



D5.1 Testing and evaluation methodology for AI-driven CCAM systems

Dissemination level	Public (PU)
Work package	WP5
Task:	T5.1
Deliverable lead:	VIF
Version	V1.0
Submission date	31/01/2024
Due date	31/01/2024

Authors

Authors in alphabetical order		
Name	Organisation	Email
Arnautovic, Edin	TTTA	edin.arnautovic@tttech.com
Bassoumi, Islem	CAF	islem.bassoumi@conti-engineering.com
Beemelmanns, Till	ika	till.beemelmanns@ika.rwth-aachen.de
Braat, Michiel	TNO	michiel.braat@tno.nl
den Ouden, Jos	TUE	j.h.v.d.ouden@tue.nl
Elrofai, Hala	TUE	h.b.h.elrofai@tue.nl
Buermann, Maren	TNO	maren.buermann@tno.nl
Hillbrand, Bernhard	VIF	bernhard.hillbrand@v2c2.at
Konstantinos, Gkentsidis	SIE-BE	konstantinos.gkentsidis@siemens.com
Küppers, Guido	ika	guido.kueppers@ika.rwth-aachen.de
Naranjo de las Heras, Ruben	VICOM	rnaranjo@vicomtech.org
Nieto, Marcos	VICOM	mnieto@vicomtech.org
Okanovic, Ilma	VIF	ilma.okanovic@v2c2.at
Ruh, Jan	TTTA	jan.ruh@tttech.com
Ryabokon, Anna	TTTA	anna.ryabokon@tttech.com
Säman, Timo	Valeo	timo.saemann@valeo.com
Sarrazin, Mathieu	SIE-BE	mathieu.sarrazin@siemens.com
Stettinger, Georg	IFAG	georg.stettinger@infineon.com
Wijbenga, Anton	MAPtm	anton.wijbenga@maptm.nl

Control sheet

Version history			
Version	Date	Modified by	Summary of changes
0.1	20.11.2023	Bernhard Hillbrand (VIF)	Initial ToC
0.2	15.12.2023	All authors	First contributions to the technical chapters
0.3	22.12.2023	All authors	Additional contributions to all chapters
0.4	10.01.2024	All authors	Additional contributions to all chapters
0.5	14.01.2024	All authors	Adding final contributions
0.6	14.01.2024	Bernhard Hillbrand (VIF)	Final formatting before the internal review
0.7	15.01.2024	Holger Loewendorf (IRU)	Proofreading
0.8	23.01.2024	All authors	Changes after review
1.0	31.01.2024	Esther Novo (VICOM)	Final version ready for submission

Peer review		
	Reviewer name	Date
Reviewer 1	Nerea Aranjuelo (VICOM)	18/01/2024
Reviewer 2	Justyna Beckmann (FIA)	17/01/2024



Funded by
the European Union

Project funded by



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Education,
Research and Innovation SERI

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI).

TABLE OF CONTENTS

Authors	2
Control sheet.....	2
TABLE OF CONTENTS	4
List of Figures.....	5
List of Tables	6
1. Introduction.....	7
1.1. Althena concept and approach	7
1.2. Existing evaluation methodologies	7
1.3. Purpose of this deliverable	8
1.4. Structure of this deliverable	8
2. Use case overview.....	9
2.1. UC-1: Reliable Pedestrian Detection in Urban Environments	10
Approach and Overview	10
Model-Driven Strategies.....	11
Data-Driven Approaches	11
Sensor-Driven Fusion Techniques.....	11
UC-1 Story & Scenario	12
2.2. UC-2.1: Collision prediction with Hybrid AI data fusion models.....	13
2.3. UC-2.2: Robust Prediction modules for Robo-taxi in urban environment	15
2.4. UC-3: Trustworthy and Human understandable decision-making	19
2.5. UC-4: AI Models and Traffic Management	23
3. Validation approach for each use case	25
3.1. High-level requirements.....	25
3.2. Benchmark scenarios	31
3.3. Key performance indicators (KPIs)	33
4. AI lifecycle in reference to the use case activities.....	36
Conclusion.....	39

LIST OF FIGURES

Figure 1: Overview of the interaction between the 4 Althena use cases.....	9
Figure 2: Possible demonstrator vehicle with mounted sensor rack consisting of multiple LIDAR and camera sensors.....	10
Figure 3: UC-1 1 st Althena Cycle - "Offline" Demonstrator - Components	11
Figure 4: Schematic illustration of the UC-1 scenario. An urban scenario is considered where an obstacle blocks the view on a pedestrian.....	13
Figure 5: This diagram shows the technical components behind the solution.....	15
Figure 6: Block diagram: Robust prediction modules for Robo-taxi environments.....	16
Figure 7: Information flow of prediction based on environmental awareness.	16
Figure 8: Abstracted functional architecture of software modules demonstrated in UC ₃	20
Figure 9: Schematic illustration of SCEN_09 demonstrating the explanation of non-intuitive braking due to an occluded vehicle.	22
Figure 10: Schematic illustration of SCEN_10 demonstrating the robustness of the Hybrid-AI planning stack against unknown contexts.	22
Figure 11: Use case/demonstrator links to the AI lifecycle.....	36

LIST OF TABLES

Table 1: List of requirements of the 4 Althena use cases.....	25
Table 2: List of scenarios of the 4 Althena use cases.....	31
Table 3: List of key performance indicators of the 4 Althena use cases.....	33

1. Introduction

1.1. Athena concept and approach

Connected, Cooperative and Automated Mobility (CCAM) solutions have emerged thanks to novel Artificial Intelligence (AI) which can be trained with huge amounts of data to produce driving functions with better-than-human performance under certain conditions. The race on AI continues building hardware (HW) and software (SW) frameworks to manage and process even larger real and synthetic datasets to train increasingly accurate AI models. However, AI remains largely unexplored with respect to explainability (interpretability of model functioning), privacy preservation (exposure of sensitive data), ethics (bias and wanted/unwanted behaviour), and accountability (responsibilities of AI outputs). These features will establish the basis of trustworthy AI, as a novel paradigm to fully understand and trust AI in operation, while using it at its full capabilities for the benefit of society. Athena will contribute to build Explainable AI (XAI) in CCAM development and testing frameworks, researching three main AI pillars: data (real/synthetic data management), models (data fusion, hybrid AI approaches), and testing (physical/virtual X- in the loop (XiL) set-ups with scalable Machine Learning Operations (MLOps)). A human-centric methodology will be created to derive trustworthy AI dimensions from user-identified group needs in CCAM applications. Athena will innovate proposing a set of Key Performance Indicators (KPI) on XAI and an analysis to explore trade-offs between these dimensions. Demonstrators will show the Athena methodology in four critical use cases: perception (what does the AI perceive and why), situational awareness (what is the AI understanding about the current driving environment, including the driver state), decision (why a certain decision is taken), and traffic management (how transport-level applications interoperate with AI-enabled systems operating at vehicle level). Created data and tools will be made available via European data sharing initiatives (OpenData and OpenTools) to foster research on trustworthy AI for CCAM.

1.2. Existing evaluation methodologies

A Common Evaluation Methodology (CEM) for CCAM is currently being developed by the FAME (Framework for coordination of Automated Mobility in Europe) project¹. The goal of the CEM is to provide guidance on how to set up and carry out an evaluation or assessment of direct and indirect (including wider socio-economic) impacts targeting different user groups. The Common Evaluation Methodology will become part of the European framework for testing of CCAM on public roads. At the time of writing this deliverable, the EU-CEM Handbook is not yet available. The FAME project will deliver the first draft of CEM for CCAM in May 2024, and the final version of the Handbook will be available in June 2024 on the Knowledge Base² on Connected and Automated Driving.

The FESTA methodology for Field Operational Tests (FOT) was developed in 2008 as part of the FESTA project. The methodology is described in the FESTA Handbook³, which has been updated since then (by FOT-Net, CARTRE and ARCADE EU-funded projects). FOT are described as: “A study undertaken to evaluate a function, or functions, under normal operating conditions in road traffic environments typically encountered by the participants using study design so as to identify real-

¹ FAME is Research & Innovation Action funded under the Horizon Europe programme:
<https://www.connectedautomateddriving.eu/about/fame/>.

² <https://www.connectedautomateddriving.eu/about/>

³ <https://www.connectedautomateddriving.eu/methodology/festa/>

world effects and benefits”. The latest version of the FESTA Handbook dates from September 2021.

The Micro-FESTA was developed in August 2021 (by the ARCADE project) to support the evaluation methodology of small pilot projects of CCAM. The document provides an overview of the main steps in the FESTA methodology and comments on their role in testing on a smaller scale.

1.3. Purpose of this deliverable

This deliverable is the initial report of work package 5 (“Deploy & Test”) and is related to task 5.1 (“Testing and validation methodology”). The entire WP and therefore also this task is divided into two parts at the end of each cycle (mid-term and end of the project).

D5.1 lays the foundation for the testing and validation work in the following tasks, which are specifically about the use cases.

- UC-1: Trustworthy Perception Systems for CCAM
- UC-2: AI extended Situational Awareness/Understanding
- UC-3: Trustworthy and Human understandable decision-making
- UC-4: AI-based Traffic Management

These tasks will start at the end of T5.1 and will use the content of D5.1 to apply it to their use cases. This will lead towards the fulfilment of the objective to demonstrate the capabilities of AI-driven CCAM systems in established ODDs (Operational Design Domains).

Therefore, a strong focus will be on the requirements of the individual demonstrators. Also, benchmark scenarios and KPIs are defined for the needs of each demonstrator.

It should be mentioned here that the time budget is not distributed equally between the use cases. Use Case 2 accounts for around half of the person months, which also explains the different numbers of contributions in this deliverable.

1.4. Structure of this deliverable

This deliverable is structured as follows:

- Chapter 2 introduces the use cases and demonstrators of Althena. This introduction is not intended to be complete but focuses on the relevant topics of this deliverable. A complete description will be provided in Althena deliverable 1.2.
- Chapter 3 focuses on the validation approaches of the different use cases and describes the requirements, benchmark scenarios and KPIs defined for each demonstrator.
- Chapter 4 shows the AI lifecycle in reference to the use case activities. Each use case focuses on certain parts of the lifecycle. The activities towards those parts will be described. This chapter addresses the WP5 objective “Identification of similarities and differences of the developed AI-approaches with respect to their specific AI-lifecycle”.

2. Use case overview

The software stack of an automated vehicle can be divided into the three abstract tasks of perception, understanding, and decision-making. The task of perception is to process raw sensor data and to perform an initial modelling of the vehicle environment. This model is enriched with additional information like map data or information received through external communication through the modules associated with the task of understanding. The finalised environment model serves as the information basis for decision-making. Based on this information, the respective modules generate reasonable decisions that lead to vehicle trajectories which can be followed by applying the vehicle controls.

Within Althena the entire processing pipeline of an automated driving software stack is investigated in three specific use cases accounting for Perception (UC1), Understanding (UC2) and Decision-Making (UC3) respectively. In addition, Use Case 4 (UC4) will apply the Althena methodology from a Traffic Management perspective.

While UC1, UC2 and UC3 are on the vehicle level and each use case, or scenario within a use case, determines how a single vehicle (eventually) behaves in traffic, UC4 focuses on the impact several of such vehicles have on the traffic dynamics on a network level. Since there are multiple scenarios and not all use cases are necessarily applied to the same vehicle, there can be multiple types of AVs or, to refer specifically to AVs using AI from the Althena project, Althena vehicles (see Figure 1).

Use Case 4 will evaluate the impact different compositions and penetration grades of such Althena vehicles will have on the network traffic dynamics. Obviously, before Use Case 4 can perform such an evaluation, the other use cases will have to be of a sufficiently mature level so that the impact of algorithms, models, etc. of those use cases on the resulting vehicle behaviour can be identified.

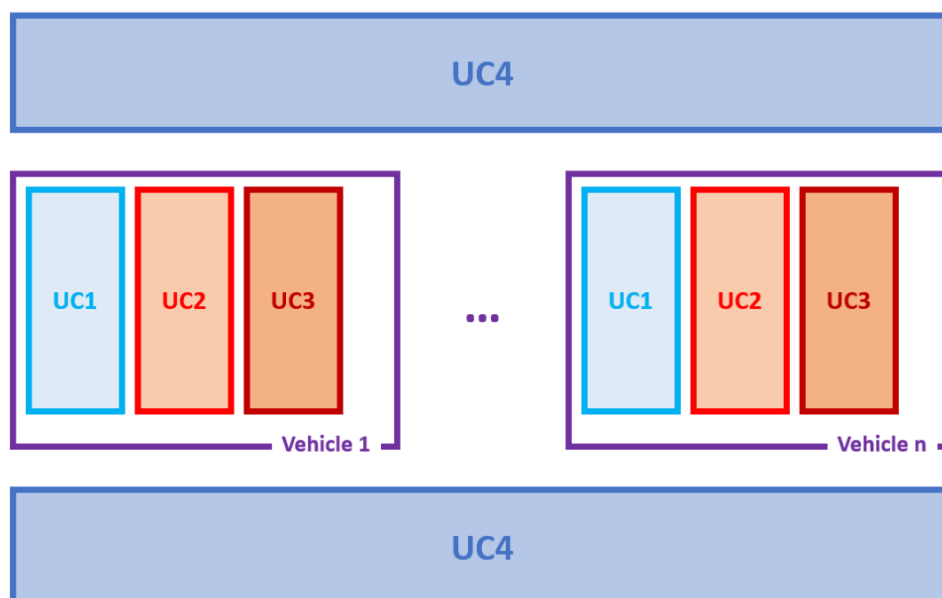


Figure 1: Overview of the interaction between the 4 Althena use cases

2.1. UC-1: Reliable Pedestrian Detection in Urban Environments

Approach and Overview

The development of trusted AI systems is imperative for comprehending object perception, sensor data utilization, redundancy, fusion, and discrepancy resolution. This UC seeks to address these critical aspects to foster reliable and explainable pedestrian detection in urban environments for CCAM applications.

The primary objective of this initiative is to facilitate the utilization of a pedestrian detection system in a use case demonstrator. This aims to enhance safety measures, especially in scenarios where pedestrian detection lies within the safety-critical path. Use Case 1 follows the Althena cycle, and it can be divided into two steps:

1st Althena Cycle “Offline” Demonstrator

In the first UC demonstration, an *offline* demonstrator is being showcased. The demonstration is based on recorded data from a demonstrator vehicle, simulated data or based on existing publicly available datasets on perception. This allows for faster prototyping, testing, and evaluation of different approaches developed in the project.

2nd Althena Cycle “Online” Demonstrator

The demonstration will be carried out with one of the demonstrator vehicles on one of the proving grounds. The tested and evaluated XAI methods from the 1st cycle are integrated into the software stack of the demonstrator vehicle and can be executed during runtime of the perception system. One of the possible demonstrator vehicles is shown in Figure 2. In addition to the demonstrator vehicle, static components of this UC will be showcased that explain and document the integrated AI functionalities in the real-world demonstrator.

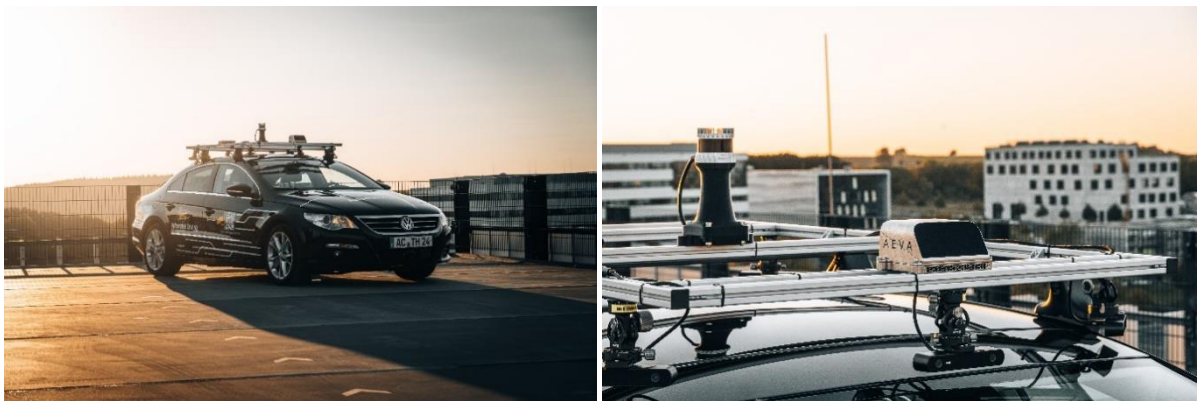


Figure 2: Possible demonstrator vehicle with mounted sensor rack consisting of multiple LIDAR and camera sensors.

The UC aims to showcase a multifaceted approach by integrating **Model-Driven**, **Data-Driven**, and **Sensor-Driven** methodologies. This comprehensive strategy ensures the reliability, explainability, and transparency of the pedestrian detection system and the associated AI functionalities across the software, sensor, and AI stack. The different strategies are researched and developed within the framework of WP3. The following sections provide an overview of these strategies.

Model-Driven Strategies

- **Explainable Layers (XAI T3.2):** Implementation of transparent layers within the model architecture to enhance interpretability and comprehension of decisions [5].
- **Visualization XAI Interface (XAI T3.2):** Development of a user-friendly interface to visualize the explainable components, fostering a better understanding of the system's decision-making process.
- **Model Cards (ML Life Cycle MGMT T3.1):** Creation and maintenance of model documentation to encapsulate crucial details across the machine learning lifecycle.

Data-Driven Approaches

- **Data Cards (ML Life Cycle MGMT T3.1):** Development and management of comprehensive data cards to document essential information throughout the machine learning lifecycle.

Sensor-Driven Fusion Techniques

- **Reliable Fusion of Sensor Modalities (XAI T3.2):** Integration of **multiple sensor modalities** to ensure robust and reliable pedestrian detection. Safety mechanisms are incorporated to mitigate discrepancies in data from various sources.
- **Explainable Multi-Object Tracking (XAI T3.2):** Implementation of explainable methods for tracking multiple objects, enhancing system transparency in monitoring pedestrian movements.

The fusion of these approaches is instrumental to enable accurate pedestrian detection based on diverse sensor data and to ensure the interpretability and reliability of the system, thus catering to safety-critical applications within urban settings.

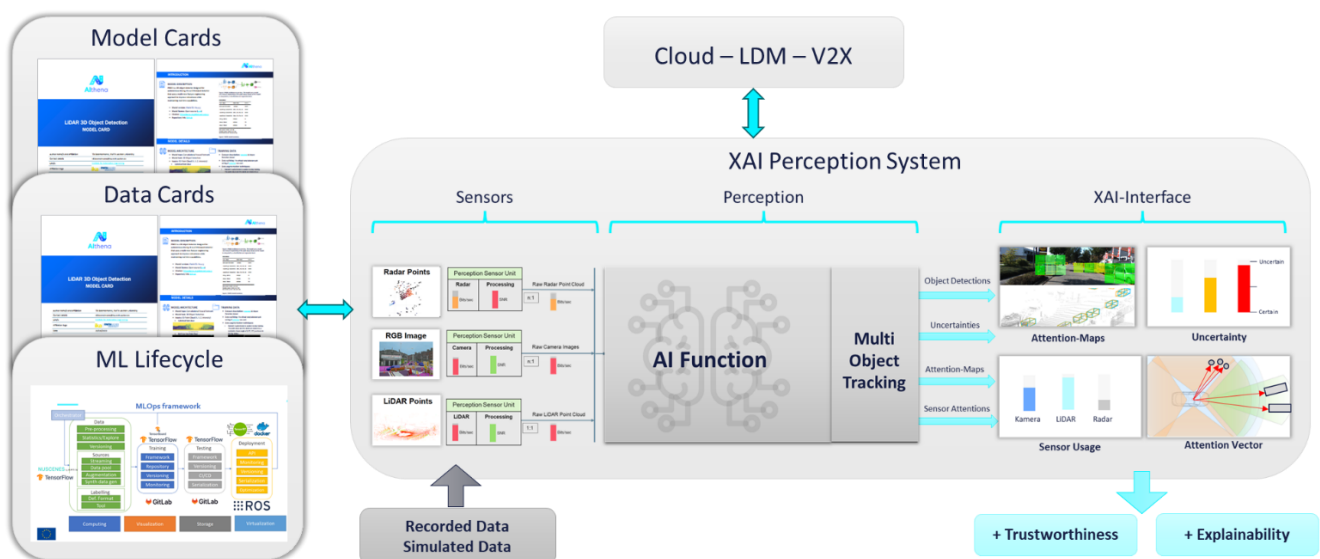


Figure 3: UC-1 1st Athena Cycle - "Offline" Demonstrator - Components

Figure 3 provides an overview of how the different components of UC-1 interact. **Model Cards**, **Data Cards** and the **ML Lifecycle** documentation are “static” components that describe and explain the AI functionalities used in the perception system. Inputs to the **XAI Perception System** are sensor streams from multiple sensor modalities that are generated by the demonstrator or by simulation. For the first Athena cycle, simulated and recorded data is being used. The AI “black box” functions are enriched with XAI methods that allow the user to get a better understanding regarding the inner workings and the state of the AI. This additional information is presented to the user via the “XAI-Interface”, which is a composition of several visualizations that translate the results of the XAI methods into a human understandable format during runtime of the system.

UC1 Target Audience/Stakeholders for:

- **Development Engineers and Researchers:** Development engineers are interested in the technical aspects of the AI models, including the inner working of the models, improving the model performance, robustness, or gather new training data to reduce biases.
- **System Users (End-Users):** Users are concerned with the overall safety and reliability of the system. They are interested in understanding how the integrated AI functionalities enhance safety, particularly in safety-critical scenarios. Clear visualization and explanation of system decisions are crucial for gaining user trust.
- **Certification Authorities and Regulatory Bodies:** Certification authorities are interested in the transparency and reliability of the pedestrian detection system. They need to be assured that the system complies with safety regulations and that the development process adheres to industry standards. Documentation such as model cards and data cards are vital for the certification process.

UC-1 Story & Scenario

1. **Ego Vehicle Setting:** The scenario unfolds as the Ego Vehicle navigates through a complex urban environment, sharing the road with various participants such as vehicles and pedestrians.
2. **Challenges Faced by Perception System:** The Perception System of the Ego Vehicle encounters a dynamic situation where conflicting information arises from different sensor modalities. The challenges include:
 - a. **Obstacle Blocking View:** An obstacle obstructs the line of sight to a pedestrian.
 - b. **Contradicting Detections:** The system receives conflicting detections from different sensors.
 - c. **Sensor Failure:** One or more sensors experience a failure.
 - d. **Weather Conditions Impact:** Adverse weather conditions influence the performance of certain sensors.
3. **Conflict Resolution by Perception System:**
 - a. **Option A - Successful Conflict Resolution:** The Perception System, enriched with the developed XAI methods, effectively resolves the conflicts arising from diverse sensor inputs. This involves making informed decisions on the presence and movements of pedestrians in the environment.
 - b. **Option B - User Information and Safe Mode Activation:** In cases where conflicts cannot be completely resolved, the system provides pertinent information to the user. The user is then presented with options such as transitioning to a safe mode or taking manual control of the vehicle.

Figure 4 depicts a scenario exemplifying challenge 2a.

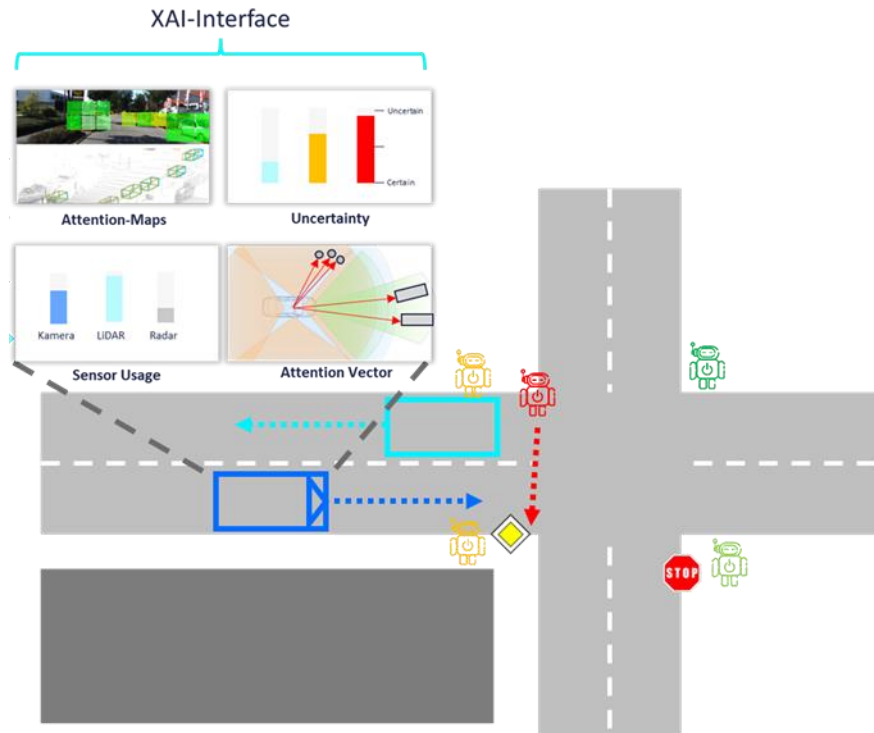


Figure 4: Schematic illustration of the UC-1 scenario. An urban scenario is considered where an obstacle blocks the view on a pedestrian.

2.2. UC-2.1: Collision prediction with Hybrid AI data fusion models

Collision prediction can be learnt with **AI models**, which learn from images or other raw data from sensors, and produce detected events, such as Time-To-Collision (TTC) values, Cut-in probability, or other equivalent collision-risk estimators, as forms of prediction of the near future road situation.

The evaluation of such AI models (in **real world scenarios**) imposes several challenges:

- Orchestrate a recording campaign with instrumented vehicles (with cameras, LIDAR, and other sensors) with the goal of capturing naturalistic driving situations, but also containing situations of interest for the evaluation of the AI functionality.
- Annotation of recorded data to generate ground truth, i.e., the certified or validated description of the reality captured in the recordings. This ground truth is needed to compare the output of the AI function with the correct or desired result and produce Performance Indicators (PI).
- Data used to compute PIs must be representative and relevant to the situations of interest. Real world recordings may contain very few or infrequent situations of interest, often known as edge cases. Filtering out content that is not representative and relevant is crucial to obtain meaningful PIs, both to decrease the necessary redundant testing and to save computational power for testing processes.
- Labelling perception-level ground truth can be semi-automated using already built AI models which perform part of the labelling task and leave the validation of the results to human operators. However, identifying edge cases is neither a trivial task for automated processes, as it implies some level of semantic understanding of the situation, nor for human operators as they may not have sufficient data or criteria to identify such situations.

A user of type **test engineer**, with the task of evaluating the AI model (aka the System Under Test, SUT), might find difficulties filtering out the raw material from recording campaigns to select those scenes that produce relevant/representative PIs. **Supervisors** or **other stakeholders** may want to understand the report by the test engineer and thus request that the engineer provides explanations for how the edge cases were selected.

The ground truth for collision-related information requires precise **perception-level ground truth**, i.e., spatio-temporal labels of the participants in the road scene, such as pedestrians, vehicles, etc. Geometries that model their physical appearance can be used, such as cuboids, polylines, or other geometries, while other labels and attributes can define the object class (e.g., “pedestrian”, “car”). The temporal dimension provides the ability to model the dynamicity of such information, and thus label trajectories of time-consistent objects, actions happening during specific time intervals, instantaneous events triggering actions, etc. The ASAM OpenLabel 1.0.0 standard is the first international standard that defines how to represent perception-level labels with the required complexity.

However, finding **edge cases in real world data** is a difficult task, as these edge cases may only be representable in the form of semantic expressions. For instance, a certain scene may contain a near-miss or a cut-in situation, which is often more difficult to label for human operators, but also for machines which may have not the ability to detect such situations accurately.

A **Semantic Labelling System (SLS)** is proposed to find edge cases, running logical inference on user-defined rules over perception-level annotations. These rules contain the expert knowledge that defines the actions of interest. Actions and rules can be human-interpretable and thus seen as semantic explanations of both what the scene contains and how the edge cases were found. These explanations are suitable for consumption by different user types, e.g., test engineers willing to filter out their own recordings as well as supervisors or customers that want to understand what scenes were selected and why.

The main objectives are to produce a Semantic Labelling System (SLS), using Graphical Database technologies and a set of rules that detect edge cases and create rich explanations of the scene. Other objectives are to define a data model for perception-level ground truth in a semantic graph to enable execution of rules, based on ASAM OpenLabel 1.0.0, and to demonstrate the utilisation of the SLS on a large dataset to discover edge cases where SUT performance can be evaluated.

The work consists of several steps:

1. Select a dataset of road scenes, with recorded images, LIDAR, or other sensorial data, which is already labelled at perception-level, i.e., it contains sufficiently rich object-level data, such as the presence, position and class of objects around the ego-vehicle.
2. Produce a semantic data model devised to semantically model relationships between objects, using ASAM OpenLabel 1.0.0 as labelling format due to its inherent capabilities to model semantic relationships.
3. Convert the object-level annotations of the selected dataset into the semantically enriched data model.
4. Define edge cases of interest (NOTE: make sure the dataset contains some).
5. Develop the SLS and load the semantically enriched data into the SLS.
6. Prepare semantic queries addressing the edge cases.
7. Run the semantic queries against the SLS and verify that the results are the expected edge cases.

The main innovation is the creation of a SLS that can produce semantically rich detections as explanations of the scenes. The SLS can run expert rules, which are user-defined and easy to read by humans, to increase the interpretability of the search process.

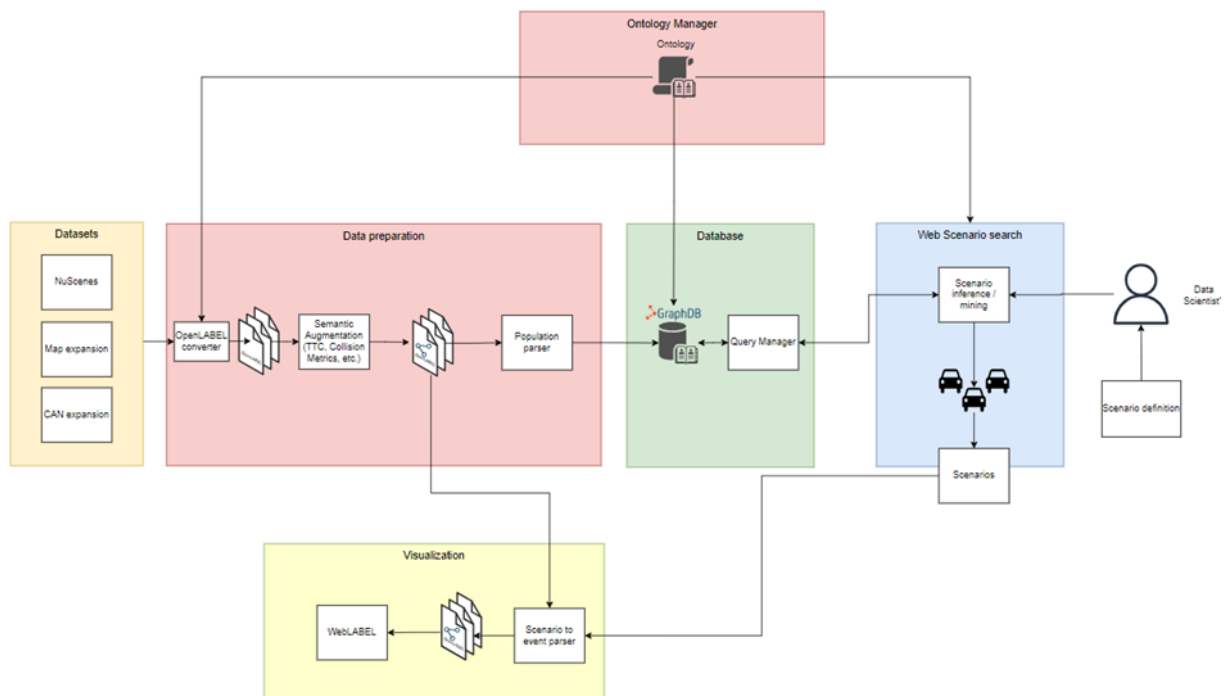


Figure 5: This diagram shows the technical components behind the solution.

This use case aims to produce a component, the SLS, which is not AI-based, but that can be used in the context of the evaluation of AI models like the Hybrid-AI based Collision Risk Prediction model.

The key technologies for this use case are:

- Graphical databases: a database that structures information as graphs and enables execution of semantic queries, e.g., Neo4j, GraphDB.
- Semantic Query Language: a language that permits the construction of rules as queries, to be run as logical inference queries against the graphical database, e.g., SPARQL, Cypher.
- Ontologies: abstract constructs that define classes, properties, and relationships, materializing a data model. Ontologies can be created with semantic languages like OWL or RDF.

2.3. UC-2.2: Robust Prediction modules for Robo-taxi in urban environment

The goal of this demonstrator is to develop a trustworthy and robust prediction of traffic participants' intended motion, enabling a safe and predictable operation of robo-taxis.

In interactive urban traffic environments, vehicles as well as pedestrians and other traffic participants navigate on highly complex road networks under a variety of environmental conditions while interacting with different kinds of road users. In this context, motion planning can only guarantee the safety of all participants, if the characteristics of the scenarios are acknowledged. This is called situational awareness. Figure 6 shows the block diagram for demonstrator 2.2, and Figure 7 gives an overview of the specific steps to be performed to demonstrate the use case overall.

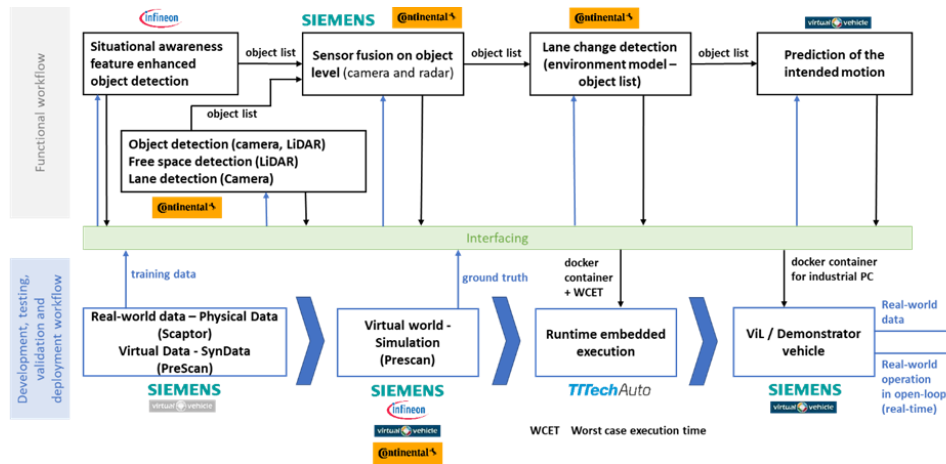


Figure 6: Block diagram: Robust prediction modules for Robo-taxi environments.

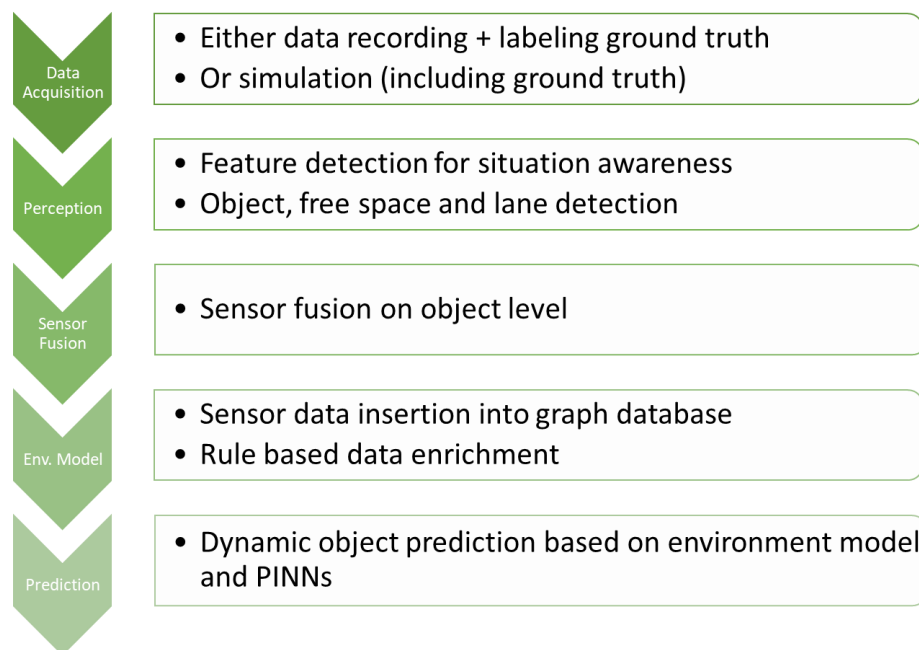


Figure 7: Information flow of prediction based on environmental awareness.

The first step in this UC-2.2 is to generate real-world data that will be used for training and validating machine learning models. This involves collecting data using their own data acquisition system, filtering, cleaning, and visualizing the data that will be used as input to the machine learning models. In the context of this UC, all involved partners were requested to provide their requirements for generating such datasets and vehicle setup, variety of scenarios, and ODDs which were considered when performing the necessary recording campaigns during Task 2.1. This step was vital since it is important to define from the beginning in which scenarios, ODDs, and sensor setup the machine learning model will be validated, and under which circumstances it is able to operate. Dedicated real-world datasets will be shared according to the Siemens GDPR guidelines with the involved partners for labelling the datasets and further refining automatically generated annotations using pretrained machine learning models.

An X-in-the-loop methodology will be developed to bridge the gap between virtual and physical testing. Due to the limited person months in the Athena project, the approach will be to prepare the setup for ViL testing and have a functional Drive-By-Wire for future integration of ML-based controllers. The main objective is to understand how the vehicle under test responds when a controller uses data generated from Simcenter Prescan and how the developed machine learning model can be deployed on an ECU in real time. The model will be tested for its performance insufficiencies in a wide coverage of the relevant driving scenario space (defined from the involved partners in Task 2.1), such as insufficient object detection, tracking and trajectory estimation, false negative and false positive predictions. The validation methodology assures that based on examination and tests the acceptance criteria will be achieved with a sufficient level of confidence. Necessary data analysis will also be conducted in the defined requirements, environmental conditions, and triggering conditions for misbehaviour of the model. Reports will be generated on the root cause of a possible machine learning model failure (e.g., the machine learning algorithm failed to accurately model the surrounding environment). Acceptance criteria that need to be achieved with a degree of confidence for validating the model will be the probability that the machine learning model is not controllable in the defined scenarios, the severity of the accidents when performing these tests, and the number of incidents leading to an unexpected behaviour.

The steps of data acquisition, perception, sensor fusion, environmental modelling, and prediction for situational awareness form a *computation chain*. System virtualisation and/or containerisation, and a common middleware facilitate independent development and testing of software components that are part of such a computation chain. We use ROS2 as a middleware for communication between independently developed components and Docker containers to package ROS2 applications that form a software component. The correctness of a computation chain depends not only on the correctness of its computational result but also on the timeliness of the computation. The timing requirements of a computation chain, i.e., the periodicity of its execution and the time until a new computational result must be available within a period, are dictated by the real-world process. This *timing information* of containerized software components, namely, their *period, deadline, and worst-case execution time (WCET)*, acts as input to an offline schedule generation before the integration of its components on the embedded platform. Executing such a generated schedule will ensure the timing of a computation chain. The resulting scheduling problem is an NP-complete problem, meaning given a collection of software components with their timing information and a predefined number of embedded platforms, it is hard to find a correct schedule. However, it is easy to verify a schedule's correctness after it was generated. As a result, to ensure the correctness of generated schedules during design time, we must first formulate constraints for the scheduling hierarchy formed by the time-triggered container execution and the ROS2 instances in each container scheduling its respective applications. Second, the containerized software components as part of a situational awareness computation chain have data dependencies, i.e., the output of one software components acts as input of one or more preceding software components. Finally, the situational awareness computation chain monitors the real world that dictates the so-called end-to-end latency of the performed computation, meaning the maximum time it takes from the first software component in the chain acquiring data to the last software component generating a prediction.

Another contribution to this UC is to develop a sophisticated algorithmic framework for advanced perception in autonomous vehicles. This framework is crucial for the reliable operation of robo-taxis, ensuring they can effectively detect objects, identify free space, and recognize lane boundaries. This is a fundamental step before predicting lane changes, which is essential for situational awareness and safety in complex urban environments.

To provide a clear understanding of the perception system, let us explain in more detail the distinct algorithms and their sensor-based functionalities:

- The **Object Detection** algorithm relies on data from both the frontal camera and LIDAR. Frontal cameras provide high-resolution images that are essential for recognizing and classifying various objects such as vehicles, pedestrians, and other elements in the vehicle's visual field. LIDAR complements this by providing depth information.
- The **Lane Detection** algorithm is primarily based on frontal camera data. These cameras capture road markings and lanes, and the algorithm processes these images to delineate the current lane structure. This information is critical for maintaining the robo-taxi within its lane and for executing safe lane changes, as it establishes a reference for the vehicle's positioning on the road.
- The **Free Space Detection** algorithm uses LIDAR data processed through the ROS2 Foxy framework. This algorithm, which relies on the dataset provided by Siemens, receives input in the form of a standard point cloud, as defined by ROS2's official format. Its output, `FreespaceOutput.msg` enables the algorithm to deliver a multi-dimensional understanding of the vehicle's surroundings, critical for the safe navigation and operational efficiency of autonomous robo-taxis in complex urban landscapes.
- The **Data Fusion** algorithm synthesizes the information gathered by both cameras and LIDAR. This algorithm ensures that object and lane detection data are not considered in isolation; instead, it creates a unified environmental model that the lane change prediction system can use.

The **Lane Change Prediction** algorithm is a vital component that synthesizes all the data gathered from the previous algorithms — object detection, lane detection, and free space detection. Its primary role is to predict when vehicles are likely to change lanes. In fact, by analysing the comprehensive environmental model generated by the Data Fusion algorithm, the Lane Change Prediction algorithm can anticipate potential movements of other vehicles. This foresight is critical for the robo-taxi's decision-making process, enabling it to adjust its course or speed proactively to avoid collisions and ensure a safe driving experience. Therefore, this prediction is crucial for the autonomous vehicle's safety and smooth operation in busy urban environments. In conclusion, the framework aims to enhance the safety and efficiency of robo-taxis in urban settings aligning with Althena's purposes. Each algorithm — from Object Detection and Lane Detection to Free Space Detection and Data Fusion — plays a specific role in creating a detailed and accurate understanding of the vehicle's surroundings and results in the development of the Lane Change Prediction algorithm, a crucial component that brings together all the processed data to anticipate and react to dynamic traffic conditions.

Finally, assuming that the output of the perception system is provided along with available map information, an environmental model can be created, and the prediction of the comprising traffic participants' motion can be performed. Each iteration of perception system processes, with the additional task of object-level data fusion, is expected to be completed in real time, ensuring continuous awareness of the traffic situation. However, situational awareness goes beyond simply collecting information on the traffic situation by inferring connections between heterogeneous sources of information. A comprehensive situational awareness would have to, for example, define the relations between each update of the traffic participant's state and the known road infrastructure. These relations could also be semantic, such as those establishing relations between individual traffic participants. Therefore, defining an extensive list of relations can have a significant impact in furthering the understanding of the situation.

Once the understanding of the traffic situation is established, it can be translated into a format understandable to the motion prediction model itself. The transformed input can then be used to generate potential trajectories for each detected traffic participant. For the accuracy of the predictions, it becomes essential that detailed traffic information, both direct and inferred, is passed through the environment model onto the input of the prediction model. Such input ensures the introduction of the observational bias, i.e., the conditioning of the model and, consequently, the predictions based on the input provided at the training stage. However, the feasibility of the predictions cannot be guaranteed with observational bias alone [1]. Instead, the model must incorporate constraints based on prior knowledge (e.g., physical laws). These constraints can be reflected in the model's architecture and learning algorithm, known as inductive and learning bias, respectively. Developing a model with these biases also supports AITHENA's general goal of increasing the explainability of the used models. Should the operation of a previously black-box model become more understandable to the developer, the explanations generated for the users would become more trustworthy. Therefore, the choice of KPIs must meet the accuracy and feasibility standards necessary for the safe and comfortable navigation of a robo-taxi. Both sets of indicators must account for the particularities of the urban environment in which the robo-taxi operates (e.g., the prediction time horizon or information on traversable areas).

2.4. UC-3: Trustworthy and Human understandable decision-making

Use Case 3, called Trustworthy and Human understandable decision-making is related to the task of vehicle guidance in an automated driving software stack. According to Section 2, the local environment is perceived through the vehicle sensors and perception algorithms, and the modules of situation awareness are responsible for combining the information derived from the sensors with additional information like map-data or information received through external communication with other traffic participants or digital infrastructure. Based on this information, the modules responsible for vehicle guidance need to generate reasonable decisions that lead to vehicle trajectories that are followed by applying the vehicle controls.

The focus of this use case and the corresponding demonstrator is to explain decisions to the user of the system before, while or after they have been executed. Next to this, trustworthiness should increase by providing a robust system that is capable of handling situations in an understandable manner, even if the present situation confronts the vehicle with unforeseen circumstances.

The use case objectives will be tackled through a combination of a Hybrid-AI Motion Planning Stack in combination with a Situation Awareness Module measuring the competence of the utilized Neural-Network based Behaviour-Generation module in specific situations. This setup intends to increase the robustness of the entire system in unforeseen situations. In addition, non-intuitive behaviour of the vehicle should be explained to vehicle occupants. Therefore, this use case will focus on V2X information which is not directly perceivable by the vehicle occupant. This shows a gap in the level of information between the occupant and the vehicle which may require additional explanation by the system.

The implementation of the demonstrated algorithms is carried out within the scope of WP3, more precisely in Task 3.5. For the development and testing of these algorithms, simulation environments and scenarios will be used in an initial phase, which will be developed as part of WP4. Furthermore, data sets provided through WP2 may be used for the development of corresponding ML models. Finally, the integration of the developed algorithms in simulation scenarios as well as demonstrator vehicles is carried out as part of WP5. In addition, a testing and validation methodology is developed as part of WP5. Requirements, test scenarios, and evaluation metrics will be defined to evaluate the algorithms and methods developed for the realisation of this use case.

Functional Description

The task of vehicle guidance can be separated into three different tasks namely navigation, guidance, and stabilization [3]. On a functional level, the algorithms act sequentially, meaning that the output from the navigation level, that is usually a route in a road network, is an input to algorithms acting on the guidance level and so on. Moreover, the generation of routes, behaviour decisions, trajectories and vehicle controls are repeated cyclically. Usually, the frequency for the recalculation increases from the navigation to the stabilization level. The different algorithms acting on the specific levels of vehicle guidance use different approaches like graph- and tree-search, numerical optimization, deep neural networks, or different control techniques like model-predictive-control. Thus, inputs, outputs and the optimization objectives vary for all these algorithms. Nevertheless, the overarching objectives of the vehicle guidance system are the optimization of comfort, efficiency, and safety.

The algorithms that are demonstrated in this demonstrator are developed in WP3, more precisely in Task 3.5: Explainable and robust decision-making. Figure 8 shows an abstracted functional architecture of the software modules that are demonstrated through UC3.

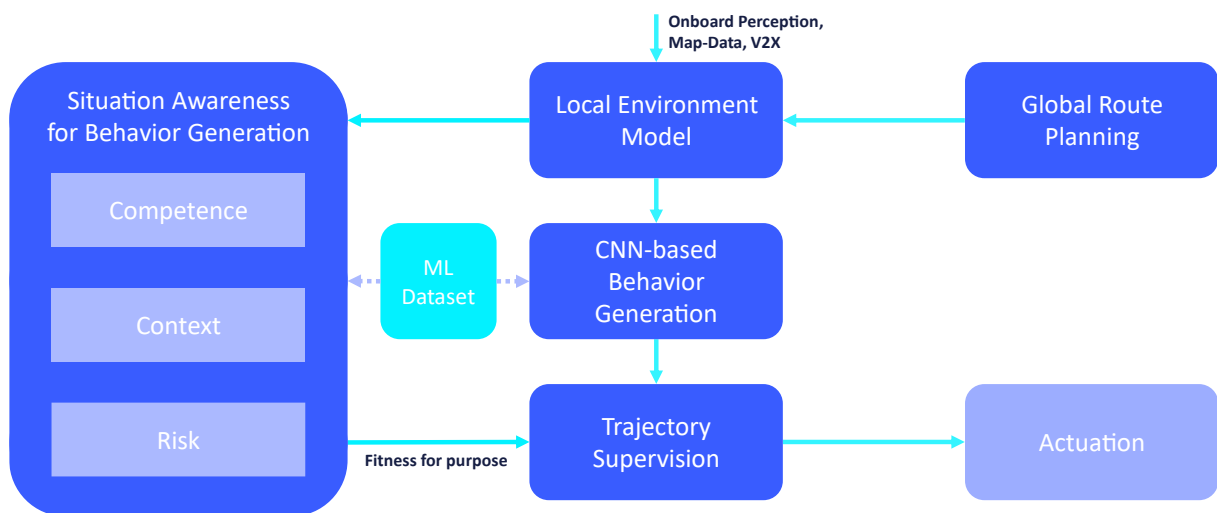


Figure 8: Abstracted functional architecture of software modules demonstrated in UC3.

The functional architecture can be subdivided into the actual motion-planning software stack and the situation awareness module that evaluates the Convolutional Neural Network (CNN)-based behaviour generation module of the motion-planning stack.

Hybrid-AI Motion Planning Stack

As illustrated in Figure 8, the motion planning stack is composed of several modules for the different vehicle guidance tasks. First, a global route from the current vehicle position to a given destination is planned through the *Global Route Planning*. The resulting route is fed into the *Local Environment Model* that combines additional information provided by the vehicle perception modules as well as map data and V2X information. As the term *local* indicates, all information in the local surroundings of the vehicle is extracted and prepared for further processing in the following modules.

At a first stage a driving behaviour is generated through a neural-network in the *CNN-based Behaviour Generation* module. More specifically, since a CNN is used, the input to this neural network is represented as an abstracted Birds-Eye-View representation of the vehicle surrounding. Output of the neural network is a chronological sequence of future vehicle states, also referred to as trajectory.

This trajectory is fed into the *Trajectory Supervision* module, which is responsible for evaluating the previously generated trajectory. For this purpose, a set of different rules (e.g., constraints in driveability, occurrence of collisions...) is evaluated and, if necessary, the specified trajectory is modified slightly. If the CNN-based trajectory planning cannot provide a satisfactory solution, alternative actions can be initiated by the *Trajectory Supervision*. The output of the *Trajectory Supervision* is also a trajectory for which it is ensured that the vehicle controllers are capable to follow.

Both CNN-based Behaviour Generation as well as Trajectory Supervision operate at the guidance level. The control algorithms then take over the final task of vehicle guidance, namely the stabilization. The development of the control algorithms is out of scope of this project, even though they are essential for automated vehicle control to close the control loop.

Competence Observation

To achieve an overall robust guidance system, the competence of the CNN-based Behaviour Generation in specific contexts shall be observed. Competence is related to experience. Situations and environments that have been encountered more frequently and in greater variety can be handled with more competence than their unfamiliar counterparts. To make statements about the competence within a given situation (or context), comparisons to previously experienced contexts can be made to find similar examples on which to base the decision-making process. The more examples that are like the present context exist, and the closer their resemblance, the higher the expected competence within the current context is.

We propose the modelling of road contexts using a knowledge graph database. A knowledge graph aims to model knowledge using a graph structure, which contains nodes, edges, and attributes, and focuses on modelling the relationships between entities. This can translate to nodes modelling lane segments, with edges between them indicating the directional flow of traffic, as well as actors and infrastructure objects interacting with the road layout (e.g., a traffic light being positioned on a specific lane segment) or with each other (e.g., a vehicle having to give priority to another vehicle). This allows for storing contexts as queryable structures (the knowledge graph database) that can be used for comparing a current context with previously observed ones. The knowledge graph aims to model both explicit (road structure, actors) and implicit context (traffic rules, social norms), which can be used to compare the currently observed context the vehicle is finding itself in.

The goal is to match the perceived context with patterns in the knowledge graph database to find similar graph structures to estimate competence.

For instance, in the scenario where the Ego Vehicle is approaching a roundabout, which is a road-structure that is rarely observed in North American data samples, we would expect to observe a decrease in competence as the vehicle approaches the roundabout, since it transitions from a context it is competent in to one it is less familiar with.

In situations in which the measured competence is low, the vehicle should account for this: In the presented motion planning architecture, the knowledge on the competence in an experienced situation can be fed back to the Trajectory Supervision module. Based on this, suitable actions for the vehicle can be derived.

UC Scenarios

Use Case 3 will demonstrate specific scenarios that target the explanation of non-intuitive situations on the one hand, and the robustness against unknown contexts and situations on the other hand. We will demonstrate two scenarios: “*Explanation of non-intuitive braking due to occluded vehicle*” (SCEN_09) and “*Robustness of Hybrid AI planning stack against unknown contexts*” (SCEN_10). Figure 9 and Figure 10 show a schematic illustration of the proposed scenarios. For a further description of the proposed scenarios refer to section 3.2.

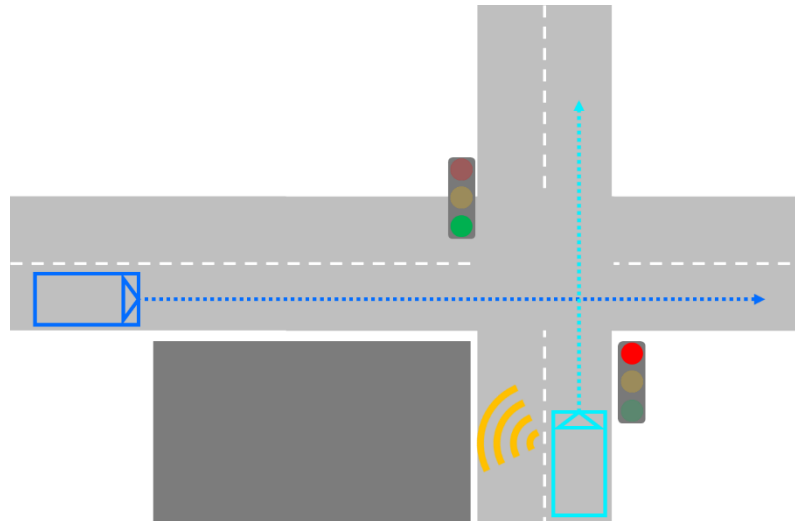


Figure 9: Schematic illustration of SCEN_09 demonstrating the explanation of non-intuitive braking due to an occluded vehicle.

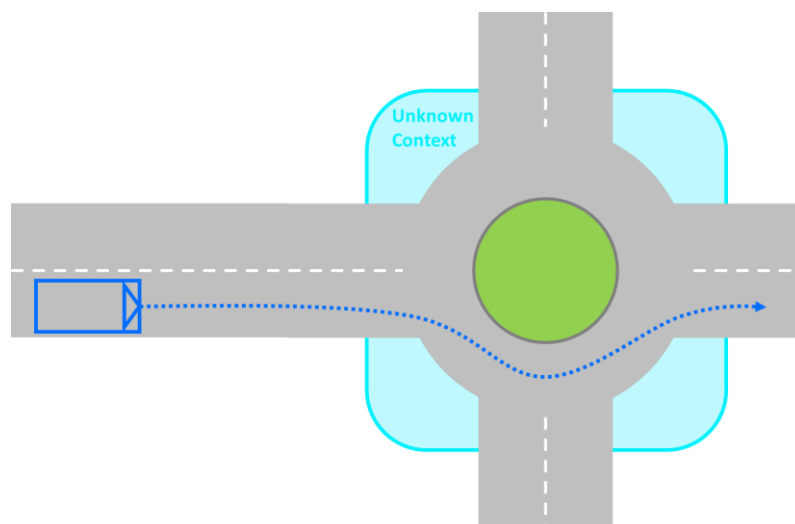


Figure 10: Schematic illustration of SCEN_10 demonstrating the robustness of the Hybrid-AI planning stack against unknown contexts.

2.5. UC-4: AI Models and Traffic Management

Automated vehicles (AVs) currently behave differently than vehicles driven by humans. This different behaviour has an impact on traffic dynamics. To understand the scope of that impact, the behaviour of AVs needs to be studied at the transport level. To be able to benchmark the impact of the AV behaviour, a reference is needed. The reference is what is considered “good behaviour”, also referred to as “desired behaviour” or “acceptable behaviour”. This is the behaviour that will be applied to non-AVs.

To study the dynamics on the transport level, macroscopic scenarios will be used. These scenarios will be implemented and evaluated using microsimulation software (e.g., SUMO, Vissim, Aimsun). For the non-AVs, existing vehicle models in such software will be used. However, since the focus is on some specific situations that are relevant to the other three use cases and/or from a traffic management perspective, the behaviour of these existing vehicle models will be evaluated and, if possible, adapted, using insights gained from the other use cases and through literature study of ‘acceptable’ behaviour.

For the AVs, i.e., the ‘Althena vehicles’, the results of the other Althena use cases and demonstrators will be the basis. Using the output of those demonstrators, specific behaviours of the vehicles can be identified. Those behaviours can then be parameterised, and those parameters will be used to adapt an existing vehicle model to the microsimulation software (a different one than the one used for the non-AV) such that it reflects the behaviour of Althena vehicles.

Next, the microsimulation can be run using only non-AVs and different mixes of non-AVs and percentages and types of Althena vehicles (e.g., 10%, 20%, 50%). In addition, different levels of traffic demand, expressed as the Level of Service (LOS) [2], will be used to assess the potential influence of light or heavy traffic conditions. KPIs will be used to evaluate the performance of the traffic dynamics with respect to network efficiency, traffic safety, and environmental impact (sustainability).

The above focuses on examining the influence of the AI models in AVs on the traffic dynamics at the transport system level. On the other hand, traffic management (TM) systems can influence the behaviour of AVs by providing specific information to the AI models. Such information can support the situational awareness of the vehicle to increase comfort, safety, and efficiency. There are different levels at which traffic management system information can play a role:

- On the operational level, the TM systems can provide information about other actors and their trajectories to support object avoidance and trajectory planning.
- At a tactical level, information about traffic conditions, obstacles, and traffic light status (even when not in line of sight) can support AVs in their short-term planning (e.g., lane change choices, braking and accelerating in the context of upcoming intersections or road works, etc.).
- Lastly, on the strategic level, TM systems can support AVs in their navigational choices to circumvent delays to optimise their travel time (individual) or the network travel time (collective). In addition, the TM systems might know about situations that are outside the Operational Design Domain (ODD) because of (semi-)dynamic factors (traffic composition, road works, weather, damaged infrastructure, accidents, etc.). That information can help AVs plan different routes that do not cause the automated driving system (ADS) to disengage.

Depending on the scope and situations studied in the other Athena use cases, Use Case 4 will examine which TM information can support the AI model(s) to improve comfort, safety, and/or efficiency. More specifically, the ODD of the AV might benefit from or require some outside (i.e., TM) information source to maintain its automated state. In the TM4CAD⁴ project the above is part of the Distributed ODD attribute Value Awareness (DOVA) concept. This is a theoretical framework about the exchange of ODD information between AVs and other traffic systems. Use Case 4 is used to explore the technical implementation of (parts of) that theoretical framework.

Using TM information, the AI models can adapt and improve their behaviour, thereby closing the gap towards the “good behaviour”. Taking that into consideration, the scenarios initially run without TM support can be rerun with TM support to assess the benefit TM information might provide to AVs.

With respect to the AI lifecycle, Use Case 4 primarily focuses on the Evaluate & Analyse phase, although the other use cases must iterate through the other phases to incorporate the concept of adapting to TM information. In the end, Use Case 4 will provide transparency and predictability with respect to the ways in which AVs could impact the traffic dynamics on the transport level and how TM information could limit that impact and/or improve the comfort, efficiency, and safety for the individual AV (user).

⁴ <https://tm4cad.project.cedr.eu/>

3. Validation approach for each use case

The requirements for the individual use cases are based on the use case descriptions in chapter 2. Some requirements are also relevant for several use cases. These are listed and described in Table 1. Furthermore, the benchmark scenarios of the individual use cases are described in this chapter (see Table 2), and key performance indicators are defined (see Table 3).

3.1. High-level requirements

Table 1: List of requirements of the 4 Athena use cases.

ID	UC	Description	Rationale	Verification
REQ_01	2.2	Software architects shall specify timing information for containerized software components (WCET estimate, period, deadline with deadline \leq period).	Input to schedule generation for Linux-based time-triggered container execution.	Review of containerized software components' timing information.
REQ_02	2.2	End-to-end latency requirements computation chains shall be provided.	Input to schedule generation ensuring timeliness of computation chains.	Measure integrated computation chain's end-to-end latency.
REQ_03	2.2	Data dependencies between containerized software components shall be provided.	Input to schedule generation ensuring adherence to precedence constraints in computation chains.	Verify orderly execution of integrated software components, i.e., adherence to precedence constraints.
REQ_04	2.2	The prediction module shall incorporate one or multiple design components (biases) enhancing the explainability of the overall model.	Support the general goal of trustworthy AI development.	Report on the model architecture and implementation details.
REQ_05	2.2	The prediction module shall provide accurate, feasible, and robust trajectory predictions for a variety of urban traffic scenarios.	Enable safe operation of a robo-taxi in an urban environment.	Report on predefined KPIs with respect to validation scenarios.
REQ_06	2.2	The prediction module shall provide long-term horizon predictions of traffic participants' trajectories.	Enable timely reaction of a robo-taxi interacting with numerous traffic participants.	Report on predefined KPIs with respect to length of prediction horizons.
REQ_07	2.2 / 4	The prediction module shall aggregate heterogeneous scenario data provided by multiple data sources (e.g. prediction system, traffic management system, map information).	Provide comprehensive situational awareness.	Report on scenario data usage and processing.

REQ_08	2.2	During WP2 the system should define the sensors that will be needed, data formats and the scenarios/ODDs that will be recorded.	Support on providing the setup for data acquisition and generating datasets.	Report on the mentioned requirements.
REQ_09	2.2	Establish initial criteria for validating simulation accuracy against real-world data and verifying the simulated vehicle's behaviour against reality.	Understanding how a simulated vehicle responds in various conditions allows for the optimization of vehicle designs, control algorithms, or components for better performance and efficiency. Input will be needed from partners on how to include the controller for ViL testing.	Evaluate the performance of sensors by comparing the outputs of simulated sensors (such as LIDAR, radar, cameras) with data collected from actual sensors installed on physical vehicles in similar conditions.
REQ_10	2.2	The Situational Awareness Feature Enhanced Object Detection (SAFEOD) function block shall detect dynamic objects such as trucks, cars, motorbikes, bicycles, and pedestrians.	The requirement ensures a proper detection of specific dynamic objects.	Correlation in terms of meeting the stated KPIs (IoU, F1 score and mAP) twice with a maximum difference of 10%, once using virtual data and once using data sets.
REQ_11	2.2	The SAFEOD function block shall detect static objects of type traffic light, including the current active traffic light colour.	The requirement ensures a proper detection of a specific static object.	Correlation in terms of meeting the stated KPIs (IoU, F1 score and mAP) twice with a maximum difference of 10%, once using virtual data and once using data sets.
REQ_12	2.2	The SAFEOD function block shall detect activated vehicle signal lights of type brake and left/right turn indicators related to dynamic objects of type car.	The requirement ensures a proper detection of specific features related to dynamic objects.	Correlation in terms of meeting the stated KPIs (IoU, F1 score and mAP) twice with a maximum difference of 10%, once using virtual data and once using data sets.
REQ_13	2.2	The SAFEOD function block shall detect the bounding box orientation of dynamic objects of type car and truck.	The requirement ensures a proper detection of the bounding box orientation.	Correlation with virtual data and data sets.

REQ_14	2.1	Semantic Labelling System (SLS) shall be able to load an ontology.	The ontology ensures proper understanding of scenarios and edge cases.	Functional system	SLS
REQ_15	2.1	SLS shall be able to run logical inference queries.	Ability to extract edge-case scenarios from the data model.	Functional system	SLS
REQ_16	2.1	Logical inference queries shall be writable as expert rules and be human-readable.	Usability of the system by different stakeholders.	Test with different stakeholders (technical and non-technical).	
REQ_17	2.1	The ontology shall contain actions related to collision prediction, such as hard braking scenario, hard acceleration scenario, lane changing, pedestrian crossing, cut in, cut out, overtaking, speeding, car following, and near miss collision.	The requirement ensures the ability of the SLS system to detect edge cases for TTC prediction.	Capability of the SLS system to extract relevant edge-case scenarios from the data model.	
REQ_18	2.1	An open dataset with object-level annotations shall be used, which contains some of the edge cases defined in RF5.	Ensure unbiased and reliable testing scenarios.	No validation required.	
REQ_19	2.1	The dataset shall be converted to OpenLabel 1.0.0 format.	OpenLabel format is the first and only international standard that harmonises heterogeneous label content for multiple type of applications.	Validation with OpenLabel Parser.	1.0.0
REQ_20	2.1	The resulting detected edge cases by the SLS shall be writable using OpenLabel 1.0.0 format.	OpenLabel format is the first and only international standard that harmonises heterogeneous label content for multiple type of applications.	Validation with OpenLabel Parser.	1.0.0
REQ_22	2.1	Openly available datasets, properly anonymized shall be used.	Ensures security and privacy of the data.	No validation required.	
REQ_23	2.1	The rules shall be human-readable for users of type test engineers.	Usability of the system by different stakeholders.	Test with different stakeholders (technical and non-technical).	

REQ_24	2.1	The rules shall be extensible to controlled natural language to be human-readable by non-technical stakeholders.	Usability of the system by different stakeholders.	Test with different stakeholders (technical and non-technical).
REQ_25	3	The vehicle follows a given route from the current vehicle position to a given destination.	Satisfies the overall goal of an automated vehicle.	Validated through demonstrator scenarios.
REQ_26	3	The implemented vehicle guidance system is able to keep the vehicle within its respective driving lane along the route.	Major objective of lateral vehicle guidance task. Staying in its respective driving lane ensures a safe behaviour.	Report on predefined KPIs with respect to lateral vehicle guidance.
REQ_27	3	The implemented vehicle guidance system is able to adjust the vehicle speed according to the vehicle environment.	Major objective of longitudinal vehicle guidance. Ensuring to safely stop in case of a various objects blocking the driving lane or traffic rules that prohibit to continue driving.	Report on predefined KPIs with respect to longitudinal vehicle guidance.
REQ_28	3	The vehicle is able to receive, and process specified V2X information. More specifically the Cooperative Awareness Message (CAM) should be processed.	Required to demonstrate SCEN_09.	No validation required.
REQ_29	3	The vehicle considers Cooperative Awareness Messages provided through other vehicles in its surrounding. If a potential collision is detected, the vehicle will decelerate or come to a standstill.	Showcasing the potential of V2X communication by increasing the safety of connected vehicles in situations with occlusion.	Comparing the lateral and longitudinal vehicle guidance KPIs with and without V2X communication.
REQ_30	3	To explain the vehicle behaviour based on V2X information to the occupant, the additional knowledge should be made accessible to the occupant through visualisation.	Especially V2X information is not directly perceivable for the occupant of the vehicle, requiring an additional interface for explanation.	Test with different stakeholders (technical and non-technical).

REQ_31	3	To demonstrate the capabilities of the Competence Observation module it is required that the dataset does not contain training samples with roundabouts.	Ensure that there are samples which are out of the domain of the training set for the CNN-based behaviour Generation module.	Report on the composition of the road layouts that are present in the dataset.
REQ_32	3	The Competence Observation is able to measure the competence of the CNN-based Behaviour Generation module based on the context the vehicle is located in. This use case will focus on roundabout scenarios as unknown context.	Improve the general trustworthiness of the behaviour generation module.	Verify that the competence decreases when measures in contexts that are not in the training set compared to ones that are. Correlation between the competence and performance of the behaviour generation module.
REQ_33	3	The implemented vehicle guidance system considers the competence indication provided by the Competence Observation module. The system will decrease the trust in the predicted trajectory provided by the CNN-based Behaviour Generation module and initiate an alternative action.	To provide a robust overall system, competence of ML based models for behaviour generation should be monitored since their exact limitations may not be clearly defined.	Comparing the lateral and longitudinal vehicle guidance KPIs with and without competence observation.
REQ_34	1	The perception system is able to process multiple sensor streams from multiple modalities in real time.	The perception system must run in real time in the demonstrator vehicle.	The throughput measured in FPS should be faster than the sample rate of the slowest sensor.
REQ_35	1	The perception system can detect pedestrians in the near field (<20m) of the Ego Vehicle with very high accuracy.	Reliable detection of pedestrians is necessary to avoid collisions.	Measured with typical metrics for 3D Object Detection such as mAP, F1-Score, Accuracy, Recall, or TP and FP rates.
REQ_36	1	The system shall provide a user-friendly interface, known as the "XAI-Interface," to visually present conflict resolution decisions and information to the user during the runtime of the system.	Enhancing transparency and user understanding of system actions is critical for user trust and intervention when necessary.	Conduct usability testing to ensure that the XAI-Interface effectively communicates conflict resolution information to the user.

REQ_37	1	The pedestrian detection system shall demonstrate robustness in handling dynamic scenarios, including obstacles blocking the view, conflicting sensor information, sensor failures, and adverse weather conditions.	Ensuring robustness is crucial for the safe operation of the Ego Vehicle in complex urban environments with diverse challenges.	Conduct simulation and real-world tests to validate the system's performance in various challenging scenarios, including those mentioned in the UC-1 Story & Scenario.
REQ_38	1	Model Cards, Data Cards, and ML Lifecycle documentation shall be created and maintained to encapsulate crucial details across the machine learning lifecycle.	Documentation is essential for transparency, trustworthiness, and accountability of the pedestrian detection system.	Regularly update and review the documentation throughout the machine learning lifecycle to ensure accuracy and completeness.
REQ_39	4	The vehicle model must be susceptible to input from CITS/V2X sources and able to process messages (e.g. MCM, CPM, SPAT, MAP).	The primary way to directly exchange information between actors is V2X. This way vehicles can get TM information not perceivable through their own sensors.	One of the other use cases must incorporate V2X information in their models.
REQ_40	4	TM information must be available that can support the vehicle models to improve their behaviour.	To support AVs from the infrastructure/TM side, they need additional information which is not available through their own sensors.	A set of TM information to be processed by the vehicle model.
REQ_41	3 / 4	A specification for the 'desired behaviour' given a specific scenario.	Benchmark the Athena vehicle against the 'norm', the desired behaviour needs to be known.	Report on 'desired behaviour' (also referred to as 'good behaviour' or 'acceptable behaviour').
REQ_42	4	A macroscopic network reflecting the scenario of the corresponding use case or demonstrator used in one of the other use cases.	To simulate the same conditions as a scenario from the other use cases, the road network and relevant infrastructure should be implemented in a microsimulation environment.	A network with the relevant attributes needed to recreate the scenario/use case in a microsimulation environment.

REQ_43	4	Vehicle models that show the desired or 'acceptable' behaviour given the scenario/use case of one of the other use cases.	These vehicles form the benchmark against which the Athena vehicles are evaluated.	Vehicle model(s) for the microsimulation environment that align with the 'acceptable' behaviour report.
REQ_44	4	It must be possible to capture the behaviour of an Athena vehicle from one or more use cases and translate it into specific parameters or functions that when applied to a simulated vehicle result in the same behaviour.	Eventually, in specific scenarios Athena vehicles show a certain behaviour that is likely different than non-AVs. That different behaviour needs to be translated towards microsimulation.	A description of the identified behaviour of an Athena vehicle and the parameters/functions that reflect that behaviour.
REQ_45	4	A vehicle model from a microsimulation environment must reflect the behaviour of an Athena vehicle.	To benchmark the behaviour of Athena vehicles, simulated vehicles in a macroscopic scenario should have the same behaviour/impact as a real vehicle.	Report on adapting a vehicle model to reflect the behaviour of an Athena vehicle and a simulation environment including such vehicles.

3.2. Benchmark scenarios

Table 2: List of scenarios of the 4 Athena use cases.

ID	UC	Description
SCEN_01	2.2	We assess end-to-end latency between two containers in a time-triggered Linux kernel. We compare performance with introduced network interference and competing real-time workloads. Establishing a baseline, we measure latency in a controlled environment. Simulating suboptimal network conditions, we utilize a ping flood workload attempting to disturb critical traffic delivery. Adding real-time tasks competing for network resource we validate the robustness of the time-triggered Linux kernel facing different load scenarios.
SCEN_02	2.2	The prediction module's performance is to be assessed on datasets representative of the urban traffic environment. These datasets, derived both from synthetic (CR, dataset generated by SIE-NL) and real-world (inD, dataset generated by IDI) sources, will include diverse scenarios with varying complexity of road infrastructure (straight roads, three- and four-way intersections, multiple lanes with same and opposite driving directions, etc.), manoeuvres (lane-changing, overtaking, turning, etc.), and varying types of interacting traffic participants (VRUs, vehicles with different dynamic constraints, etc.).

SCEN_03	2.2	The robustness of the prediction module will be evaluated on datasets incorporating cases of upstream failures, such as faulty tracking or faulty annotation of traffic participants or their environment. Should the dataset lack existing failure cases, synthetic cases will be appropriately added to the validation dataset ensuring comprehensive evaluation.
SCEN_04	2.2	Qualitative analysis of the prediction module will consider several challenging traffic scenarios such as those showcasing traffic participants navigating a four-way intersection with or without traffic signalling. These scenarios will be carefully selected so that multiple scenario descriptors affecting their complexity are simultaneously present in the scenarios.
SCEN_05	2.2	Generate real-world datasets based on the requirements of the UC2.2 partners that will be used for training and validating machine learning models. Preliminary work will be performed on how the developed ML-based models in UC2.2 can be validated for their performance in a ViL setup. For this type of scenario, a test vehicle will be prepared.
SCEN_06	2.2	Single and multi-lane urban intersection scenarios involving trucks, cars, motorbikes, cyclists, and pedestrians as traffic participants.
SCEN_07	2.1	Extraction of edge case scenarios from an open large-scale dataset (e.g., nusscenes).
SCEN_08	3	The Ego Vehicle is approaching a junction with a green traffic light (cf. Figure 9). Nevertheless, the vehicle decelerates which unsettles the occupant because it shows non-intuitive behaviour for the occupant. The system will explain to the occupant that it slowed down because it received an information via V2X about an occluded vehicle that violates a red traffic light.
SCEN_09	3	The Ego Vehicle is approaching a roundabout (cf. Figure 10). Since roundabouts are not part of the dataset that was used to train the CNN-based Behaviour Generation module might fail in predicting a reasonable trajectory. Through Competence Observation the lower competence in this situation is measured and provided to the Trajectory Supervision module. In this case the Trajectory Supervision will rely less on the CNN-based Behaviour Generation's prediction and instead initiate an alternative trajectory to safely guide the vehicle around the roundabout.
SCEN_10	1	SCEN_10 illustrates a real-world challenge faced by the Perception System in an urban environment consisting of an urban intersection. The scenario is dynamic, including obstructed views, contradictory sensor data and failures. The scenario should include an occluded pedestrian, who is in the safety critical path, as shown in Figure 4.
SCEN_11	1	Evaluate the cross-modal fusion techniques in the pedestrian detection system by assessing the performance in scenarios where multiple sensor modalities, including LIDAR and camera sensors, contribute to the detection of pedestrians in an urban environment. Vary the environmental conditions, such as lighting and weather, to ensure robustness across diverse settings.
SCEN_12	1	Evaluate the pedestrian detection system's robustness in scenarios where one or more sensors experience failures. Simulate sensor outages and assess the system's ability to adapt and maintain accurate pedestrian detection using the remaining functional sensors.
SCEN_13	1	Simulate scenarios where pedestrians exhibit dynamic movements, including sudden changes in speed, direction, or interactions with other traffic participants. Evaluate the system's ability to accurately detect and track pedestrian trajectories in such dynamic situations.

3.3. Key performance indicators (KPIs)

Table 3: List of key performance indicators of the 4 Athena use cases.

ID	UC	Description	Target value	Contributing to: 1. Fairness, 2. Transparency 3. Accountability, 4. Privacy 5. Accuracy & Robustness (see D1.1)
KPI_01	2.2	Deterministic end-to-end latency between communicating containerized software components executing on the embedded time-triggered runtime	Required end-to-end latency for a computation chain	Accuracy & Robustness
KPI_02	2.2	Quality of independently developed software-components' available timing information	Number of deadlines misses less than 5%	Accuracy & Robustness
KPI_03	2.2	End-to-end latency performance degradation between communicating containerized software components facing computational and network interference	End-to-end latency degradation less than 10%	Accuracy & Robustness
KPI_04	2.2	To evaluate the accuracy of the proposed motion prediction module, following metrics will be calculated: (1) average L2 displacement error (ADE) across prediction horizons of varying length, (2) final L2 displacement error (FDE) for the predicted trajectory state at the end of the horizon, and (3) miss rate (MR) defined as the number of scenarios whose trajectory predictions are not within specified range relative to the ground truth trajectory.	The ADE, FDE, and MR will be determined based on the values achieved with open-source state-of-the-art methods available at validation stage.	Accuracy, Robustness
KPI_05	2.2	To evaluate the feasibility of the predicted trajectory two metrics have been adopted: (1) the number of scenarios for which values of physical measures (e.g., velocity, acceleration, turning rate) are within the threshold specific to the type of traffic participant, and (2) the number of scenarios whose predicted trajectories remain in the traversable area for the duration of the scenario.	Both metrics should target the value of zero infeasible scenarios and scenarios that are not compliant with the provided traffic scene information.	Accuracy, Robustness & Transparency

KPI_06	2.2	Analyse representative datasets based on the requirements described in T2.1 from own measured datasets and the ones generated by partners.	Reflect the quality and comprehensiveness of the dataset analysis that meets the established requirements. However, determining an exact numerical target might depend on various factors: quality of representation, quality of partner data, performance metrics, compliance to the requirements specified in T2.1	
KPI_07	2.2	F1-Score Evaluation of the performance of the SAFEOD algorithm by means of F1-Score.	0.9	Accuracy & Robustness
KPI_08	2.2	Mean average precision (mAP) Evaluation of the performance of the SAFEOD algorithm by means of mAP.	0.7	Accuracy & Robustness
KPI_09	2.2	Intersection over Union (IoU) Evaluation of the performance of the SAFEOD algorithm by means of IoU.	0.7	Accuracy & Robustness
KPI_10	3	Lateral distance to driving lane boundaries	$ d_{left/right} \geq \frac{d_{width}}{2}$	Accuracy & Robustness
KPI_11	3	Time Headway (THW) - The value of the time gap to an object (e.g., a lead vehicle (bumper to bumper) or pedestrian, which is travelling in the vehicle's path of travel).	> 0	Accuracy & Robustness
KPI_12	3	Correlation between the competence measure and the performance of the CNN-based Behaviour Generation module.	There should be a positive correlation.	Accuracy & Robustness, Transparency
KPI_13	3	Competence measure should be sensitive to the presence of the context in the training dataset of the CNN-based Behaviour Generation module.	Less presence should lead to less competence.	Accuracy & Robustness, Transparency
KPI_14	4	Throughput - Total number of vehicles per hour through a particular road section or intersection approach, normalised	Maximum, within constraints.	Transparency Accuracy & Robustness

		to number of lanes and proportion of green time (where relevant).		
KPI_15	4	Average network speed - Average space mean speed of the vehicular fleet on a specific road network.	Maximum, within constraints.	Transparency Accuracy & Robustness
KPI_16	3 / 4	Mean time headway - The mean value of the Time Headway (THW)	Larger = safer. Smaller = more efficient	Transparency
KPI_17	4	Total fuel consumption (l) - Total fuel consumed by all vehicles on the road network during the analysis timeframe.	Less = better.	Transparency
KPI_18	4	Mean of time-to-collision (TTC) - The mean time required for two vehicles (or a vehicle and an object) to collide if they continue at their present speed and on the same path. Measures a longitudinal margin to lead vehicles or objects.	Larger = safer. Smaller = more efficient	Transparency Accountability
KPI_19	2.1	False Positive Rate (FPR) Evaluation of the amount of False Positive scenarios produced by the SLS	0.1	Accuracy & Robustness, Transparency
KPI_20	1	Measure the accuracy of the pedestrian detection system by evaluating the detection rate, false positive rate, and false negative rate.	Achieve a detection accuracy of at least 95%	Transparency, Accuracy &Robustness
KPI_21	1	Evaluate the system's performance in dynamic scenarios, including scenarios with sudden changes in pedestrian movements, unexpected interactions, and complex traffic dynamics.	Successfully handle and adapt to dynamic scenarios with at least 90% accuracy.	Transparency, Accuracy &Robustness
KPI_22	1	Assess the effectiveness of the XAI-Interface in communicating system decisions and conflict resolution to the user.	Usability testing for the XAI-Interface. 60 % of the users should find the XAI interface helpful.	Transparency
KPI_23	1	Evaluate the transparency of the pedestrian detection model through the implementation of explainable layers and model cards.	All deployed AI models must have a model and data cards. Explainable layers should exist for key models.	Transparency
KPI_24	1	Evaluate the latency of the perception system through inference tests on the used hardware (CPU, GPU). Measure the inference speed in FPS.	The FPS of the sensor data processing algorithms should be faster than the sample rate of the slowest perception sensor.	Accuracy & Robustness

4. AI lifecycle in reference to the use case activities

This chapter provides an overview of the various phases of the AI lifecycle and which of these phases are covered by the respective use cases. The AI lifecycle is divided into six phases and refers in part to the following source [4]:

- (Re-)Design
- (Re-)Develop
- (Re-)Deploy
- Operational Use
- Monitor
- Evaluate& Analyse

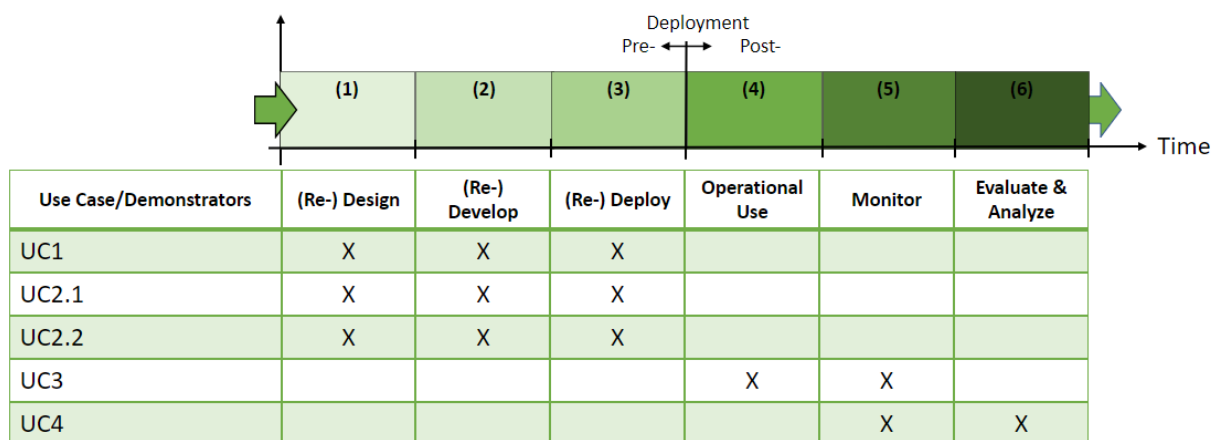


Figure 11: Use case/demonstrator links to the AI lifecycle.

UC1:

(Re-) Design:

Design the architecture of the pedestrian detection system, focusing on the integration of Model-Driven, Data-Driven, and Sensor-Driven methodologies. This involves planning for Explainable Layers (XAI T3.2), Visualization XAI Interface (XAI T3.2), Model Cards (ML Life Cycle MGMT T3.1), Data Cards (ML Life Cycle MGMT T3.1), Reliable Fusion of Sensor Modalities (XAI T3.2), and Explainable Multi-Object Tracking (XAI T3.2). Define the structure and components of the XAI-Interface for visualizing explainable components during runtime.

(Re-) Develop:

Develop the pedestrian detection system by implementing Model-Driven Strategies, including Explainable Layers, Visualization XAI Interface, and Model Cards. Implement Data-Driven Approaches by developing Data Cards. Integrate Sensor-Driven Fusion Techniques, including Reliable Fusion of Sensor Modalities and Explainable Multi-Object Tracking. Conduct thorough testing and evaluation using recorded data, simulated data, or publicly available datasets.

(Re-) Deploy:

Deploy the developed pedestrian detection system in an offline demonstrator using recorded data, simulated data, or publicly available datasets. Integrate the XAI methods into the software stack of the demonstrator vehicle, allowing for real-time execution during the perception system's runtime. Showcase the integration of Model-Driven, Data-Driven, and Sensor-Driven methodologies to ensure reliability, explainability, and transparency in a real-world scenario.

UC2.1:**(Re-) Design:**

Design of the SLS pipeline, including the data preparation process, an ontology that represents the data model, a graphical database, and a rule-based query language. Definition of edge cases tailored to the data model.

Design AI model trained on purely simulated data and assessed collision risk on real-life data provided by IDIADA.

(Re-) Develop:

Development of the ontology for the selected data model, a query manager for scenario mining, and the necessary data pre-processing steps. Evaluation and tests on the entire SLS pipeline and extraction of KPIs.

Develop AI model trained on purely simulated data and assessed collision risk on real life data provided by IDIADA.

(Re-) Deploy:

Deployment of the graphical database and ontology is partially covered in the context of benchmarking and KPI extraction of the SLS pipeline.

UC2.2:**(Re-) Design:**

Design of the targeted SAFEOD approach and related structure based on the specified requirements. Preparation of the necessary virtual simulation data and real-world data sets to be used for training. Design of the motion prediction approach and the acquisition and preparation of the virtual and real-world datasets to be used in model training.

(Re-) Develop:

Development of the targeted SAFEOD algorithm based on virtual simulation data and real-world data sets. Data pre-processing and labelling activities in case needed for the development. Evaluation of the specified metrics and related KPIs.

Development of the motion prediction model, evaluation of the specified primary metrics, and the analysis of the model's explainability. Performing of the necessary data pre-processing on the environment model constructed from training datasets.

(Re-) Deploy:

The deployment argument is based on evaluation results using the specified metrics and KPIs generated by virtual simulation data and data sets. Furthermore, the algorithm will not be deployed on real hardware system and will stay in the simulation domain only.

Testing of the developed motion prediction model will be done in a virtual simulation environment only. However, the evaluation of defined secondary metrics will still be performed.

UC3:**Operational Use:**

The proposed stack for motion planning of automated vehicles allows monitoring of the models' performance during runtime. Thus, specific situations with good or weak system performance can be identified. This information can then be used, for example, to generate specific training data and trigger retraining and subsequent redeployment of the models. It should be noted that the implementation of such an ML-Ops process in the implemented stack is not the focus of this use case or the associated work packages.

Monitor:

The situational awareness module that observes the competence of CNN-based Behaviour Generation module is a system created to monitor the deployed AI system. It monitors the competence of the Behaviour Generation module in the current context to ensure trustworthiness of the module.

Through trajectory supervision, the predictions of ML based trajectory planning are monitored during runtime. The module ensures compliance with driveability constraints and traffic rules. If required, the trajectory can be adjusted before it is passed on to the vehicle actuators. If the modifications are not sufficient to achieve safe driving behaviour, alternative actions can be initiated.

UC4:

Monitor:

Monitor the output of UC1, UC2 and UC3 with respect to what the impact is of the perception, situation awareness and decision-making on the behaviour of the automated vehicle. In addition, monitor what conditions (i.e., (digital) infrastructure, other actors) lead to that behaviour. In case the behaviour deviates from the reference behaviour, try to identify shortcomings/causes for that deviation. Next, behavioural parameters are determined and estimated to capture the automated vehicle characteristics. These parameters are input for the vehicle model which is used in the Evaluate & Analyse phase in macroscopic scenarios.

In the second cycle, the use cases can incorporate traffic management information which could change the deviation between the shown behaviour and the reference behaviour. The steps above can be repeated to monitor the effectiveness of traffic management information.

Identify deviations between the shown behaviour and the reference behaviour, and the evaluation of the behaviour/decision considering human-centric principles (like transparency and understandability) from a traffic management perspective. From insights gained from that evaluation, in addition to relevant knowledge and findings from relevant experiences (e.g., CoEXist, TANGENT), support the definition of suitable reference behaviours for the macroscopic scenarios.

Determine and estimate behavioural parameters to capture the automated vehicle characteristics. These parameters are input for the vehicle model which is used in the Evaluate & Analyse phase in macroscopic scenarios.

Evaluate & Analyse:

Using the parameters from the monitoring phase adapt a vehicle model that reflects the characteristics of the automated vehicle of UC1, UC2 and/or UC3. In addition, if needed, a vehicle model is adapted to reflect the reference behaviour. Implement the infrastructure (i.e., network) for the macroscopic scenario(s) and run the scenario(s) with different mixes of automated and normal vehicles in different LOS. Evaluate the output with respect to the output of the same scenario(s), using only normal vehicles.

In the second cycle, traffic management information could support the automated vehicles, which could impact the deviation between the shown behaviour and the reference behaviour. The automated vehicle model is adapted to capture that impact and the scenario(s) are rerun to evaluate the impact of traffic management information on the network level.

Conclusion

This deliverable is the first in WP5 and thus on the topic of testing and validation. Since the report focuses on the four Althena use cases and is published at an early stage in the project, the use cases were described first. They cover aspects of four main domains where AI is gaining importance in modern vehicles: perception, situation awareness/understanding, decision-making, and traffic management. This deliverable focuses on the content of the use case that is important for the further course of the document. A detailed description, which also takes other aspects of the use cases into account, is planned in Althena deliverable 1.2.

From the descriptions of the use cases, important factors can be derived which are dealt with in chapter 3. There, requirements were defined that must be fulfilled for individual use cases. Some of these requirements are also valid for several use cases, which is partly due to the interaction between them. These requirements, as well as the defined scenarios and key performance indicators, are important for the further activities in this work package.

This deliverable also shows in which phases of the AI lifecycle the individual activities of the use cases are located. The use cases not only cover different phases, but they extend across all aspects of the AI lifecycle.

The validation approaches defined in the deliverable will be applied in the further stages of WP5 and used for the work in Tasks 5.2 to 5.5.

Bibliography

- [1] G. E. Karniadakis, Ioannis G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” 2021
- [2] HCM 2010: Highway Capacity Manual. Washington, D.C.: Transportation Research Board, 2010.
- [3] Donges 1982: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen, Automobil-Industrie 27 (2), 1982
- [4] D. De Silva and D. Alahakoon, “An artificial intelligence life cycle: From conception to production”, vol. 3, no. 6, p. 100489. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666389922000745>
- [5] T. Beemelmans, W. Zahr, L. Eckstein: “Explainable Multi-Camera 3D Object Detection with Transformer-Based Saliency Maps”, Machine Learning for Automated Driving Symposium, 2023

List of acronyms and terms

ADS	Automated Driving System
AI	Artificial Intelligence
ASAM	Association for Standardization of Automation and Measuring Systems
AV	Autonomous Vehicle
C-ITS	Cooperative Intelligent Transport Systems
CCAM	Connected, Cooperative and Automated Mobility
CAM	Cooperative Awareness Message
CEM	Common Evaluation Methodology
DOVA	Distributed ODD attribute Value Awareness
FAME	Framework for coordination of Automated Mobility in Europe
FET	Field Operational Tests
HW	Hardware
KPI	Key Performance Indicator
LIDAR	Light Detection and Ranging
LOS	Level of Service
MCM	Manoeuvre Coordination Message
ML	Machine Learning
MLOps	Machine Learning Operations
MR	Miss Rate
NP	Nondeterministic Polynomial
ODD	Operational Design Domain
PI	Performance Indicator
ROS	Robot Operating System
SAFEOD	Situational Awareness Feature Enhanced Object Detection
SLS	Semantic Labelling System
SPAT	Signal Phase and Timing
SUT	System Under Test
SW	Software
THW	Time Headway
TM	Traffic Management

TTC	Time-to-Collision
UC	Use Case
V2X	Vehicle to Everything
VIL	Vehicle in the Loop
VRU	Vulnerable Road User
WCET	Worst-case Execution Time
XAI	Explainable Artificial Intelligence
XiL	X-in-the-Loop