented as well so that they can reflect, through scenario modelling, the expected system’s behaviour with regards to design.

- Ensuring that the design process (and subsumed safety concepts) cover the aforementioned constraints
- Ensuring the implementation’s compliance with the specifications.

<u>On the accountability for coverage:</u>

In terms of accountability, ensuring coverage is the responsibility of the system provider. The auditing process should perform verifications on control points in the process as well as the results obtained by the provider, including verification of the safety demonstration of the system. The manufacturer must be accountable of the coverage of the testing process.

The PRISSMA method provides tools allowing to provide proofing elements that can be implemented either by the system provider or by the technical service through controlled tests either on tracks or on simulation.

In the case of automated driving on open roads, the scenario approach is necessary and complementary to conventional methodologies in order to guarantee a meaningful, relevant and representative coverage of situations that the vehicle could encounter. This is, a sufficient enough coverage such that reasonably foreseeable risk scenarios are treated. This approach is necessary both in validation as well as in the design phase. Deliverable 1.4 of the PRISSMA project in section 4.2 [1] explicitly addresses means for generating scenarios to ensure coverage which are linked to the system requirements and the ODD. This includes nominal scenarios (coming from the design phase and the intended purpose of the system), and scenarios from accident, risk and driving analyses.

At this stage, there is no existing global methodology to demonstrate the safety of AI-based systems. Deliverable 2.7/section 3.5 [2] of the PRISSMA project provides elements on a possible optimum in the evaluation phase.

<u>Coverage levels</u>

Several levels should ensure that the relevant situations are properly covered when developing (including the entire development cycle) AI-based systems for AVs.

Throughout the entire process, and depending on the system being developed, suitable coverage criteria and the subsequent coverage rates should be studied and documented.

In a top-down view coverage should be addressed in the set of scenarios chosen, as well as the actors considered as relevant (which can include surrounding actors considered as *outlookers* that could have an impact on the AI-based systems, sub-systems or components), the environment, all static and dynamic parameter ranges considered (for such actors and conditions) and finally for all instantiations of these parameter ranges for testing.

The following example illustrates how through the scenario approach, the idea of coverage can be apprehended and then quantified when performing validation so that the coverage criteria and coverage rates can be documented for validation. The example addresses (Figure 1) a subset of functional scenarios modelled for an AV (one of PRISSMA's Proofs of Concept) that are considered as covering for a specific service. These functional scenarios 'cover' from a high-level perspective the key situations and interactions of the system with its surroundings, including Dynamic Drive Tasks (DDT). This initial coverage from the highest level will lead to a subsequent decomposition and coverage from the logical scenario perspective, see Figure 2 for the whole coverage tree. This decomposition should ensure the coverage of all:

- Dynamic Driving Tasks (DDT) and the subsequent subtasks,

- strategic planning tasks,
- Fallback tasks,
- Relevant environment characteristics (weather, road geometry and other features or states, static or dynamic objects, luminosity, among others) included in the system's ODD.

With respect to the parameter variation at the actors level as well as the environment in logical scenarios, the choice of these intervals as well as the strategy for instantiation of these parameters in the testing phase will subsequently determine the coverage rate that should be documented for audit. It is then crucial that such strategies in the scope of the scenario approach are properly documented from the design phase to that verification and validation activities feed results back to these structures in coherence to the intended purpose.

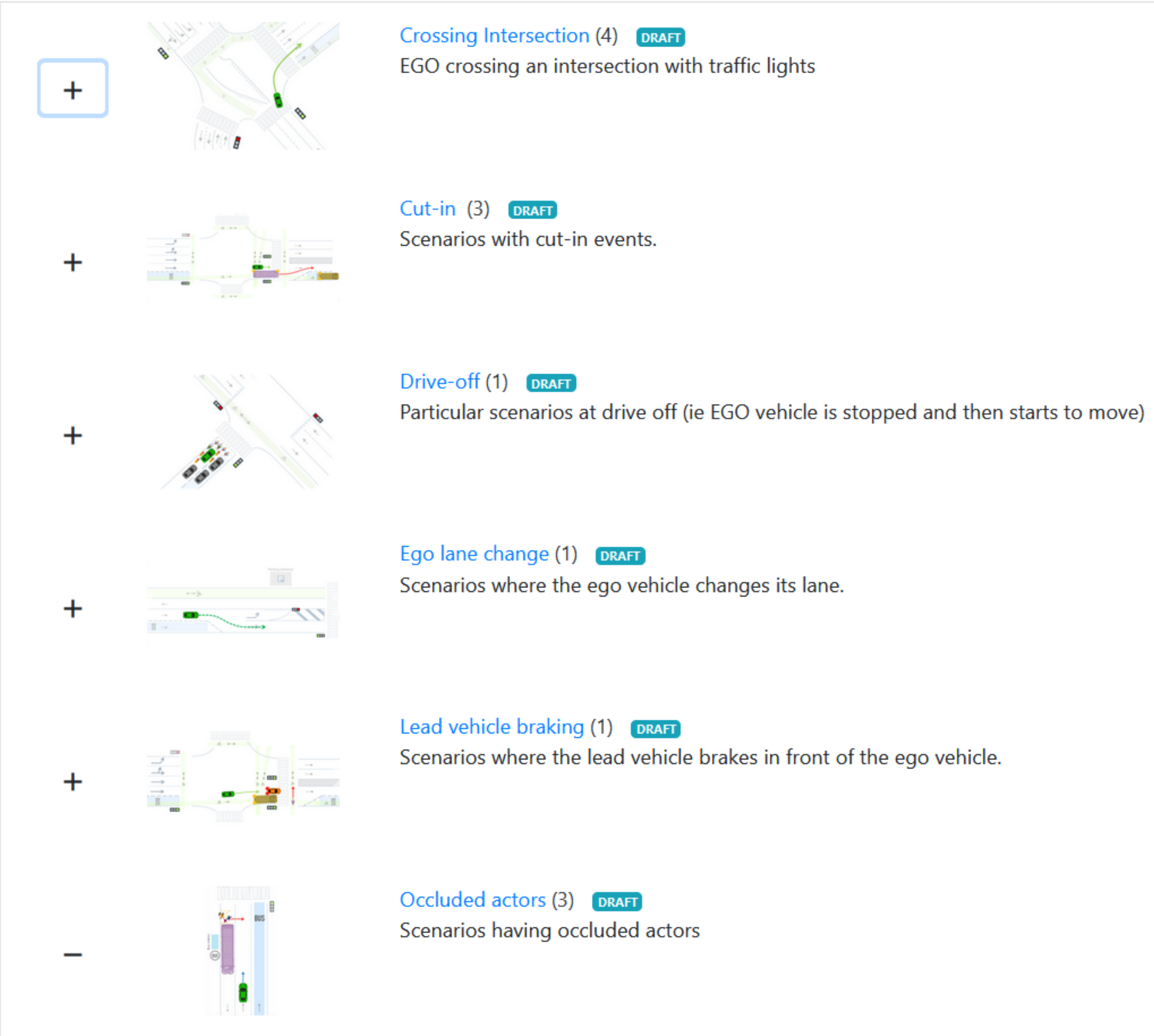


Figure 1. Subset of functional scenarios (PRISSMA Project POC)

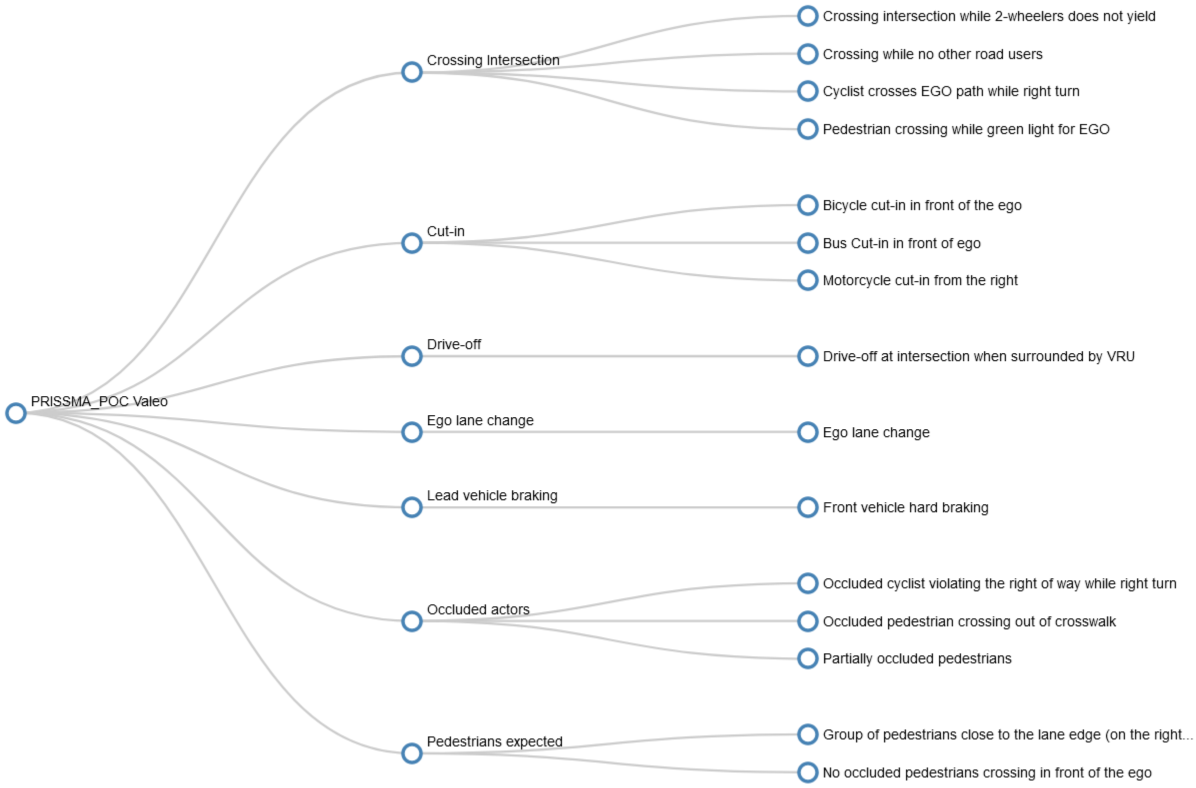


Figure 2. Coverage tree for all functional and logical scenarios (example - POC Valeo in the PRISSMA project)

3.5 Information document

We can define the main characteristics of an AI based system. In a risk-based approach and use cases approach, general-purpose AI systems are treated separately due to the breadth of their potential use cases.
A Risk classification can be developed for a system because of the AI law. It establishes different risk categories and requirements for each of these categories.

- All AI systems will need to be inventoried and assessed to determine their risk category and the resulting responsibilities.
  - Prohibited systems are systems that pose an unacceptable risk to safety, security, and fundamental rights of individuals will be prohibited for use in the EU.
  - High-risk AI systems are the systems that will be subject to the majority of requirements, including the establishment of risk management and quality systems, data governance, human oversight, cybersecurity measures, post-market surveillance, and maintenance of required technical documentation (additional requirements may be specified in later regulations on AI for healthcare, financial services, automotive, aviation, and other sectors).
  - Minimal-risk AI systems do not require any additional obligations.

- High-risk systems will need to undergo a compliance assessment to prove their conformity before being placed on the market (The application of harmonized standards, currently under development, will allow AI system providers to demonstrate their conformity through self-assessment. Third-party conformity assessment, conducted by an accredited independent evaluator "notified body", may be required)
- Another way to to assess High-risk systems is to implement support measures for innovation, such as Regulatory "sandboxes" (also called TEF). They will allow various European actors to innovate, experiment, test, and validate the compliance of their AI systems with the AI law in a safe environment.

To assess conformity, competent national authorities can designate third-party "notified bodies" to carry out conformity assessments of AI systems. In the case of internal conformity assessment processes, the supplier, distributor, importer, or any other third party must demonstrate conformity through a "self-assessment-based evaluation," i.e., through self-certification. These entities must conduct three main checks :

- Verify that the implemented quality management system complies with the requirements stated in Article 17 of the EU AI act
- Review the information contained in the technical documentation to determine if the requirements for high-risk AI systems are met
- Ensure that the design and development process of the AI system and its post-market surveillance (Article 61 of the EU AI act) comply with the technical documentation

Based on the coverage stakes, the following chart presents the necessary information to provide to detail the capabilities of the vehicle. This information document is adapted from Regulation 2022/1426

Item	Response :
AUTOMATED DRIVING SYSTEM (ADS): yes/no	
AI Embedded : yes/no	
1. General ADS description: …	
1.1. Operational Design Domain/Boundary conditions: …	
1.2. Basic Performance (e.g. Object and Event Detection and Response (OEDR), planning, etc.)): …	
2. Description of the functions of the ADS:	
2.1. Main ADS Functions (functional architecture): …	
2.1.1. Vehicle-internal functions: …	
2.1.2. Vehicle-external functions (e.g. backend, off-board infrastructure needed, operational measures needed): …	
3. Overview major components of the ADS	
3.1. Control units: …	
3.2. Sensors and installation of the sensors on the vehicle: …	
3.3. Actuators: …	
3.4. Maps and positioning: …	
3.5. Other hardware: …	
4. ADS layout and schematics	
4.1. Schematic system layout (e.g. block diagram): …	
4.2. List and schematic overview of interconnections: …	
5. Specifications	
5.1. Specifications in normal operation: …	
5.2. Specifications in emergency operation: …	
5.3. Acceptance criteria: …	
5.4. Demonstration of compliance: …	
6. Safety concept	
6.1. Manufacturer statement that the vehicle is free from unreasonable risks: …	
6.2. Outline of the software architecture (e.g. block diagram): …	
6.3. Means by which the realization of ADS logic is determined: …	
6.4. General explanation of the main design provisions built into the ADS to generate safe operation under fault conditions, under operational disturbances and the occurrence of conditions that would exceed the ODD: …	
6.5. General description of failure handling main principles, fallback level strategy including risk mitigation strategy (minimum risk maneuver): …	
6.6. Conditions for triggering a request to the on-board operator or the remote intervention operator: …	
6.7. Human machine interaction concept with vehicle occupants, on-board operator and remote intervention operator including protection against simple unauthorized activation/operation and interventions: …	
7. Verification and validation by the manufacturer of the performance requirements including the OEDR, the HMI, the respect of traffic rules and the conclusion that the system is designed in such a way that it is free from unreasonable risks for the driver, vehicle occupants and other road users: …	
7.1. Description of the adopted approach: …	
7.2. Selection of nominal, critical and failure scenarios: …	
7.3. Description of the used methods and tools (software, laboratory, others) and summary of the credibility assessment: …	
7.4. Description of the results: …	
7.5. Uncertainty of the results: …	
7.6. Interpretation of the results: …	
7.7. Manufacturer's declaration: The manufacturer(s) … affirm(s) that the ADS is free of unreasonable safety risks to the vehicle occupants and other road users.	

8. ADS data elements	
8.1. Type of data stored: …	
8.2. Storage location: …	
8.3. Recorded occurrences and data elements: …	
8.4. Means to ensure data security and data protection: …	
8.5. Means to access the data: …	
9. Cyber security and software update	
9.1. Cyber security type-approval number: …	
9.2. Number of the Certificate of Compliance for cyber security: …	
9.3. Software update type approval number: …	
9.4. Number of the Certificate of Compliance for software update: …	
9.4.1 Information on how to read the RxSWIN or software version(s) in case the RxSWIN is not held on the vehicle.	
9.4.2 If applicable, list the relevant parameters that will allow the identification of those vehicles that can be updated with the software represented by the RxSWIN under item	
10. Operating manual (to be annexed to the information document)	
10.1. Functional description of the ADS and expected role of the owner, transport service operator, on board operator, remote intervention operator, etc.: …	
10.2. Technical measures for safe operation (e.g. description of the necessary off board infrastructure, timing, frequency and template of maintenance operations): …	
10.3. Operational and environment restrictions: …	
10.4. Operational measures (e.g. if on-board operator or remote intervention operator needed) :	
10.5. Instructions in case of failures and ADS request (safety measures by vehicle occupants, transport service operator, on board operator and remote intervention operator and public authorities to be taken in the event of malfunctioning of the operation): …	
11. Means to enable a periodical technical inspection: …	
12. Definition of embedded AI	
12.1 Name and address of the provider and/or authorized representative	
12.2 List of AI systems covered under the same quality management system.	
12.3 technical documentation for each AI system covered under the same quality management system.	
12.4 documentation concerning quality management.	
12.5 procedure in place to ensure quality management system.	
12.6 written declaration that the same application has been lodged with any other notified body	
12.7 : type of AI (none, symbolic AI, supervised/semi-supervised/unsupervised, reinforcement learning)	
12.8 : safety functions (perception, planning, actuation)	

4 TYPOLOGY OF EVIDENCE

4.1 Introduction

AI-based systems, used in automotive products, may allow a compromise of various model characteristics: model drift and staleness, model complexity, robustness, verifiability, predictability and overfitting etc. while guaranteeing a certain level of safety and security.

Further recurrent evaluations might be necessary to check whether the provisions regarding software updates (in the recommendations on uniform provisions concerning cyber security and software updates) adequately address updates of AI-based systems. AI-based systems can contribute to improve vehicle safety, with additional beneficial consequences on road safety, e.g. by allowing AD systems to predict currently unforeseeable behavior of other road users (e.g. detection of potential collision opponents). The use of AI and machine learning algorithms in type approved functions is limited for the time being. Even though, there are already well-established processes for how to test conventional software before and during deployment of an automotive product those processes might not be sufficient for AI based software.

With this consideration, in a supervised learning AI vehicle, we will see what is expected from the manufacturer.

A Trajectory prediction using drivable path prediction from labelled data must also be part of the driving functions.

As for non-driving functions, detection of the driver eye gaze for DMS, fault detection and predictive maintenance must be present.

4.2 Performance requirements

Under the AI Act, companies have several actions to consider. They must Inventory all developed or deployed AI systems and determine if any fall within the scope of the AI Act. They have also to identify the applicable requirements. Additionally, compliance must be integrated into all functions responsible for AI systems throughout their value chain and lifecycle. Finally, they must develop and implement a plan to ensure that frameworks for responsibility and governance, risk management and control systems, quality management, monitoring, and documentation are in place at the time the law comes into force.

This section lists all the specific requirements dedicated to AI in an automated driving system:

- The system equipped with AI shall demonstrate anticipatory behavior in interaction with other road user(s), in order to ensure stable, low-dynamic, longitudinal behavior and risk minimizing behavior when critical situations could become imminent, e.g. with unobstructed and obstructed vulnerable road users (pedestrians, cyclist, etc.) or with other vehicles crossing or cutting-in in front of the fully automated vehicle.
- The AI based ADS shall detect and respond appropriately to objects and events relevant for the DDT within the ODD
- The AI based ADS shall be able to detect the risk of collision with other road users, or a suddenly appearing obstacle (debris, lost load) and shall be able to automatically perform appropriate emergency operation (braking, evasive steering) to avoid reasonably foreseeable collisions and minimise risks to safety of the vehicle occupants and other road users.
- The protection of other human life outside the AI based fully automated vehicle shall not be subordinated to the protection of human life inside the fully automated vehicle.
- The AI based ADS shall recognize its ODD conditions and boundaries of the ODD and it must recognize exits from the ODD.
- The manufacturer shall have processes to manage the safety and continued compliance of the AI based ADS over lifetime (wear and tear of components especially for sensors, new traffic scenarios, etc.).
- The AI based ADS shall support software updates.

The following chapters deal with the tests that are used to validate the requirements. The entire test sequence to be conducted is then divided into three Work Packages (simulation, controlled environment and real-conditions), which make a complementarity between the tests.
Firstly, simulation tests for the definition, specification of requirements, and necessary characteristics for the use of tests in simulation environments, and the constraints to adhere to in order to extract "acceptable evidence" for a certification process.
Then, controlled environment tests for the definition, specification of requirements, and characteristics concerning the use of tests in controlled environments (including the importance of integrating infrastructure into this process).
And finally, real environment tests for the implementation of tests in real-world conditions, prior to authorizing the deployment (and commercial operation) of the autonomous or automated mobility system/service. This latter point is connected to a safety demonstration approach for an autonomous or automated mobility service.

4.3 Simulation campaign report

4.3.1 Introduction

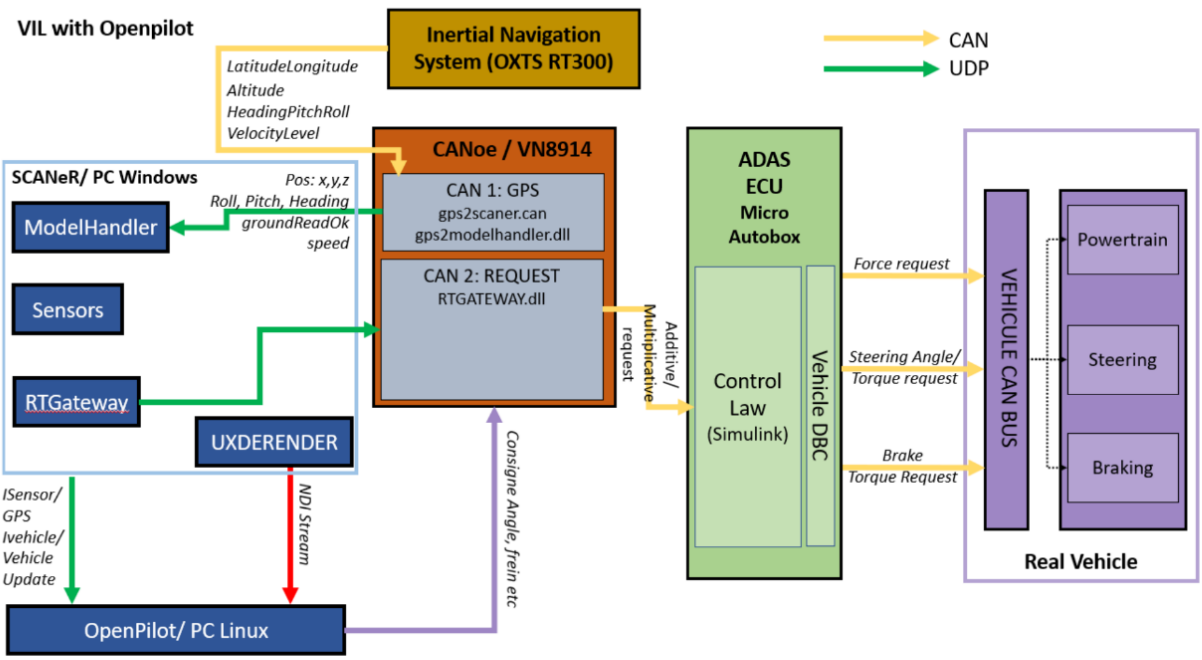
This section aims to propose procedures and protocols for the evaluation and validation ofcritical AI-based systems and subsystems from the evaluation environments. It was tested with several POCs adding a level of complexity linked to the applications (AI-based systems, system of systems, communication and cyber-security systems). Evaluation and validation procedures are pivotal for critical AI-based systems.
The validation process is based on the following steps :

- Test run preparation phase
- Test run execution
- Test run compilation

Two different virtual testing methods were used: the first one is Model-In-the-Loop, fully virtual and the second one is the Vehicle-In-the-Loop (VIL) which combines virtual and physical. The aim of the POC is to see how VIL architectures simulation can be used and what they can bring in validation loop of AI-based system. Testing it with a POC helps discussing about methodology of validation, giving real examples as benefits of simulation and challenges the tools (software and hardware) to improve them so that they can better fit with the AI testing and validation issues. As UTAC and AVSimulation are not AI system developers, the study relies on testing an external AI system. The one used in this POC is Openpilot from CommaAI which is an open-source automated driving software. Openpilot is an automated driving system of level 2. It has an Adaptive Cruise Control (ACC) and a Lane Centering Assist (LCA) to drive the vehicle, an AI algorithm is also embedded for object detection and classification.

The Vehicle-In-the-Loop at UTAC is a test facility that enables the ADAS of a real vehicle to be tested in a virtual environment. It can be used to test any type of driving aid (AEB, LSS). It makes it possible to carry out more complex and dangerous scenarios and to increase test productivity. An inertial measurement system is used to synchronise the position and speed of the virtual vehicle with those of the real vehicle. The CAN messages (longitude, latitude, altitude, etc.) are recovered and converted into x,y,z, etc. So that they can be read by SCANeR studio, the simulation software developed by AVSimulation. We then place virtual sensors on the virtual vehicle. The MicroAutoBox (MAB) from DSpace acts like an ADAS ECU and sends the requests directly to the vehicle's actuators.

After integrating Openpilot with SCANeR, the second part was to integrate Openpilot in the Vehicle-In-the-Loop test facility. Originally, VIL system is based on SCANeR studio for the virtual environment and a MicroAutoBox is in the trunk to allow to load control/ADAS systems we want to test. So the main issue is how we can send information from Openpilot to the MicroAutoBox. We needed to create a new python scriptbridge. In this script, messages and data from Openpilot are converted to CAN messages in order to send them directly to the MicroAutoBox. Then in the MAB, a simple Simulink model is in charge to transmit the right requests to the ECU.



4.3.2 Evaluation objectives

When implementing this POC, the main aim is about testing the methodology by analysing three main aspects:

- Analysing the representativity of the methodology: By comparing similar scenarios in fully virtual simulation and with the VIL, a score of representativity can be computed. Such criteria can be established by comparing measured data between virtual and real scenarios. These can be the trajectories of the ego, the gas and brake pedal command sent by Openpilot or the detected distances between targets and the ego.
- Studying the repeatability of the algorithm: Using controlled scenario by using fully virtual simulation or VIL architecture let us test the repeatability of Openpilot. Indeed, the same scenario can be replayed and then the responses of Openpilot can be tested. As the AI relies sometimes on random calculation, we could expect that results obtained with Openpilot can have differences for a same scenario. These differences can be quantified.

- Measuring the precision of the AI: Since the AI of Openpilot is used to get perception and take decision, some of the steps of the algorithm can be evaluated. An example is comparing detected distances between what Openpilot detects and the real ground truth distance that can be obtained from SCANeR studio. The Openpilot algorithm works with a messaging library (cereal) that let us extract some log information (such as the measured distance) so that we can evaluate it to validate the system.

4.3.3 Scenario definition

In order to have a complete process of validation of the AI, there is a need of generating the corresponding scenarios to play with Openpilot. The aim of these generated scenarios is to bring the objectives of representativity and coverage.

Scenarios used for this POC were :

- Test plan scenario
- Long drive scenario
- Standard validation scenario

	Test plan scenario	**Long drive scenario**	**Standard validation scenario**
Scene	Infrastructure: Portion of N104 Actors: Ego and leading vehicles	Infrastructure : Straight long road Actors: Ego and diverse type of objects (cars, pedestrians, obstacles)	Infrastructure: Highway Actors: Ego and leading vehicle
Event	The leading vehicle does a cut-out manoeuvre at the middle of the scenario	Objects are crossing in front of the ego vehicle	The leading vehicle is braking from 50km/h to 0km/h
Action	The ego vehicle must correctly evaluate the distance to the targets and adapt the cruise speed	The ego vehicle should detect the objects and brake to avoid collision	The ego vehicle must brake to avoid collision
Criteria	The estimated distances and speed must be accurate and no collision should happen.		

4.3.4 Scenario configuration, algorithm, and ground truth

Scenario configuration: For each example of this POC, the scenarios have been configured in SCANeR Studio to respect a defined ODD and OEDR. The terrains designed in SCANeR Studio respect the highway ODD of the three cases. Varying the weather condition has been done using the weather parameters of SCANeR to choose between normal weather, snowy weather or foggy weather.

Algorithms: Relying on Unreal Engine to compute the camera images lets us test the computer vision part of Openpilot and the robustness of the bridge between SCANeR Studio and Openpilot.

Ground truth: The analyzing tools of SCANeR Studio (and using the export channels) lets us extract the correct position and distances between vehicles. These measurements are the core values that represent the ground truth which is compared with what Openpilot evaluates.

4.3.5 Scenario execution and evaluation metrics for the three generated scenarios

Test plan generated scenario For the scope of this POC, we studied one scenario where the ego follows a vehicle that does a cut-out maneuver at the middle of the scenario. A specific architecture has been implemented to replay and activate Openpilot the same way during several executions. The test plan replayed 7 times each scenario while varying the weather condition (one normal condition, one fog condition and one snow condition). The resulting detection shows disparity in distance precision regarding the weather conditions.

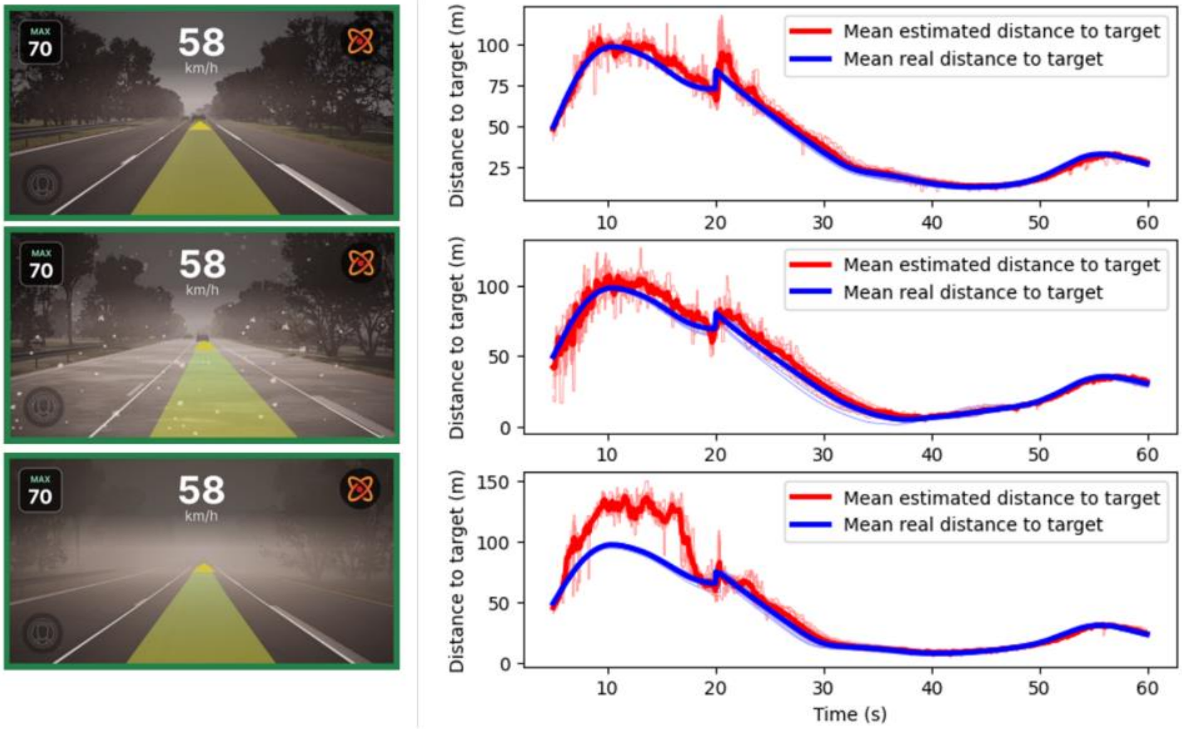


Figure 142: The measured distances by Openpilot for several replays (in blue : the real distance, in red : Openpilot's estimation)

When replaying 10 times the same scenario, the behavior of Openpilot is slightly different,hence a demonstration of the lack of repeatability of the AI in this case. Table 12 shows theobtained metrics for 10 replays for each weather condition. This kind of obtained results should be tested for each scenario case to ensure a complete validation of the AI with fixed threshold to validate or not the system.

Table 12: Obtained metrics concerning the distance to target for a test plan generated scenarios with 10 replays (only the first 4 replays are displayed)

Environment	Normal					Snow					Fog				
Accuracy (%)	3.0	2.4	2.4	3.0	...	5.4	12.1	24.3	7.4	...	11.4	10.9	12.3	11.4	...
	Mean : 2.9					Mean : 10.7					Mean : 11.8				
RMSE (m)	4.6	3.6	4.7	4.8	...	6.3	6.7	5.6	6.7	...	16.5	16.1	15.5	17.1	...
	Mean : 4.55					Mean : 6.51					Mean : 16.35				
Max err (m)	20.5	18.9	25.2	29.9	...	29.0	46.4	30.1	52.4	...	52.0	60.2	57.4	58.3	...
	Mean : 25.9					Mean : 36.1					Mean : 57.5				
Std Dev (m)	2.52					4.12					3.52				

An important part of the work still is to establish the scenario database to do such tests (theODD must be precisely defined for Openpilot and the coverage of the corresponding scenarios) and to fix the corresponding threshold to validate the system. Moreover, the number of replays to be done to have a reliable enough result should be correctly quantified. In the case of this POC, doing 7 replays is sufficient to show that the variability is wide.

Long drive scenario

Another test has been done with Openpilot on a long drive (in the virtual N104 terrain) where the driving system has been tested in front of different actors to test the perception efficiency of Openpilot. Such a scenario lets the system explore the diversity of situations that can occur. In this case, the chosen test lets Openpilot meet different kind of objects.

Table 13: Obtained metrics concerning detection behaviour of Openpilot for different objects

Detected object	**Detection**	**Detection max distance**	**Braking decision**	**Distance to brake**
Peugeot 308 (Grey)	✔	86 m	✘	-
Holden Astra (Red)	✔	87 m	✔	46 m
Opel Vivaro (Spring green)	✔	79 m	✔	52 m
Citroën C3 (Green)	✔	86 m	✔	51 m
Farm cow	✔	123 m	✘	-
Farm horse	✔	106 m	✘	-
Sewer plate	✔	120 m	✘	-
Tarmac hole	✘	-	✘	-
Speed limit sign	✘	-	✘	-
Speed bump	✘	-	✘	-
Crosswalk markings	✘	-	✘	-

Table 13 shows the kind of criteria to evaluate the perception of different objects. Through a long drive scenario, the list of objects to be detected can be diversified and the resulting detection distances can be more accurate through numerous replays.

Standard validation scenario

Another scenario studied is from the Euro NCAP protocol about Assisted Driving Grading lately published in January 2023. We chose the Car-to-Car Rear braking scenario where a leading vehicle is driving at 50kph and the ego vehicle is following with the ACC engaged.

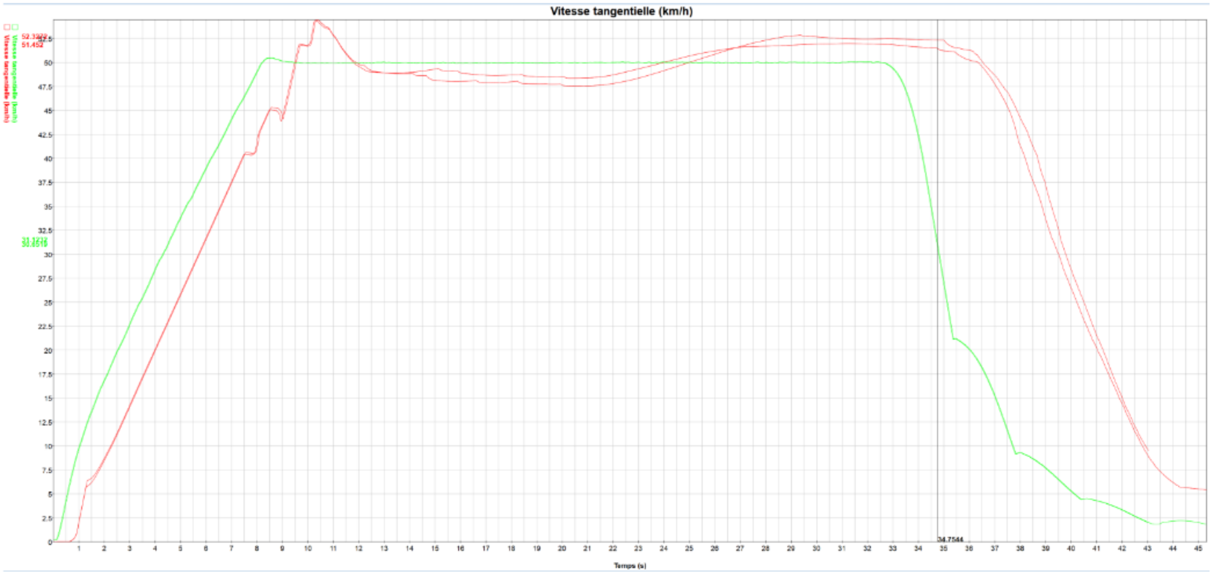


Figure 143: Ego Speed comparison in red, in green the target vehicle speed

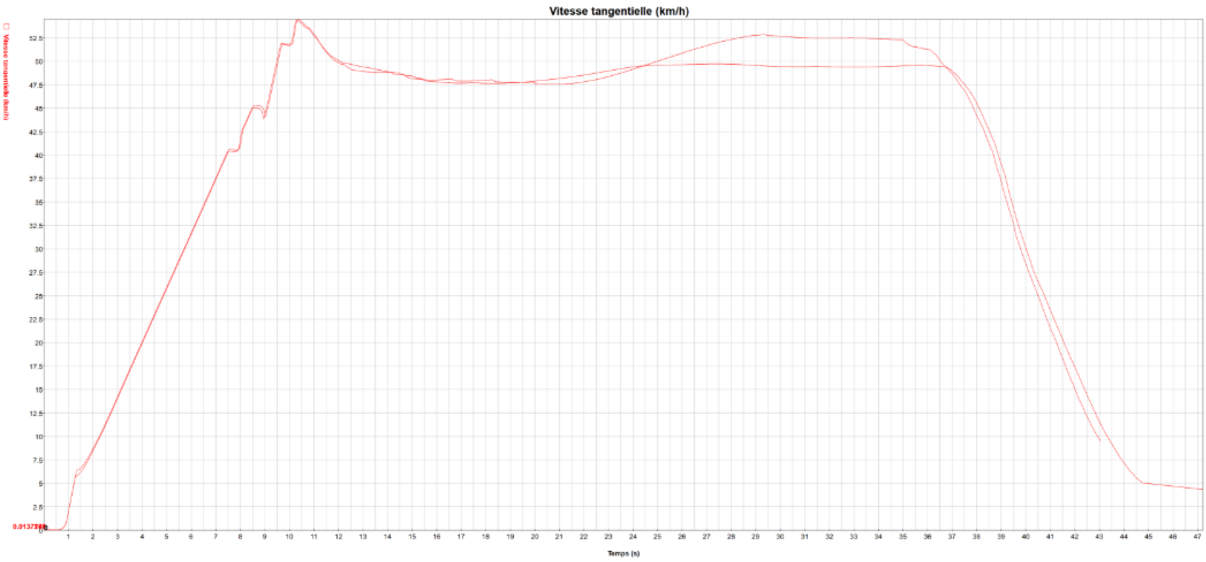


Figure 144: Ego speed comparison in red

On the figures above the ego speed is displayed in red, they represent three different runs. We can observe big differences in the figure 144 when the ego vehicle is approaching the leading vehicle, these differences lead to a lower impact speed for the best case. The same thing happens in figure 143 but the gap is smaller. Then, for the second test scenario, the Openpilot behavior is different despite the test case is exactly the same.

4.4 Controlled environment report

This chapter is an illustration by which the manufacturer can present his tests results.

4.4.1 Test program

4.4.1.1 Protocol version

The following scenarios refer mainly to the ENCAP 2023 protocol for the geometry, the corridors, and the test speeds.
Then, some variants of known scenario have been created for this study.

4.4.1.2 Tests description

Refer to deliverable **L3.2** shared in January 2023.
The scenarios have been divided in 4 categories:

- Repeatability
  This category allows to evaluate the repeatability of the systems, it means to perform many times the same test, in the same conditions and check if the behavior is the same for all the repetitions.

- Robustness
  This category allows to evaluate the robustness of the systems, it means to perform a specific scenario and change different parameters (Speeds, Angles, visual aspect…) and see the behavior.

- Pre-critical
  This category allows to evaluate the anticipation of the systems on existing scenarios or new ones.

- Random
  This category allows to evaluate the systems in new random scenarios, unknown by the systems, and check the feasibility and the relevancy of it.

4.4.2 Vehicle under test

Three different vehicles have been tested during this campaign.

Golf 8 :

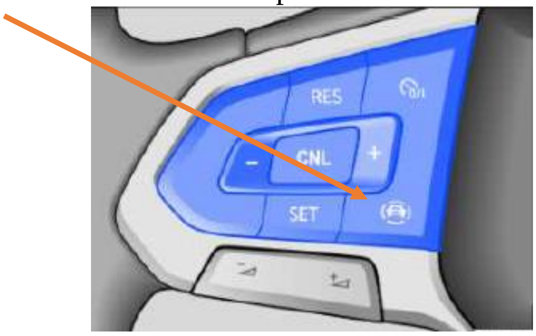
Valeo Robot Taxi Drive4U

Zoe Nexyad :

With all the vehicles, a first step of subjective testing has been done to have a first idea on the behavior of each vehicle.

The Golf is equipped with the ADAS system called “Travel Assist” allowing to anticipate some situations.
This system combines two driver assistance functions, Adaptive Cruise Control (ACC) for longitudinal assist and Lane Assist for lateral assist.

This function is activated by a button on the multifunction steering wheel, which therefore triggers longitudinal speed assist and lateral position assist.



For safety reasons, the driver must keep his hands on the steering wheel for the guidance to be effective.

To this longitudinal speed guidance can be added an anticipation function. The system calculates the position of the Golf based on GPS and route data from the navigation system and must adapt the speed in advance to the approach of bends, roundabouts, crossings, speed limit zones etc…

At the same time, it uses the traffic sign recognition system via the front camera and must adapt the speed as soon as a limitation is detected.

4.4.3 Testing equipment and track

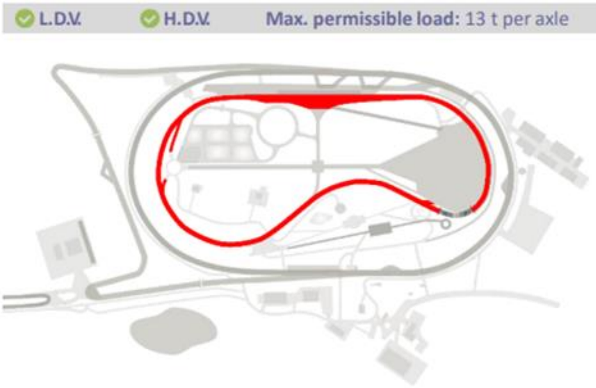
The vehicle is equipped with motion measurements (accelerometers), driving control systems and HMI analysis tools.

Other equipments are road users target, vulnerable road users (adults or child) and road signs.

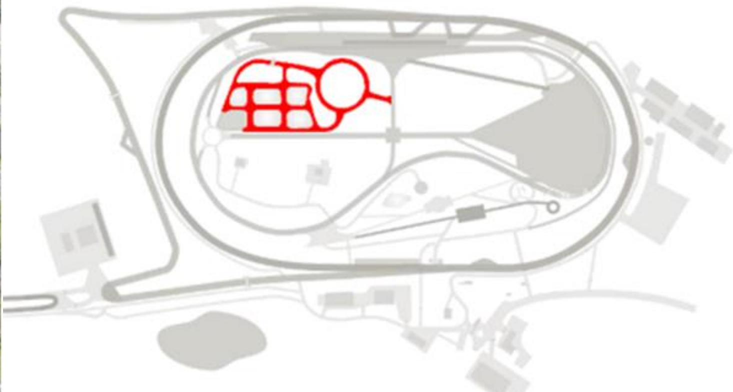


UTAC tracks in Montlhéry :

- TEQMO highway

- TEQMO City :

-

4.4.4 Testing result

4.4.4.1 Post processing

We define the PASS/FAIL as:

– PASS: The system reacted and allowed to avoid the collision
– FAIL: The system didn't react OR reacted too late to avoid the collision

To go further in the analysis, we check the following values in the raw data (.txt file):

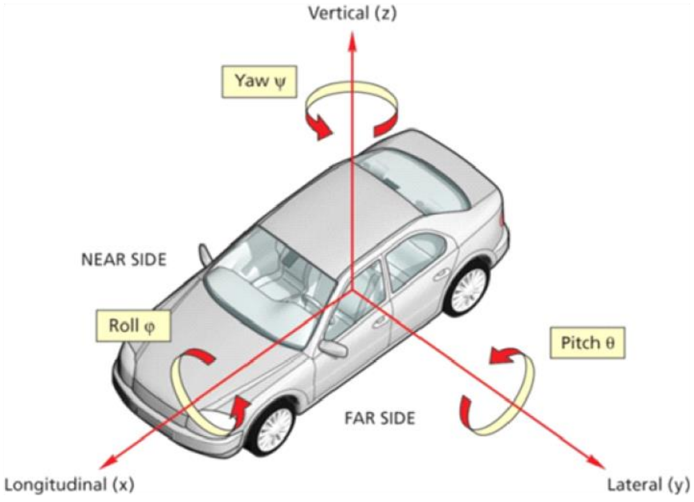
- **Maximum Speed (kph)** of the vehicle during the test
For that, we use the channel named “Speed (kph)” and we check the maximum during the test.

- **Minimum distance (m)** between the vehicle and the Target
This distance is 0 in case of Impact and in case of avoidance we use the channels named “Speed (kph)” and “Relative Longitudinal Distance (m)”.
First, we find the index where the vehicle stops, it means when “Speed (kph)” reaches 0 kph.
Then, we check the “Relative Longitudinal Distance (m)” value at the same index.

- **Vehicle Impact Speed (kph)** in case of impact
This is the Vehicle Speed at the time of collision with the Target. We use the channels named “Speed (kph)” and “Relative Longitudinal Distance (m)”.
First, we find the index of the collision, it means where “Relative Longitudinal Distance (m)” reaches 0 m.
Then, we check the “Speed (kph)” value at the same index.

- **Vehicle Speed (kph)** at driver avoidance in case of it.
This is the Vehicle Speed at the time of driver avoidance (steering or braking). Depending on the action, we can find the index of the avoidance (huge variation) using “Yaw Velocity (°/s)” or “Forward Acceleration (m/s²)”.
Then we check the “Speed (kph)” value at the same index.

4.4.4.2 Reference data system



4.4.4.3 Detail of tests performed and results table

Following extracts are not complete on purpose. For additional information, refer to deliverable 3.5 of WP3.

a) <u>Repeatability:</u>

First, three scenarios from ENCAP have been tested without any measurement equipment, to check which can be relevant or not:

Category	Scenarios	Number of subjective tests	Successful	Keep for objective tests

Repeatability	**CPNCO**	**2**	**NO**	**NO**
	CPFA	**3**	**NO**	**NO**
	CBLA	**3**	**YES**	**YES**

The two crossing scenarios (CPNCO and CPFA) are not relevant for this car, contrarily to the longitudinal one (CBLA) which has been selected for objective testing.
Then, 10 repetition of the same scenario CBLA have been performed and all the tests were successful (PASS).

Category	**Scenarios**	**Number of objective tests**	**Successful**
Repeatability	**CBLA**	**10**	**YES**

Here is the post-processing of the Repeatability part:

Scenario	Date	Time	Nbr of test	VUT Speed (kph)	Overlap	Success	VUT reaction	Anticipation	Comments	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
CBLA	16/02/2023	10:00	1	30	50%	YES	YES	YES	ACC REGULATE, OVERLAP CLOSE TO 25%	28,35	5,3	0	0
CBLA	16/02/2023	10:05	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	32,21	3,93	0	0
CBLA	16/02/2023	10:11	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	31,55	4,06	0	0
CBLA	16/02/2023	10:15	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	29,15	4,48	0	0
CBLA	16/02/2023	10:19	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	29,06	4,46	0	0
CBLA	16/02/2023	10:22	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,02	4,79	0	0
CBLA	16/02/2023	10:25	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,3	4,82	0	0
CBLA	16/02/2023	10:28	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	32,36	3,6	0	0
CBLA	16/02/2023	10:32	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	30,46	4,14	0	0
CBLA	16/02/2023	10:35	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,09	4,99	0	0
CBLA	16/02/2023	10:39	1	30	50%	YES	YES	YES	ACC REGULATE, NO AEB	28,23	5,06	0	0

b) Robustness:

In the same way, we started to perform the scenarios without equipment to see the relevancy:

Category	**Scénarios**	**Number of subjective tests**	**Successful**	**Keep for objective tests**
Robustness	**CBLA**	**1**	**YES**	**YES**
	Stationary CAR on emergency lane (new scenario)	**1**	**YES**	**YES**
	Stationary object on highway	**2**	**NO**	**NO**

The last scenario with an object on the road is not relevant with this car. The CBLA and the Stationary CAR are relevant; we selected those 2 for the next step.

Then, each scenario has been performed 10 times by changing different parameters (like Objects Speed, Angles, Overlaps…):

Category	**Scénarios**	**Number of objective tests**	**Successful**

Robustness	**CBLA**	**10**	**YES**
	Stationary CAR on emergency lane	**10**	**YES**

All the tests are successful, here are the details of the post-processing:

Scenario	Date	Time	Nbr of tests	ACC/AEB	VUT Speed (kph)	Overlap	Target Speed (kph)	camera	wipers with liquid	dummy accessories	Objec in the field of view	Success	Reaction	Anticipation	Comment	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
CBLA	16/02/2023		1	AEB	30	50%	15	clean	NO	standard	NO	NO	YES	NO	AEB LATE, DRIVER BRAKE	29,42	0	0	27,61
CBLA	16/02/2023	12:20	1	ACC	30	50%	15	dirty	NO	standard	NO	YES	YES	YES	acc regulate	30,79	4,02	0	0
CBLA	16/02/2023	11:50	1	ACC	30	50%	15	clean	NO	Yellow jacket	NO	YES	YES	YES	acc regulate	28,22	4,91	0	0
CBLA	16/02/2023	11:56	1	ACC	30	50%	15	clean	NO	jacket + backpack	NO	YES	YES	YES	acc regulate	30,01	4,25	0	0
CBLA	16/02/2023	12:05	1	ACC	30	50%	15	clean	YES	standard	NO	YES	YES	YES	acc regulate	29,15	4,46	0	0
CBLA	16/02/2023	11:35	1	ACC	30	75%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,11	4,96	0	0
CBLA	16/02/2023	11:19	1	ACC	30	25%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,16	4,91	0	0
CBLA	16/02/2023	11:40	1	ACC	30	0%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	27,89	5,24	0	0
CBLA	16/02/2023	11:45	1	ACC	30	100%	15	clean	NO	standard	NO	YES	YES	YES	acc regulate	28,23	1,53	0	0
CBLA	16/02/2023	12:16	1	ACC	30	50%	15	clean	NO	standard	parked car	YES	YES	YES	acc regulate	27,78	4,3	0	0

Robustness is not 100% perfect, but 90% with 1 impact (or test driver manuel avoidance) among 10 tests , as shown in red in the above table.

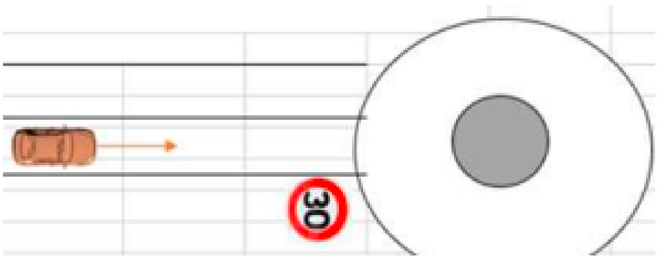
Scenario	Date	Time	Nbr of tests	ACC/AEB	VUT Speed (kph)	Overlap	Angles of driving (°)	objet	Target with roofbox	Success	Reaction	Anticipation	Comment	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
stationnary car on highway	16/02/2023	15:10	1	ACC	30	100%	0	NO	NO	YES	YES	YES		29,02	2,61	0	0
stationnary car on highway	16/02/2023	15:13	1	ACC	40	100%	0	NO	NO	YES	YES	YES		38,72	5,33	0	0
stationnary car on highway	16/02/2023	15:16	1	ACC	50	100%	0	NO	NO	YES	YES	YES		47,34	4,13	0	0
stationnary car on highway	16/02/2023	15:24	1	ACC	30	50%	0	NO	NO	NO	YES	NO	fcw very late and AEB when drive avoid (steering)	28,98	2,26	0	24,18
stationnary car on highway	16/02/2023	15:30	1	ACC	30	75%	0	NO	NO	YES	YES	YES		28,74	3,3	0	0
stationnary car on highway	16/02/2023	15:33	1	ACC	30	-75%	0	NO	NO	YES	YES	YES		29,01	3,55	0	0
stationnary car on highway	16/02/2023	15:40	1	ACC	30	100%	5	NO	NO	YES	YES	YES		29,08	3,55	0	0
stationnary car on highway	16/02/2023	16:12	1	ACC	30	100%	22	NO	NO	NO	NO	NO	no reaction (steering)	29,27	1,79	0	28,83
stationnary car on highway	16/02/2023	16:00	1	ACC	30	100%	0	YES	NO	YES	YES	YES		28,87	2,72	0	0
stationnary car on highway	16/02/2023	15:50	1	ACC	30	100%	0	NO	YES	YES	YES	YES		29,28	2,97	0	0

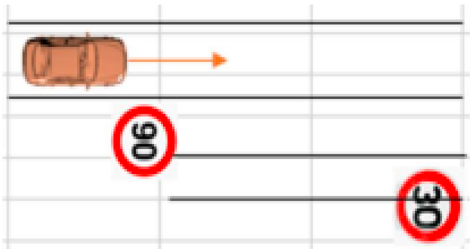
As for any vehicle, robustness is not 100% perfect, but 80% with 2 impacts (or test driver manuel avoidance) among 10 tests , as shown in red in the above table.

c) Pre-critical:

First, we tested the vehicle without equipment some situation which can generate anticipation of the system:

Category	**Scenarios**	**Number of subjective tests**	**Success-ful**	**Keep for objec-tive tests**
Pre-critical	**Improperly parked vehicle**	**1**	**YES**	**YES**
	Approach to roundabout	**1**	**NO**	**NO**
	Close and misleading traffic sign	**1**	**YES**	**YES**



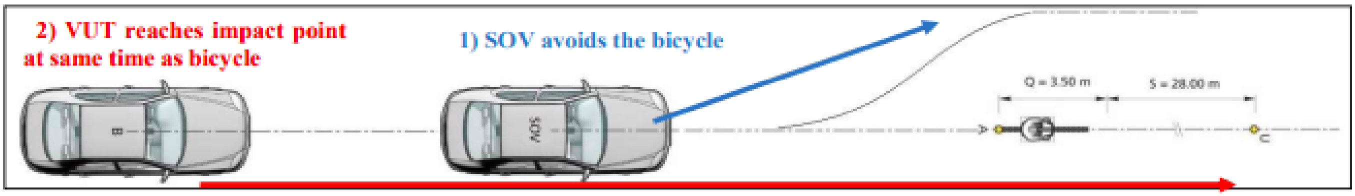


Then we performed some situation with measurement equipment.
All the tests were successfully performed, here are the details of the post-processing:

Scenario	Date	Time	Nbr of tests	VUT Speed (kph)	Success	Reaction	Anticipation	Comment	max speed before traffic sign (kph)	speed after 50kph traffic sign (kph)
Highway driving (close and misleading traffic sign 50kph EXIT)	08/03/2023	11:07	1	ACC regulation	NOK	ACC	ACC	Detection of the 50kph traffic sign (for EXIT) and speed adaptation (false positive)	85,26	48,97
Highway driving (close and misleading traffic sign 50kph EXIT)	08/03/2023	11:13	1	ACC regulation	NOK	ACC	ACC	Detection of the 50kph traffic sign (for EXIT) and speed adaptation (false positive)	85,74	48,46

d) <u>Random :</u>

We performed a new scenario that we imagined for this campaign; this is a CPLA merged with a Cut-Out:



For this scenario, the Golf 8 had a good reaction, the VUT first regulates its speed to keep a safe distance with the SOV, then after the SOV performed its Cut-out, the VUT regulates behind the bicycle.

This test is relevant, feasible and interesting to propose for future studies.

Scenario	Date	Nbr of test	VUT Speed (kph)	Overlap	Success	VUT reaction	Anticipation	Comments	max speed (kph)	remaining distance (m)	impact speed (kph)	avoidance speed (kph)
Longitudinal Bicyclist with VUT preceded by a vehicle	17-févr	1	70	50%	YES	YES	YES	ACC regulation on SOV then bike detection, then ACC regulation on bike	51,78	120,81	0	0

4.4.5 Conclusion

We tested 3 different vehicles, which are technical references (as largely explained in L3.2 in January) and with different levels of autonomy : level 1 for Golf8 and ZOE NEXYAD, level 4 for VALEO Drive4U.

Repeatability tests: as mentionned in L3.2 in January with some results, no vehicle is perfectly repeatable ; The 3 vehicles tested are not perfect in repeatabiliy and the main thing for safety is that they have no significantly lower performance than other vehicles with the same ADS functions but without AI .

Robustness tests : we built and confirm feasability of differents tests and influent parameters to change during the tests (like Objects Speed, Angles, Overlaps…) . The results of the tests show that the 3 vehicles tested are not perfect in robustness ; The main thing for safety is that they have no significantly lower performance than other vehicles without AI .

Anticipation tests: we built and confirm feasability of differents new tests to evaluate vehicle anticipation. 2 vehicles (Golf 8 and NEXYAD) showed real interesting anticipation skills so it would be good for safety to propose new tests and new evaluation of anticipation.

Random tests : we built and confirm feasability of differents new tests to evaluate vehicle anticipation. 2 vehicles (Golf 8 and VALEO) showed real interesting skills to manage some of these new tests & scenarios, so it would be good to avoid type approval overfitting to propose new tests for AI based vehicles type approval.

4.5 Real environment report

This chapter present an example of real environment tests.

The following tables are te templates used to describe the parameters used for the used cases.

TIME INFORMATIONS		
Channel names	*Units*	*Comments*
Time	s	Time starts in the path
MP Time	s	GPS time of VUT
MP Time Tracker 1	s	GPS time of VRU or GST

VUT SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Actual X (front axle)	m	X of the car (VUT) (at the bumper)
Actual Y (front axle)	m	Y of the car (VUT) (at the bumper)
Speed	kph	Absolute speed of the car (VUT)
Forward velocity	m/s	Forward speed of the car (VUT)
Lateral velocity	m/s	Lateral speed of the car (VUT)
Forward acceleration	m/s²	Forward acceleration of the car (VUT)
Lateral acceleration	m/s²	Lateral acceleration of the car (VUT)
Yaw angle	°	Yaw angle of the car (VUT)
Yaw velocity	°/s	Yaw velocity of the car (VUT)
Yaw acceleration	°/s²	Yaw acceleration of the car (VUT)

TARGET SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Head tracker reference X position	m	Position of the VRU on X axis
Head tracker reference Y position	m	Position of the VRU on Y axis
Head tracker forward velocity	m/s	Speed of the VRU on its path
Head tracker forward acceleration	m/s²	Acceleration of the VRU on its path

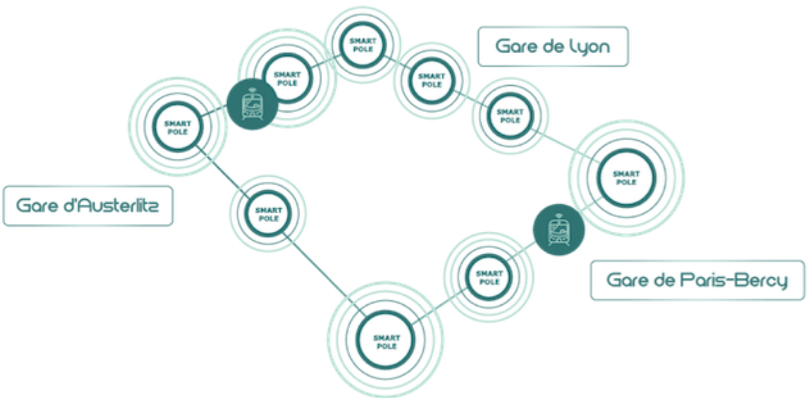
RELATIVES VUT/TARGET SPECIFIC INFORMATIONS		
Channel names	*Units*	*Comments*
Time to Collision (longitudinal)	s	Remaining time before the VUT strikes the target, assuming that the VUT and the target would continue to travel with the speed it is travelling

Relative longitudinal distance	m	Difference between the longitudinal positions of the vehicle and the target
Relative lateral distance	m	Difference between the lateral positions of the vehicle and the target
Relative longitudinal velocity	m/s	Difference between the longitudinal speeds of the vehicle and the target
Relative lateral velocity	m/s	Difference between the lateral speeds of the vehicle and the target
Relative yaw	°	Difference between the yaw angles of the vehicle and the target

4.5.1 Real environment report (ATC)

4.5.1.1 Presentation of the real environment pilot

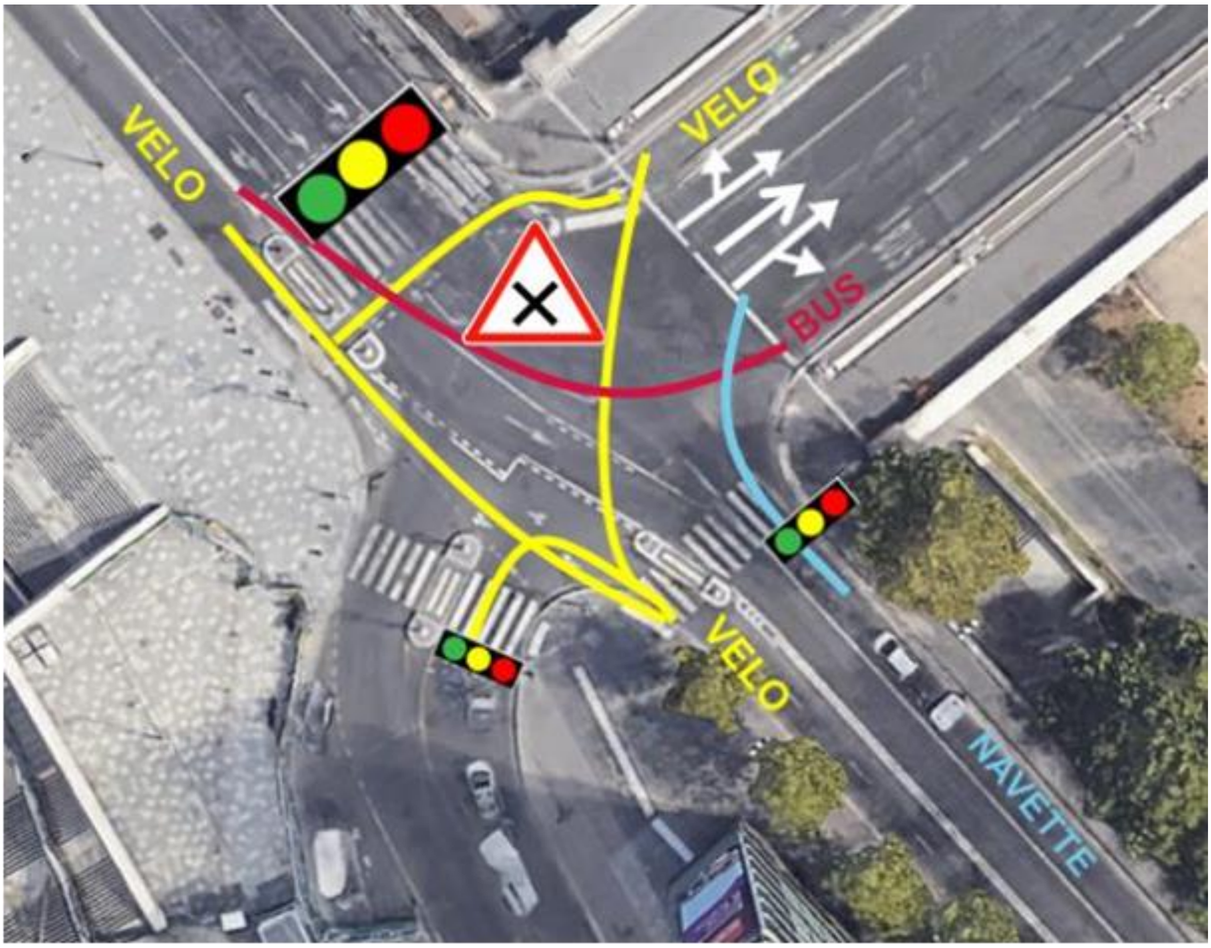
Real environment tests have been set in a very dense urban area in the core of Paris City. This area is focused on Paris 12th and 13th district and more precisely, between 3 main intermodal point such as Austerlitz, Lyon and Bercy station. It was named Paris2Connect (P2C) pathway.



This pathway is a rectangle of 3 km that has been "cut" in 50 sections (cf WP8) to define every descriptor for every scenario in PRISSMA real environment.
This pathway is mainly urban, and contains many sceneries such as straight lines, roundabout, complex crossroads, many different users (bicycles, scouters, bus, logistics, priority vehicles ...). This environment is complex too as the local legislation limits the speed of vehicles mainly between 30 and 50 km per hour, contains specific lanes for priority vehicles such as buses or taxis.

4.5.1.2 Scenarios and requirements for assessment

In order to test autonomous or automated vehicles into this pathway, adequation between technical possibilities of the vehicles, information and data collected, complexity of certain sections and development of test scenarios has been assessed (L4.3).
To this end, Austerlitz crossroads section was a good example of possibilities but also difficulties of this area.



A set of scenarios has been implemented with Valeo and RATP. Based on P2C pathway, Valeo scenarios are organized and classified into functional and logical following selection and identification as below:

Scenarios

Functional		Logical		Sections
1	Crossing Intersection	**1**	Crossing while no other road users	18
1	Crossing Intersection	**2**	Crossing intersection while 2-wheelers does not yield	18
1	Crossing Intersection	**3**	Cyclist crosses EGO path while right turn	25-28
1	Crossing Intersection	**4**	Pedestrian crossing while green light for EGO	17 / 26 / 42 / 49
2	Cut-in	**5**	Bus Cut-in in front of ego	22-23-24
2	Cut-in	**6**	Bicycle cut-in in front of the ego	29-36
2	Cut-in	**7**	Motorcycle cut-in from the right	9-14
3	Drive-off	**8**	Drive-off at intersection when surrounded by VRU	17 / 26 / 42 / 49
4	Ego lane change	**9**	Ego lane change	9
5	Lead vehicle braking	**10**	Front vehicle harsh braking (due to an occluded bicycle crossing the road)	22-23-24
6	Occluded actors	**11**	Occluded pedestrian crossing out of crosswalk	1-17
6	Occluded actors	**12**	Partially occluded pedestrians	TbD
6	Occluded actors	**13**	Occluded cyclist violating the right of way while right turn	25-28
7	Pedestrians expected	**14**	No occluded pedestrians crossing in front of the ego	1-17
7	Pedestrians expected	**15**	Group of pedestrians close to the lane edge (on the right)	41 - 42

RATP scenarios were defined by supervision analysis. Supervision of people, group of people or Vulnerable Road User like two wheels drivers in pedestrian traffic.

All observations led to create first results to show impacts of all criterions needed depending on results expected. 4 situations have been expected :

1- Real alert => "alert" with event in real situation
2- Real non-alert => "no alert" without event
3- False positive => "alert" without event
4- False negative => "no alert" with event

The goal is to launch experimentation and to evaluate the percentage of situations 1, 3 & 4

		Real situation	
		Event	**No event**
System	**Detection**	True ***[Alert]*** (% ?)	False « Positive » ***[Alert]*** (% ?)
	No detection	False « Negative » ***[No alert]*** (% ?)	True ***[No alert]***

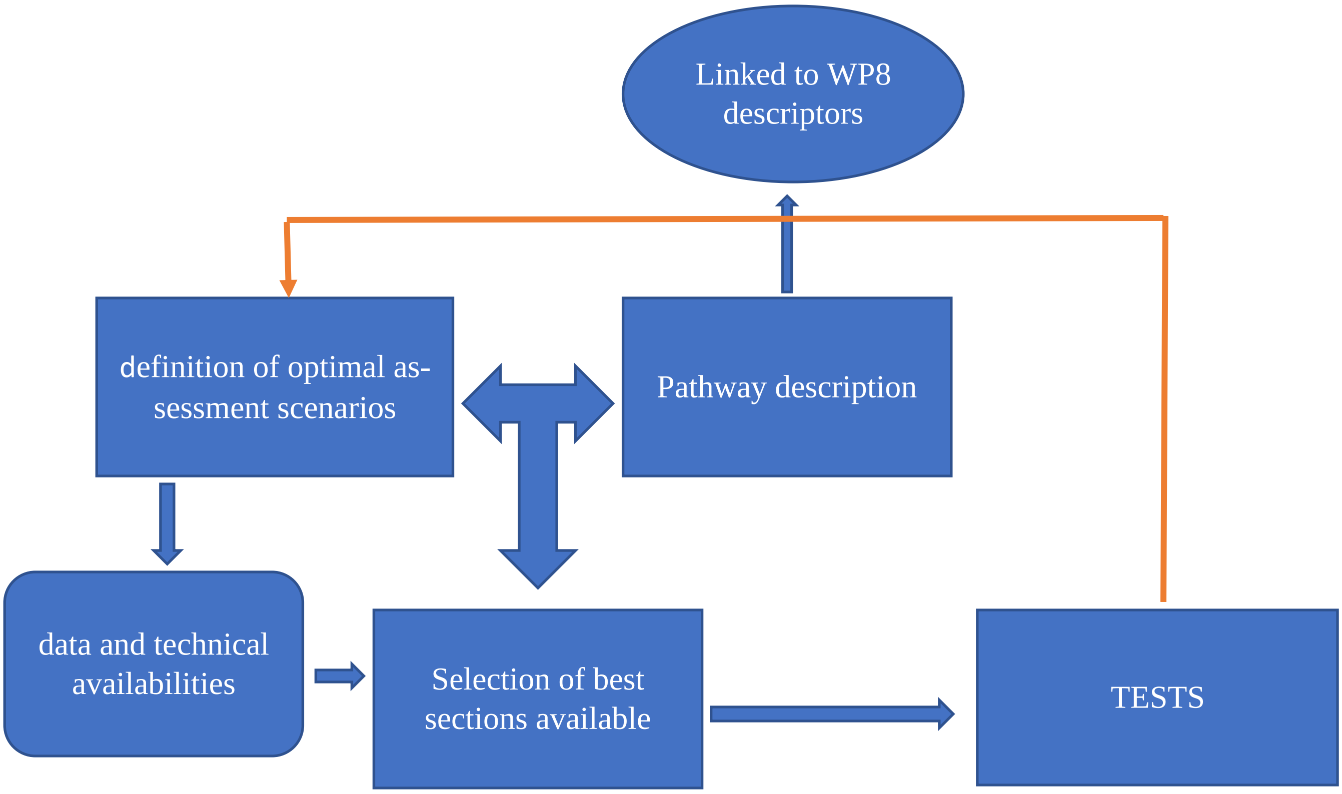
Based on P2C pathway, scenarios were organized and classified into functional, logical and concrete following selection and identification as below:

Scenarios

Functional		**Logical**	
1	Gathering of people "Bercy esplanade"	1	Gathering of people
1	Gathering of people "Bercy esplanade"	2	No gathering of people
2	Congestion at "Austerlitz"	1	Congestion
2	Congestion at "Austerlitz"	2	No congestion
3	Bike at "CdG / Van Gogh" bridge	1	Crossing

4.5.1.3 Main results and feedbacks

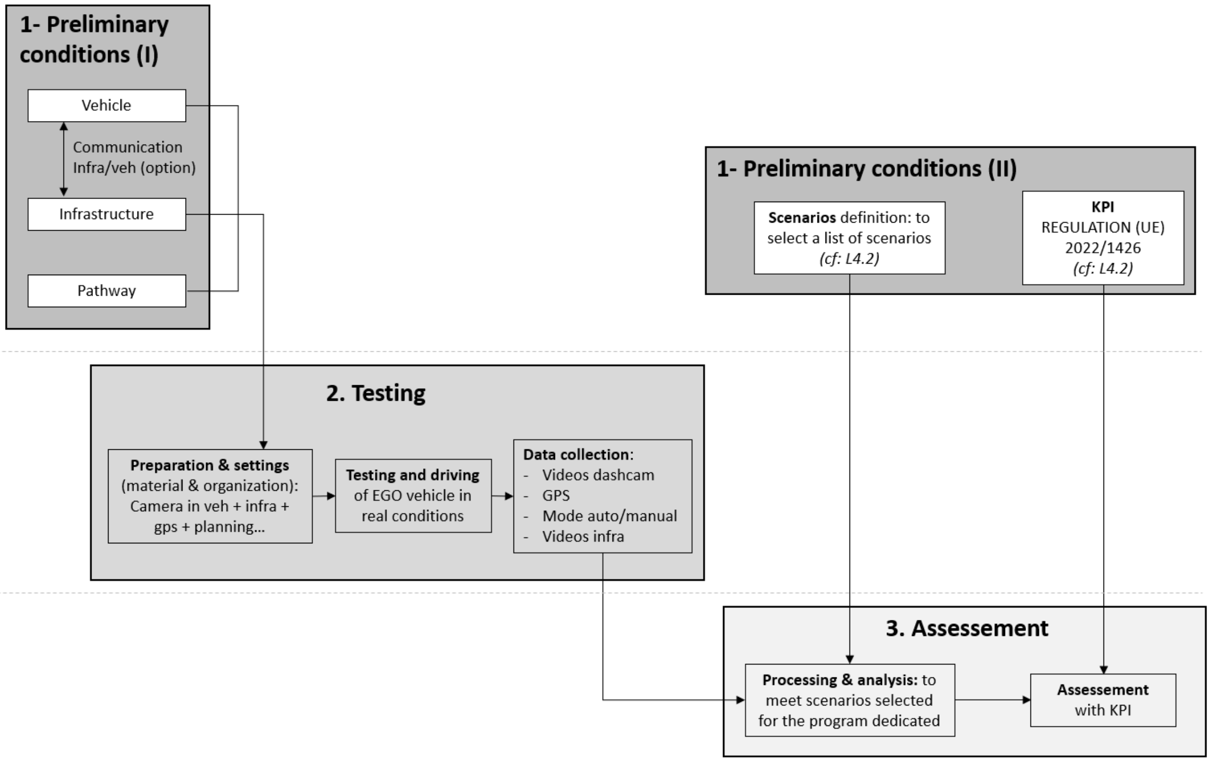
Main scheme uses in WP4 to assess vehicles / pathway validation:



As the PRISSMA project went on, optimal scenarios and PoCs were proposed by RATP and Valeo to test autonomous or automated vehicles and supervision.
Main difficulties were that the Paris2Connect pathway used its own infrastructure, with data already available, in a very constraint environment (dense urban area) with a lot of problematics. PoCs had to deal with emergency / priority vehicles, services (buses), logistics, accidents, and many diverse kinds of road users which added difficulties but also was rich enough to allow us to gather feedback.

So, one of our main targets has been to find optimal sections of the pathway to test scenarios. Then to ensure that technical and functional infrastructure, and datas were available to develop testing parts.

Finally, from this very constraint environment, PRISSMA managed to optimize the global assessment scheme.



The prerequisites were listed in the form of questions classified in the table below. These prerequisites are identified by importance: "Mandatory", "Recommended" or "Not important".

Questionnary preliminary conditions	Mandatory	Recommended	Not important
Vehicle			
Is there a safety-driver?		X	
Is it a level 4 vehicle?	X		
Is the timestamp (gps) calibrated with the infrastructure and other devices?			
Is veh. ODD adapted with pathway?	X		
What softwares are used in the system?	X		
What are data sources? Videos, CAM, GPS...			
- Videos	X		
- CAM		X	
- GPS: csv			X
Infrastructure			
Is the timestamp (gps) calibrated with the vehicle and other devices?			
What softwares are used?			
What are data sources? Videos, Spatem/Mapem...			
- Videos		X	
- Spatem/Mapem		X	

Pathway			
Is the pathway detailed through a taxon-omy?	X		
Scenarios			
Are scenarios are described in a support?	X		
KPI & metrics			
Are metrics from EU regulation 2022/1426?	X		
Other			
What are weather conditions?	X		

4.6 Supporting documentation for the platform (simulation tools)

When it comes to simulation-based assessment, the first thing to do is to define the tools required for the simulation platform. In the PRISSMA project, this key task has been addressed in deliverable 2.4 and 2.5. The choice of implementation of this platform is mainly based on the preliminary choices of requirements, ODDs, use cases and scenarios to be addressed. From there, it is possible to choose the various tools, models, simulation/graphical engines and simulation platforms needed to meet the evaluation and validation requirements and constraints. The final step is to select the evaluation and validation tools, both for the system under evaluation and for the sub-study platform. To illustrate this point, Figure 3 shows an example of the simulation environment set up for WP2.

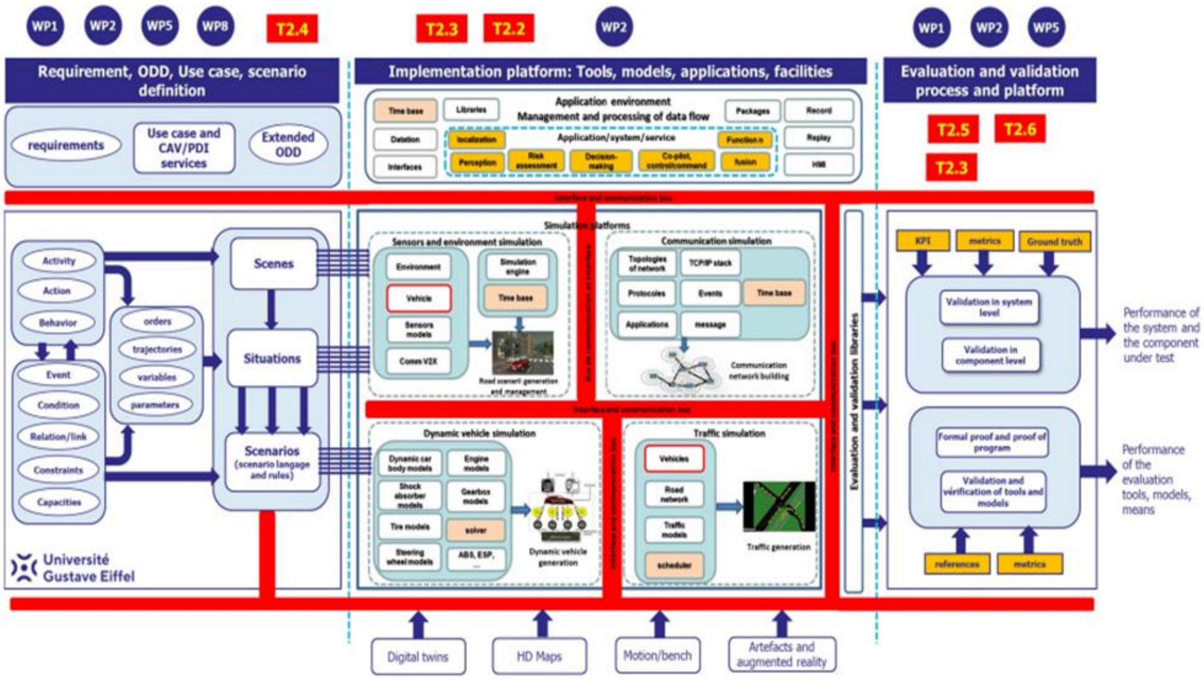


Figure 3: Global view of the simulation environment for evaluation process with its systems, functions, and components [3]

Deliverable L2.7 deals with pure simulation, Model-in-the-Loop (MiL)[1] and Software-in-the-Loop (SiL)[2] of the XiL method, which has the great advantage of enabling a very large number of scenarios to be run in a very short time, and variants to be examined in advance [3]. The XiL method enables the transparent integration of all relevant components and systems. As soon as the real components are integrated (Hardware-in-the-Loop (HiL)[3] or Vehicle-in-the-Loop (ViL)[4] on the test bench), the accuracy in relation to real experience increases. HiL and ViL are used in the analysis of critical scenarios where, for example, it is unclear whether the cause of undesired vehicle behavior is the actual function or a lack of model quality. Figure 4 presents an overview of PRISSMA scenario-based homologation methodology that used XiL method. Note that the greyed-out part is not considered in the PRISSMA project.

From [7]:
[1] MiL: software is tested as simulation model in a virtual environment
[2] SiL: software is tested as compiled target code in a virtual environment
[3] HiL: one or multiple ECUs (Electronic Control Units)
[4] ViL: a real vehicle is tested in partly virtual environment (Bench testing)

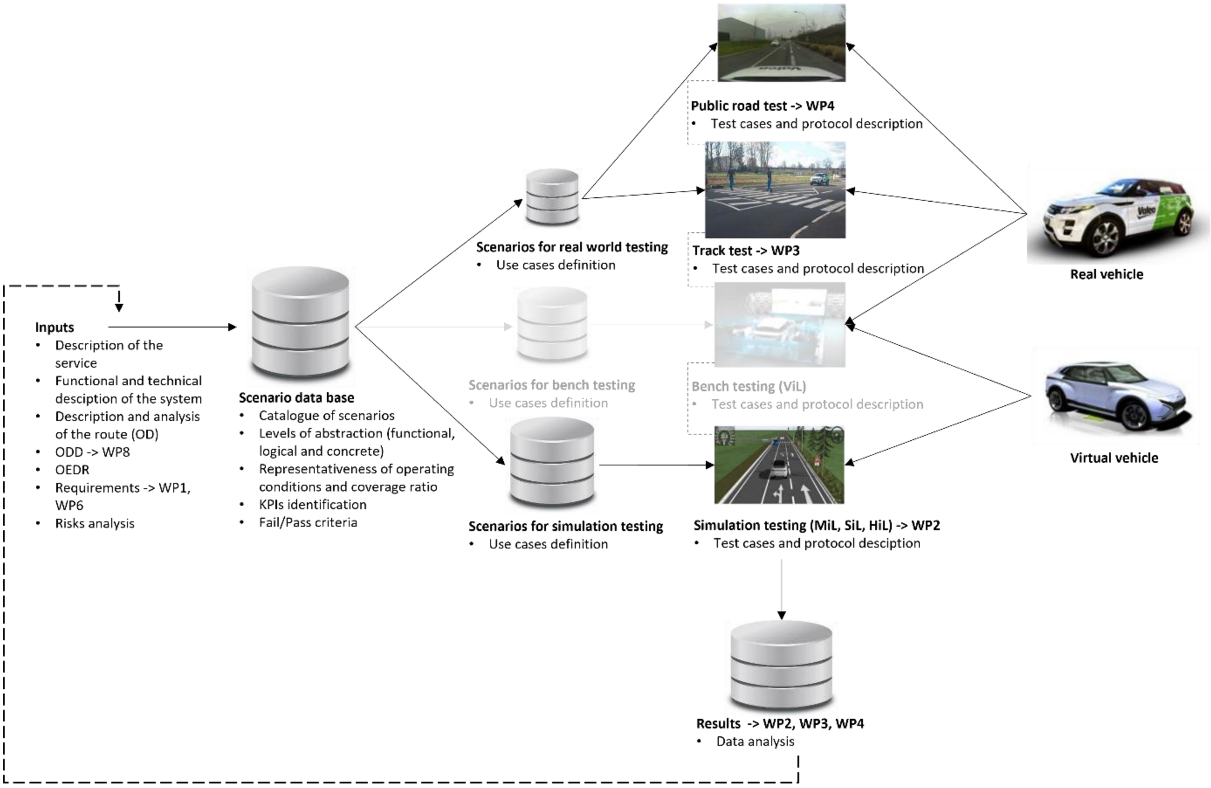


Figure 4: Overview of PRISSMA scenario-based homologation methodology

4.6.1.1 Models and components commonly used in vehicle simulations

Table 1 reviews the definitions of the required tools and components of simulation for the development and testing of automated vehicles.

Table 1: Models and components used in vehicle simulation [3]

Models used in vehicle simulation	Definitions
Vehicle dynamics model	Multi-body dynamics/Powertrain models
Environment Model	Terrain/Weather and environmental conditions models
Sensor Models	LiDAR/Radar/Camera/Navigation models
Control systems model	Electronic Control Units and control algorithms
Driver behavior model	Human driver's behavior model
Traffic model	Vehicles, pedestrians, and entities model
Simulation framework	Infrastructure to integrate and manage different models
User interface and visualization tools	Interfaces for users to interact with the simulator
Data analysis, validation and calibrations	Software tools used to analyze simulation results

4.6.1.2 Requirements for validating the usability of a simulation platform

A set of requirements is provided to validate the use of a simulation platform (i.e., simulation and graphical engines). These requirements are based on [4] and adapted to PRISSMA project objectives (cf. Table 2). The given requirements can be adapted to the different types of simulation (ViL, SiL, HiL, and MiL).

Table 2: Requirements for validating the usability of a simulation platform

Requirements	Definitions
R1. Multi-resource constraint	The platform must support multiple sources of input data to facilitate world creation and scenario generation
R2. Scenario management	The platform must support test automation across multiple created worlds and scenarios. This also involves the use of a specific scenario format such as OpenScenario and an event management mechanism to manage transitions between the scenes constituting the scenario. This also involves the use of a task and action scheduler
R3. Scripting Language	Scripting of the test automation process should be possible using standard scripting languages. This language must be usable at any time and allow the management, modification, addition, and deletion of any object, any parameter, and any action in real time
R4. Transparent code	The platform should use open-source code as much as possible for control and decision logic
R5. Modularity and adaptability	The platform must provide common core functionality with a configurable modular design. This means that the platform must be made up of easily loadable or unloadable plug-ins. This architecture must also offer an architecture allowing this processing to be distributed across several processors and several remote computers
R6. Simulation Fidelity and Quality	The platform will support physics-based worlds
R7. Sensor Modelling	The platform must support editable sensor models
R8. Sensor Types	The platform must support the most common sensors in the automotive domain. The platform must support at least RADAR, Lidar, GPS, IMU, Camera and Ultrasound sensors
R9. References and ground truths	The platform must provide ground truth data during simulation execution
R10. Ego-vehicle control	The platform must provide the ability to enable a control channel to control the simulated ego-vehicle
R11. Control of actors and extras	The platform must offer the possibility of controlling several actors. The platform must be able to populate the scene to increase loyalty
R12. Ensure process parallelization	The platform must support the ability to run and evaluate multiple control channels simultaneously
R13. Signal scheduler	The platform must offer the possibility of prioritizing control signals between the different channels
R14. Control Signals	Each control channel must provide control signals conforming to the same specification
R15. Data flow management and scheduler	The platform must be able to distinguish control signals coming from different channels
R16. Sensors and sources separability	The platform must be able to distinguish between several sensors of the same type
R17. Unexpected events and rare scenarios	The platform will support the creation of handcrafted worlds and scenarios, as well as their subsequent adaptation to take into account rare and unexpected scenes and scenarios

R18. Usability of Datasets	The scenario database must be reusable in the sense that it must be independent of the perception and control logic used
R19. Data channel adaptability	The connection of the sensors to the control channel must be configurable to adapt to the needs of each channel
R20. Cyber security and operating safety	The platform must have the ability to inject faults during execution. Moreover, the platform must provide inputs and mechanism allowing to simulate cyber-attacks (perception, communication, component)
R21. Generic scripting language	The platform must offer the ability to script fault injection in the same interface as test automation
R22. Generic architecture respecting standard	The platform must support co-simulation standards such as FMI for large and specialized simulations
R23. Reproductibility	The portability of code generated from the platform should not be limited to a particular hardware configuration
R24. Repeatability	The platform must provide the same result after n times the same scenario with the same platform configuration
R25. Real-time constraint	The platform must guarantee a real-time processing of the involved model, plug-ins, module

To obtain a simulation platform usable with a high level of fidelity and credibility for simulation-based testing for automated driving, the PRISSMA project proposes to supplement the previous list of requirements by focusing on the main functionalities and capabilities that a simulation platform can have. These capabilities are listed in Table 3.

Table 3: Functionalities/Capabilities that a simulation platform can have [3]

Functionalities/Capabilities	**Definitions**
Multi-spectral rendering and modelling of the propagation channel	Capability to mimic signals provided on several bandwidths. This capability is essential for sensors modelling. For instance, the simulation of intrinsic parameters and operating of the RADAR needs to generate and to process high frequency signals (24 Ghz or 79 Ghz). Currently, this type of high frequency management is obtained by using specific libraries and GPU capacities (use of CUDA language)
Lights generation and management	Capability to manage a large set of light sources with an accurate and efficient pixel level rendering (for real time processing). Generation of accurate and dynamic light masks.
Shadows generation and management	Capability to manage properly the different light sources and their interaction with objects and the environment. This means to propose mechanism providing several shadows renderings (ambient occlusion map, occlude shadow, cast and catch shadows, self-shading, ...)
Material and meta material	Capability to provide large range and efficient resource management (graphics (material, texture, ...), Cross Radar Section, Bidi[1]rectional Reflectance Distribution Function (BRDF), IR material emission, ...
Textures management and generation	Capability to provide shaders and functions allowing to manage and generate HDR texture (coding light intensity, see figure 19), procedural and animated textures, Multiple reflexional mechanism

	(environment reflection on car body, windows, wet road, ... with a resolution fitting with requirement of sensors
Ray tracing mechanism	Capability to provide an efficient and real time ray tracing mechanism
Filter mechanism	Capability to provide library of shaders usable by sensors and implemented specific physical models allows to apply specific modification and transformation to a raw data generated by sensors. This mechanism is useful and essential for the camera and for generating weather conditions. These filters allow, for example, planar, cubic and cylindrical reflections, noise, blur, fog, depth of field, optical distortion, color, etc.
Spatial management	Capability to implement quaternion library to avoid errors generation in the positioning and the orientation of the objects.
Physical engine	Capability to provide a library allowing to apply dynamic model for dynamic object with the management of physical interactions between objects. For instance, truck modelling needs to implement dynamic modelling of the cabin and the trailer with the physical link between both.
Particle filter	Capability to provide an efficient mechanism for adverse conditions simulation. Particle engines generate rain, snow, fog, cloud, smoke, fire effects with a high level of fidelity.
Multiple layers management	Provide the capability to manage in same time several parallel processing for specific resources and models (i.e., simulation of camera, GPS, RADAR, and IR in same time with their own physical resources and requirements)
Time management	Capability to provide an accurate mechanism of time management for orchestration/synchronization of the various simulators and models. This function needs to generate real-time operating with a high level of repeatability (several same scenarios and simulations provide the same result with the same time stamping of the data). The time generator and manager need to provide a large set of time modelling (see figure Time). It is essential to control the operating period and frequency of each sensor
Event generation and management	Capability to implement event mechanisms and functions with specific conditions, relations, constraint, situations (spatial, temporal, semantic, climate, ...). Moreover, event variable is essential to provide an automated validation process with the coverage of a large set of values for significant parameters and variables under test or generated relevant situations under test
Interfaces	Capability to interconnect different tools and models with one another has become a crucial need. It is with this intention that a general standard with the acronym FMI (Functional Mock-up Interface) was created, for easing up the exchange of models and standardizing the way of connecting and sequencing them. This interface needs to support the transfer of large amounts of information (video streaming, for example).
Plug-in mechanism	Capability to load plug-in instances during real-time operation

Digital Twin of test benches	Capability to generate not only Digital Twin of real environment but also the virtual test benches using on these road environment (open road or controlled environment)

4.6.1.3 Description and modelling of a simulation platform

With reference to deliverable 2.4, the hardware and software components of the simulation platform and the interaction between them must be described [5]. To facilitate exchanges between the various stakeholders, the PRISSMA project proposes to use an (interoperability) standard based on the ARCADIA (ARchitecture Analysis and Design Integrated Approach) methodology. The ARCADIA methodology comprises 4 stages: **Definition of the Problem - Customer Operational Need Analysis** (define operational capabilities, perform an operational need analysis), **System/SW/HW Need Analysis** (perform a capability trade-off analysis, perform a functional and non-functional analysis, Formalize and consolidate requirements), **Logical architecture Design** (define architecture drivers and viewpoints, Build candidate architectural breakdowns in components, select best compromise architecture), **physical architecture design** (define architectural patterns, consider reuse of existing assets design a physical, design a physical reference architecture, validate and check it) (cf. Figure 5).

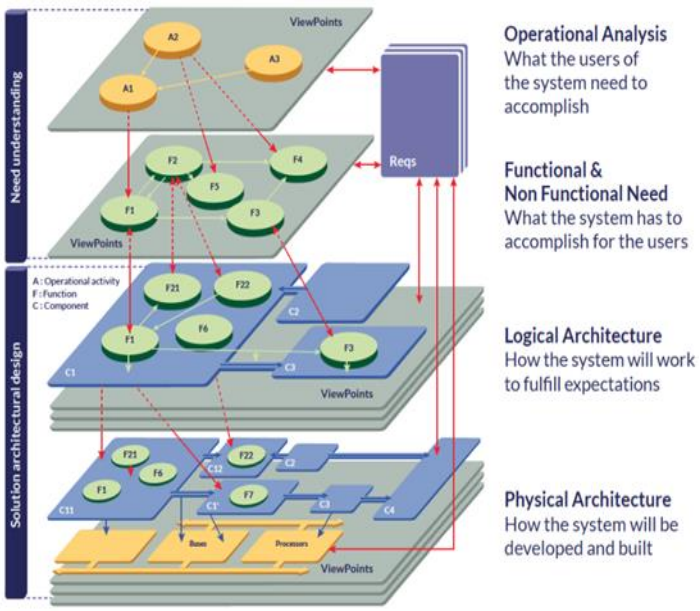


Figure 5: ARCADIA methodology [6]

Arcadia is a model-based engineering method for systems, hardware and software architectural design. It has been developed by Thales between 2005 and 2010 through an iterative process involving operational architects from all the Thales business domains. Since 2018, Arcadia is registered as Z67-140 standard by AFNOR, the French national organization for standardization.

Arcadia promotes a viewpoint-driven approach (as described in ISO/IEC 42010) and emphasizes a clear distinction between need and solution. [Capella MBSE Tool - Arcadia (mbse-capella.org)] [6].

5 LIFE CYCLE MANAGEMENT FOR SYSTEMS INTEGRATING AI BASED SOFTWARE

Performing Life Cycle Management of a system integration IA based modules is a real challenge, considering specific properties of IA technologies.

This is obviously the case for automatic shuttles, and most important feature of Life Cycle Management requirements has to cover relevant feedback and corrective action when an unacceptable operational situation has been experienced in the operational cycle of the system: this happens when an accident occurs or at least a near miss.

5.1 General process

Industrial supplier of autonomous or automated shuttles has to describe its maintenance concept he has defined for the whole Life Cycle of the system:

- How accident or near miss scenarios and use cases are recorded and detected during the operation of the system
- How they are qualified
- How they are diagnosed
- How corrective actions have to be identified and validated
- How non regression can be proved.

Concerning the recording of accident and near miss the §3.1.2 of the deliverable 6.5 from the PRISSMA project details all the specifications for the use of an Event Data Recorder.

Near miss qualification process

What qualification process is advised by the manufacturer to identify and isolate an accident or near miss situation which has obviously to be diagnosed, corrected and validated?

What criteria most characterize a near miss from the point of view of the supplier?

Cf §2.1.1 of the deliverable 6.5 from the PRISSMA project specifying the ODD and 8.11 giving the taxonomy.

Diagnosis process

What diagnosis process does recommend the supplier, if an accident or a near miss is established?

Following sequence describes example of such process:

1. When an accident or a near miss is identified, one has first to identify the single cause or multiple causes of this unacceptable behavior, and then to setup proper corrections.

2. Different kinds of corrections have to be envisaged, depending on the nature of the causes diagnosed:
   - If one cause is a failure mode of a hardware component or module, a proper corrective maintenance task can be enforced, in accordance and compliancy with the maintenance policy of the system: this failure mode refers to an identified Line Replaceable Unit which can be exchanged on site, or on another maintenance level, regarding the maintenance concept
   - If one of the possible causes is a non-AI software error, a cause analysis has to be applied to the software: it can be a specification error, or a coding error, and in

both cases update of the software may be in question, as well as to find out why in the development process this error has been let unknown

- If one of the possible causes is an AI based software error, a cause analysis has to be applied to the software; after this cause analysis, correction(s) of the software must be proposed, and impact analysis of this (these) correction(s) have to be applied; besides a diagnosis has to be applied to the development process and framework which has let this error unknown.

More details are available in the deliverable 7.3 of the PRISSMA project.

5.2 Diagnosis of the AI based software

The supplier should describe diagnosis process he recommends for an AI based software, once its contribution(s) to an accident or a near miss has been proved.

These contributions should be qualified for example in the real world, as trustworthiness of models supporting simulations remains currently partial: replicability and repeatability of the unacceptable situation to which AI component has contributed would be decisive about the fact to qualify the irrelevant behavior and internal diagnosis.

5.3 Correction of the AI based software

To find proper correction of AI based software able to reestablish convenient and acceptable behavior of the whole system in the use case addressed originally, one has to conduct a deep survey to identify part of the software to correct and precise elements to change, update or remove.

Therefore, the supplier has to describe what corrective proofs he recommends for the correction of an AI based software having contributed to a near miss or an accident, and which has to be corrected.

For example, if AI software is based on Neural Networks, one has to find out what layer (s) of the networks to modify, and what value of weights to modify and readjust to obtain correction of the global behavior of the top-level system in addressed use case.

Contrary to non-AI diagnosis tools, there is not a large panel of relevant methodologies and tools to diagnose AI bricks and systems.

The learning models of the AI bricks of the autonomous or automated driving system require diagnosis when failure cases are encountered during the operation of the autonomous or automated vehicle. These learning models have to follow an elaborate testing and certification process to avoid accidents. This process is time consuming and can take up to 6 months to 1 year for each update. However, we expect that customers will always encounter failures that are underrepresented in the training data and not taken into account in the test data or due to missing features in the learning model.

Thus, an important issue facing autonomous or automated vehicle operators is the maintenance of the autonomous driving system software of AI bricks between major software updates, in order to fix the driving behavior of the autonomous or automated module on the encountered failure cases or to add the requested missing functionalities of the model without the need to validate the whole system from the beginning. We believe that the diagnosis and maintainability of learning models are important challenges for the success of autonomous or automated shuttles. The maintainability of autonomous or automated driving systems must correct the failures of the learning models without changing the driving behavior over all the kilometers that have been successfully driven before.

That’s why the supplier has to describe the most efficient and comprehensive process to achieve proper correction of AI based software / module having caused or contributed to an accident or a near miss.

Remark: this process may be re captured from the original OEM of the AI based software.
Cf 4.8 of the deliverable 6.5 specifying test on the AI components.

5.4 Non-Regression demonstration

The supplier shall describe what process he recommends for Impact Analysis and Non-Regression demonstration when a corrective process has been applied.

Corrective action on a faulty software has to remove a faulty behavior, but at the same time, one has to be sure that it does not produce additional mis behavior on other use cases, which were not failing before. This a tricky issue which is not yet wholly covered by the state of the art but in which alternative solutions are proposed.

Cf 4.8 of the deliverable 6.5 specifying test on the AI components.

5.5 Diagnosis of System Engineering Framework

The supplier should describe how he takes into account potential failures of his system engineering framework on the fact that it could have generated near misses or accidents on the automatic or autonomous shuttle.

Many components or layers of System Engineering Framework may be also addressed to explain why use case under analysis where system has failed, has not been sufficiently taken into account, and has not been therefore anticipated in the design and development process, which implies that this process has to updated in a way, as is illustrated by following questions:

- Have requirements been sufficiently formulated? May be failed use case was not covered by these requirements…
- Has the ODD been correctly determined? May be failed use case went beyond the ODD…
- Have OEDR been correctly formulated? May be failed use case were not integrated in the OEDR…
- Which library of scenarios and use cases have been simulated in the virtual testing campaign? May be failed use case has been forgotten in this library…

- Which families of scenarios and use cases have been simulated in the controlled testing campaign? May be failed use case has been forgotten in these families…
- Which families of scenarios and use cases have been simulated in the real environment testing campaign? May be failed use case has been forgotten in these families…
- Which metrics have been applied in post processing of virtual, controlled or real testing campaigns? May be failed use case had been assessed with a not convenient metric
- Which criteria had been applied for the characterization of the near miss accidents? May be these criteria were too optimistic…
- Which platform was used for the simulation of the failed use case? May be this platform had led to optimistic results

Cf § of the deliverable 6.5 of the PRISSMA project specifying the audit to conduct on system engineering.

REFERENCES

[1] PRISSMA Project, «Deliverable L1.4 - Tests and Audit Requirements - Initial Report,» 2022.

[2] PRISSMA Project, «Deliverable 2.7 - Methodology, procedures and protocols for evaluation of application and virtual test facilities,» 2023.

[3] D. Gruyer, R. Régnier and al., "Methodology, Procedures, and Protocols for Evaluation of Applications and Virtual Test Facilities," 2024.

[4] N. Mohan and M. Torngren, "Ad-eye: A co-simulation Platform for Early Verification of Functional Safety Concepts," *SAE Technical Paper 2019-01-0126,* 2019.

[5] C. Gava, D. Gruyer and al., "Preliminary Definition of Interfaces and Simulation Environment," 2022.

[6] Capella , "Let Yourself Be Guided with ARCADIA," Capella, [Online]. Available: https://mbse-capella.org/arcadia.html. [Accessed 2024].

[7] K. V. Neumann-Cosel, *Virtual Test Drive,* Vols. Virtual Test Drive. PhD. Munchen, ¨, Munchen, Technische Universitat Munchen, 2014.