ENSEMBLE

EUROPEAN COMMISSION

HORIZON 2020
H2020-ART-2016-2017/H2020-ART-2017-Two-Stages
GA No. 769115

ENSEMBLE

ENabling **S**af**E** **M**ulti-**B**rand p**L**atooning for **E**urope

Deliverable No.	D3.3	
Deliverable Title	Generic open-source RCP-level reference implementation of the Tactical Layer	
Dissemination level	Public	
Written By	Dr. A.J.C. Schmeitz, TNO S.I. Yalcinkaya, TNO D.J. van den Brand, TNO	24-07-2019

	Lead Partner as supervisor of the deliverable: Dr. A.J.C. Schmeitz, TNO	24-07-2019
Checked by	WP Leader: D.M.C. Willemsen, TNO	24-07-2019
Approved by	Coordinator: Dr. D.M. Hoedemaeker, TNO	25-07-2019
Status	FINAL, approved by EC	06-08-2022

Please refer to this document as:

Schmeitz, A.J.C., Yalcinkaya, S.I. and Van den Brand, D.J. (2019). *Generic open-source RCP-level reference implementation of the Tactical Layer.* D3.3 of H2020 project ENSEMBLE, (www.platooningensemble.eu)

Disclaimer:

ENSEMBLE is co-funded by the European Commission, DG Research and Innovation, in the HORIZON 2020 Programme. The contents of this publication is the sole responsibility of the project partners involved in the present activity and do not necessarily represent the view of the European Commission and its services nor of any of the other consortium partners.

TABLE OF CONTENTS

Revision history

Version	Date	Author	Summary of changes	Status
0.1	24/07/2019	A.J.C. Schmeitz (TNO)	Initial Version	Prepared
0.2	24/07/2019	D.M.C. Willemsen (TNO)	Report review by WP 3 Leader	Draft
0.3	25/07/2019	D.M. Hoedemaeker (TNO)	Report review by Coordinator	Draft
1.0	25/07/2019	A.J.C. Schmeitz (TNO)	Updated based on review comments	Final

FIGURES

TABLES

1. EXECUTIVE SUMMARY

1.1. Context and need of a multi brand platooning project

Context

Platooning technology has made significant advances in the last decade, but to achieve the next step towards deployment of truck platooning, an integral multi-brand approach is required. Aiming for Europe-wide deployment of platooning, 'multi-brand' solutions are paramount. It is the ambition of ENSEMBLE to realise pre-standards for interoperability between trucks, platoons and logistics solution providers, to speed up actual market pick-up of (sub)system development and implementation and to enable harmonisation of legal frameworks in the member states.

Project scope

The main goal of the ENSEMBLE project is to pave the way for the adoption of multi-brand truck platooning in Europe to improve fuel economy, traffic safety and throughput. This will be demonstrated by driving up to seven differently branded trucks in one (or more) platoon(s) under real world traffic conditions across national borders. During the years, the project goals are:

- Year 1: setting the specifications and developing a reference design with acceptance criteria
- Year 2: implementing this reference design on the OEM own trucks as well as perform impact assessments with several criteria
- Year 3: focus on testing the multi-brand platoons on test tracks and international public roads

The technical results will be evaluated against the initial requirements. Also, the impact on fuel consumption, drivers and other road users will be established. In the end, all activities within the project aim to accelerate the deployment of multi-brand truck platooning in Europe.

1.2. Abstract of this Deliverable

In the context of the ENSEMBLE project, T 3.1 is an umbrella task in WP 3. The objective of WP 3 is the implementation of the requirements and specifications of WP 2 in demonstrator trucks (i.e. comprising hardware and software). This implementation includes the operational and the tactical layer, as well as the interface to the strategic layer. WP 3 focusses on the implementation of platooning level A, which means that platooning will be implemented as a driver support function. The automation consists of longitudinal automation with optional lateral support. The aim of T 3.1 is the development and prototyping of a reference tactical layer and V2X design, which compromise the common multi-brand functionality, according to the specifications of WP 2.

Deliverable 3.3 (D3.3) deals with the rapid control prototyping (RCP) level implementation of the unbranded Tactical Layer. The prototype serves as a basis for each OEM to develop its own brand-specific implementation in tasks 3.2–3.7, and also serves as a benchmark against which the brand-specific implementations can be evaluated. The implementation will not be automotive-grade hard- and software, but will be otherwise fully functional in a simulation environment.

The Tactical Layer consists of Tactical Layer software modules, i.e. the Platoon Coordinator, and the operational (non-automotive grade) ITS-G5 V2X communication. The detailed design of the Platoon Coordinator and the implementation of it in Simulink (open code) are provided in D3.1 (Schmeitz, 2019). The operational ITS-G5 V2X communication is the subject of D3.2 (de Jongh, 2019). The communication consists of a hardware communication unit and the (embedded) software to exchange messages according to the ENSEMBLE communication protocol, as specified in WP 2.

For rapid control prototyping purposes of the Tactical Layer, the 'full' white-label truck must be simulated and, in order to check the functioning of the developed ITS-G5 V2X communication, the (hardware) communication unit must be included in this simulation. Consequently, a Hardware-in-the-loop (HIL) simulation setup is created. In the HIL simulation, not only the communication unit hardware is 'hardware-in-the-loop', but also the Platoon Coordinator must work as a node in the network (to be run independent of Simulink), meaning that standalone C++ code has to be generated from the Simulink model of the Platoon Coordinator. In this way, the functioning of the Platoon Coordinator as standalone code can be tested in simulation on a dedicated hardware, in a network and with real input/output devices.

In order to simulate the 'full' white-label truck, next to the Tactical Layer also the Operational Layer must be simulated. This means that all Operational Modules (e.g. Longitudinal Control, HMI, communication, sensors) and the hardware components such as actuators (e.g. engine, brakes), sensors (e.g. radar, camera) and HMI (e.g. buttons) must be simulated. Consequently, models of these components have been developed and included in the HIL environment. Further, in order to simulate platooning use cases, two or more white-label trucks are required that can drive in a virtual environment containing e.g. a road or other road users.

To establish the HIL simulation platform for ENSEMBLE, it was decided to use the Robot Operating System (ROS) ecosystem, as prior experience was available using ROS. ROS is a middleware for use in complex robot control software systems and it is widely used in prototyping of automated driving systems. Next to ROS's core functionality a variety of development tools is available in the ROS ecosystem, e.g. for simulation of systems, visualisation and data recording.

D3.3 is a demonstrator type of deliverable, meaning that it consists of the demonstration that the Tactical Layer is implemented and functioning in the HIL simulation. In this context, this deliverable report is aimed to be brief description of the setup and its usage.

Finally, the HIL simulation platform is a tool for developing and testing of the Tactical Layer in ENSEMBLE. In the future, the tool will be further used in the project for testing and improvement of the Tactical Layer, as it is likely that new insights obtained during implementation and testing in WP 3 and/or WP 5 will lead to future updates of the Tactical Layer, which would require updates of the HIL as well, e.g. interfacing, simulation of specific scenarios, etc.

2. INTRODUCTION

2.1. Background

In the context of the ENSEMBLE project, T 3.1 is an umbrella task in WP 3. The objective of WP 3 is the implementation of the requirements and specifications of WP 2 in demonstrator trucks (i.e. comprising hardware and software). This implementation includes the operational and the tactical layer, as well as the interface to the strategic layer. WP 3 focusses on the implementation of platooning level A, which means that platooning will be implemented as a driver support function. The automation consists of longitudinal automation with optional lateral support. The aim of T 3.1 is the development and prototyping of a reference tactical layer and V2X design, which compromise the common multi-brand functionality, according to the specifications of WP 2.

Deliverable 3.3 (D3.3) deals with the rapid control prototyping (RCP) level implementation of the unbranded Tactical Layer. The prototype serves as a basis for each OEM to develop its own brand-specific implementation in tasks 3.2–3.7, and also serves as a benchmark against which the brand-specific implementations can be evaluated. The implementation will not be automotive-grade hard- and software, but will be otherwise fully functional in a simulation environment.

The Tactical Layer consists of Tactical Layer software modules, i.e. the Platoon Coordinator, and the operational (non-automotive grade) ITS-G5 V2X communication. The detailed design of the Platoon Coordinator and the implementation of it in Simulink (open code) are provided in D3.1 (Schmeitz, 2019). The operational ITS-G5 V2X communication is the subject of D3.2 (de Jongh, 2019). The communication consists of a hardware communication unit and the (embedded) software to exchange messages according to the ENSEMBLE communication protocol, as specified in WP 2.

2.2. Aim

Demonstrate that the Tactical Layer, consisting of the Tactical Layer software modules (Platoon Coordinator), as described in D3.1, and the reference communication unit including the ENSEMBLE protocol, as described in D3.2, are functional in a rapid control prototyping environment, a.k.a. Hardware-in-the-loop (HIL) simulation, representing platooning white-label trucks.

2.3. Structure of the report

Chapter 3 provides a description of the Hardware-in-the-loop (HIL) simulation environment. Chapter 4 gives an overview of the implementation. A brief summary of the report and conclusions are provided in Chapter 5. Finally, a glossary of used terms is provided in Appendix A.

3. RCP-LEVEL IMPLEMENTATION OF THE WHITE LABEL TRUCK

This section describes how the white label truck is implemented in TNO’s HIL setup. First, the implemented functionalities of the HIL setup are explained. Then, the hardware deployment of these functionalities is described, and finally the software design principles are discussed.

3.1. Functionalities

HIL solutions are extensively used for integration and validation purposes and have been applied already in the past, e.g. for developing the platooning software of the EcoTwin trucks that participated in the European Truck Platooning Challenge 2016. As Platooning software, communication units, protocols, use cases, etc. change with different system requirements and specifications, an ENSEMBLE specific HIL setup is created that enables the simulation of the white label truck. Nevertheless, existing HIL functionalities are reused when appropriate. Table 1 gives an overview of the main HIL functionalities.

Functionality	Description
Simulation Environment	This functionality creates a simulation world and provides road geometry and weather condition for simulated sensors. Also, it provides visualization of the world to the HIL users. The simulation environment is shared among simulated vehicles.
Simulated Radar	This functionality simulates a radar sensor. It detects objects in the simulation environment.
Simulated Camera	This functionality simulates the front camera. It detects pavement markings (lines) and objects in the simulation environment.
Simulated Lidar	This functionality simulates a lidar. It detects objects in the simulation environment. (currently not yet used for Platoon Level A in ENSEMBLE)
Simulated Vehicle Dynamics	This functionality simulates the vehicle motion.

Simulated Low-Level Controllers	This functionality simulates the behaviour and capabilities of the low-level control systems for driving, braking and steering.
Lateral Control	This functionality controls the lateral position and heading of the vehicle and sends steering setpoints to the (simulated) low level steering controller.
Longitudinal Control	This functionality controls the speed and headway time of the host vehicle and sends acceleration setpoints to the (simulated) low level longitudinal controller.
Platoon Manoeuvre Coordination	This functionality handles the sequences of the interaction protocol to conduct the manoeuvres to join, leave or split the platoon.
Platoon Status & Platoon Vehicle Property Collection & Sharing	This functionality shares information with all vehicles in the platoon.
ITS-G5 communication	This functionality enables vehicle to vehicle (V2V) communication so that platooning use cases can be integrated and validated on the HIL setup.
Sensor Fusion & Object Tracker	This function fuses object information from the different sensors and outputs a list of tracked objects.
Road Model	This function interprets the camera output (lines) and generates a road model (centre line geometry) that acts as input to the motion controller.
Host Tracking	This function estimates the host vehicle state: localization, velocity, acceleration, yaw rate, etc.
3D & 2D Visualization	This functionality visualizes the simulated trucks and other information, such as sensor data and the output of the sensor fusion.
Human Machine Interface (HMI)	This is the web interface with which a user can interact to control the vehicle functions such as enabling/disabling ACC, Platooning, Automated Lane Keeping, leaving a platoon.

Table 1: Overview of HIL functionalities.

3.2. Hardware Deployment

The functionalities listed in the previous section are used to implement truck platooning HIL simulations in the project. In this section, the hardware deployment of the functionalities is described for a two-truck platooning HIL simulation.

One of the simulated white-label trucks runs on a ROS powered Linux desktop PC, and the other one runs on a hybrid solution of a ROS powered Linux desktop PC and optionally a real-time prototyping PC. For testing the reference design, the real-time controller PC is not (yet) included, as the choice of specific real-time hardware is OEM specific. For demonstrating the deployment possibilities of the reference design it suffices to show that the reference design can run as a standalone component in the network (ROS node in the HIL setup). A third Linux desktop PC is used as Simulation PC with high GPU and CPU power so that it can execute the simulation world, sensor models, and vehicle models. The diagram in Figure 1 illustrates the hardware deployment of the functionalities and their relations.

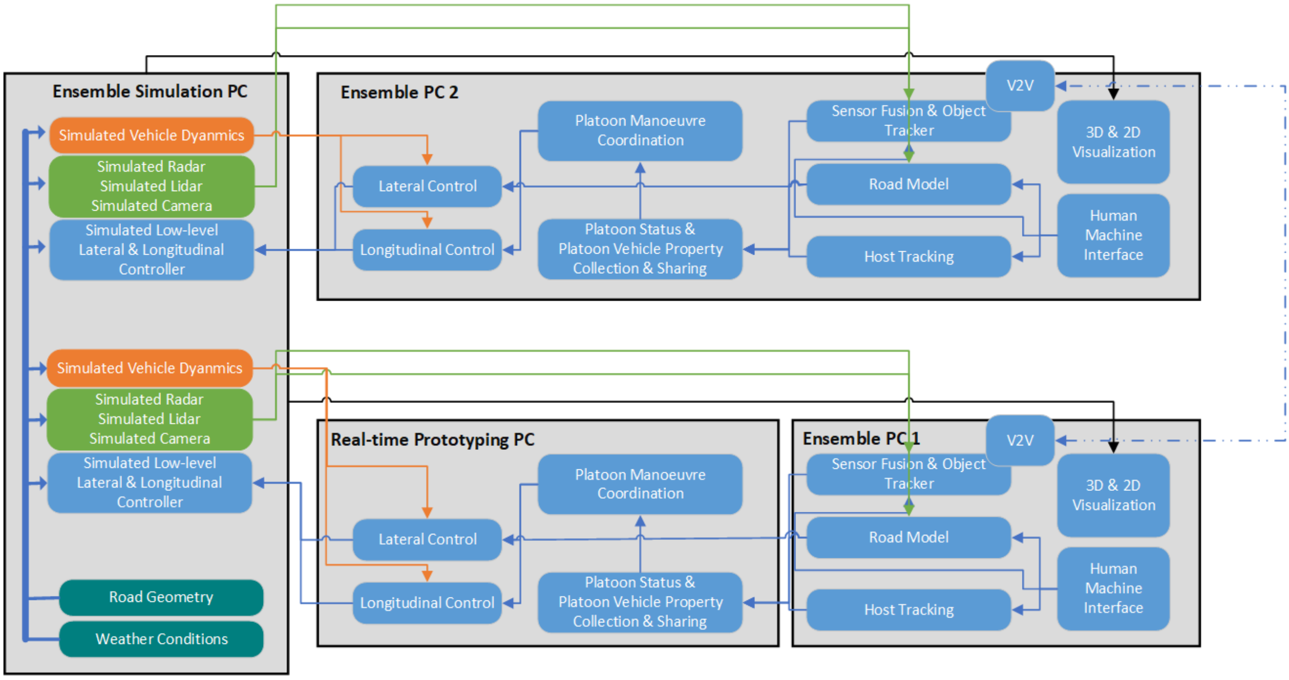

Figure 1: Hardware deployment of the functionalities in the HIL setup.

In the HIL setup, logical and physical deployment of software components and functionalities are kept parallel to increase the fidelity and abstraction of the setup. In Figure 2 the diagram of physical components of the current HIL setup is given with wiring information.

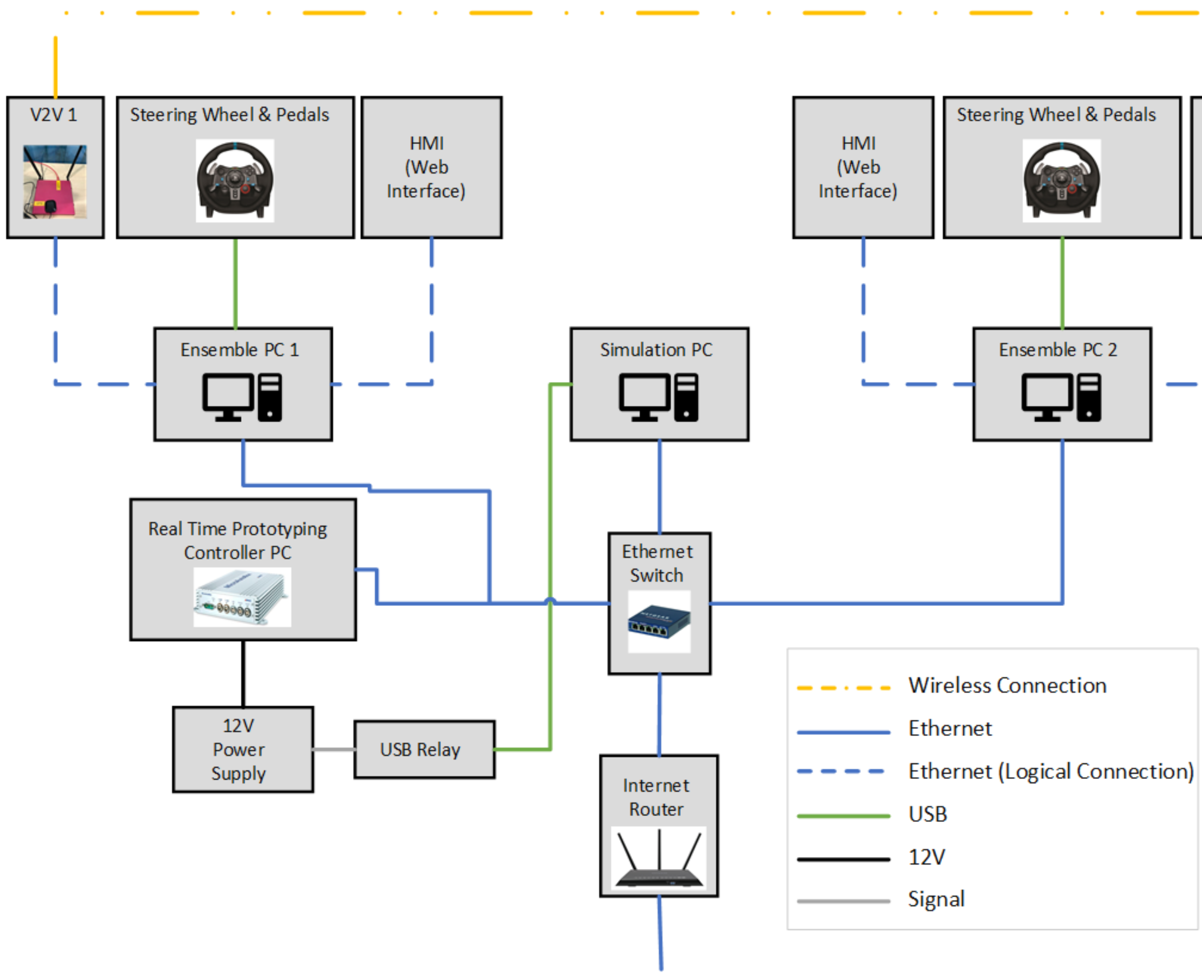

Figure 2: Physical components of the HIL setup.

Design decisions of the HIL setup are also determinative on HIL hardware deployment. The HIL hardware layout can be better explained based on these design decisions:

- **Deployable:** HIL hardware is designed in a way that software and functionality can be easily ported to real vehicles if desired. In the scope of ENSEMBLE, the Tactical Layer software modules and the V2V communication solution are integrated in the HIL. This integration effort helps to mature the interfaces of these functionalities such that these can be deployed to real vehicles without much additional effort.

- **Modularity:** Vehicle specific and simulation specific software are running on distinct individual PCs or PC suites. For instance, Ensemble PC2 is used to run the first vehicle's software, while on the other hand Ensemble PC1 and optionally a real-time prototyping controller PC are used to run software of the second vehicle. Modular design of the HIL enables combining different components in a flexible way to create different solutions for different needs.

- **Real-time:** The architecture of the HIL foresees in the possibility to include a real-time controller PC to test time critical functionality of the software on a real-time prototyping controller PC. This PC is managed by a real-time operating system so that time critical functionality, namely platooning, lateral and longitudinal controllers are tested and integrated on a system comparable to that in real prototype vehicles.

- **Network concerns:** The HIL setup contains two network channels. One is the simulation network and the other is the wireless ITS-G5 V2V network. All simulation related communication is handled by wired communication and only the simulation data is exchanged between vehicles and the simulation master. There is no V2V communication allowed on the simulation channel. All V2V communication happens via the wireless communication units, i.e. TNO's APU-based on-board units (OBUs) as described in D3.2 (de Jongh, 2019). As a result, APU integration is performed as in the real vehicles.

- **Extensibility:** The HIL setup is extensible for both functionalities and number of simulated vehicles thanks to the modular design and implementation. A new functionality can be added with only providing the necessary software interface, even the deployed hardware is not decisive (see software design section). In order to add another vehicle in the simulation, only another individual PC suite shall be connected to the simulation world.

3.3. Software Design

The HIL simulation software and the vehicle functionalities are all designed in a loosely coupled communication manner. The HIL design is realized by using ROS (Robot Operating System). In this section the software design is explained with ROS concepts to expose the details of design implementation.

ROS is a meta operating system which provides hardware abstraction, message sharing, package management and commonly used libraries and tools. ROS runs on an operating system, and provides a communication infrastructure (network) for processes (nodes). Currently, Ubuntu 16.04 is used as the operating system on the HIL PCs, except for the optional real-time prototyping controller PC. The real-time part of the system communicates with the ROS network via TCP/IP.

The HIL functionalities are designed as **ROS nodes**. Each node is a standalone process where either a single node or multiple nodes are used to realise a HIL function. The ROS client library is used to wrap the functionality implementation. This library provides utilities to interact with the ROS network. There are two implementations of the client library, roscpp (C++14) and rospy (Python 2.7). The **ROS package manager** eases the creation of nodes especially for C++ and Python projects.

Node connections are organized by a *master*. **ROS master** is a node which provides a lookup utility to other nodes. In the HIL setup, ROS master runs on the Simulation PC together with the simulation program, Gazebo.

Gazebo is an open source 3D simulation software which is well integrated with ROS. In the HIL, Gazebo is executed as two nodes, Gazebo Server and Gazebo Client. Gazebo Server is responsible for simulating physics, weather condition, road network, vehicles and sensor models. Gazebo Client is responsible for visualisation of the simulated world.

Gazebo nodes and all other nodes communicate with each other through **ROS messages**. There are many pre-defined ROS messages, and also custom messages can be defined by using other message types.

A node can receive and send messages to other nodes via **ROS topics**. A node can publish messages to multiple topics and subscribe to multiple topics to receive messages. A node can publish and subscribe to a topic without having a counterpart topic. This allows a loosely coupled communication network design.

Published topics are basically send in a many-to-many manner, however topics are not suitable for peer-to-peer request-answer messaging requirements. **ROS Services** are used for peer-to-peer communication needs.

The diagram in Figure 3 is a simple illustration of these basic ROS concepts (ROS wiki, 2019).

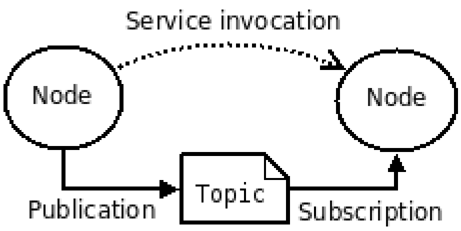

Figure 3: Diagram of basic ROS concepts.

The HIL software is designed and implemented on ROS concepts. ROS allows flexible system design and implementation with provided functionalities, especially with a loosely coupled communication infrastructure.

Finally, a possible disadvantage of ROS is that in itself, it is not a real-time operating system, but it gives sufficient extensibility and tools to communicate with real-time systems.

4. IMPLEMENTATION OVERVIEW

The HIL setup is implemented on 3 PCs, as is shown in Figure 4: the ENSEMBLE simulation PC and 2 separate PCs running the vehicle software for currently 2 trucks (i.e. PC1 and PC2). The main purpose of the simulator core is to run the ROS master and Gazebo simulator software, which orchestrate the communication between the simulator and vehicles PCs. Each vehicle PC separately runs the vehicle software.

Figure 4: HIL setup.

The simulation PC is also used to get a bird's eye overview of the virtual world, see Figure 5. This interface is used to leverage the flexibility of the simulation and enables the user to move around the trucks or close up on details during a simulation. For the sake of simplicity, the world which is created is kept to a minimal. It consists of a large completely flat grassland with a road on it. As depicted in Figure 6, the current road geometry is an oval track, which is of similar shape as the High-speed circuit of IDIADA. On this road, 2 detailed models of the trucks are rendered, including the steering of the wheels and exact positions of the sensors on these trucks. Using sensor models and these sensor positions, the sensors are simulated. In Figure 5 the field of view of the radar sensors is visualised in blue.

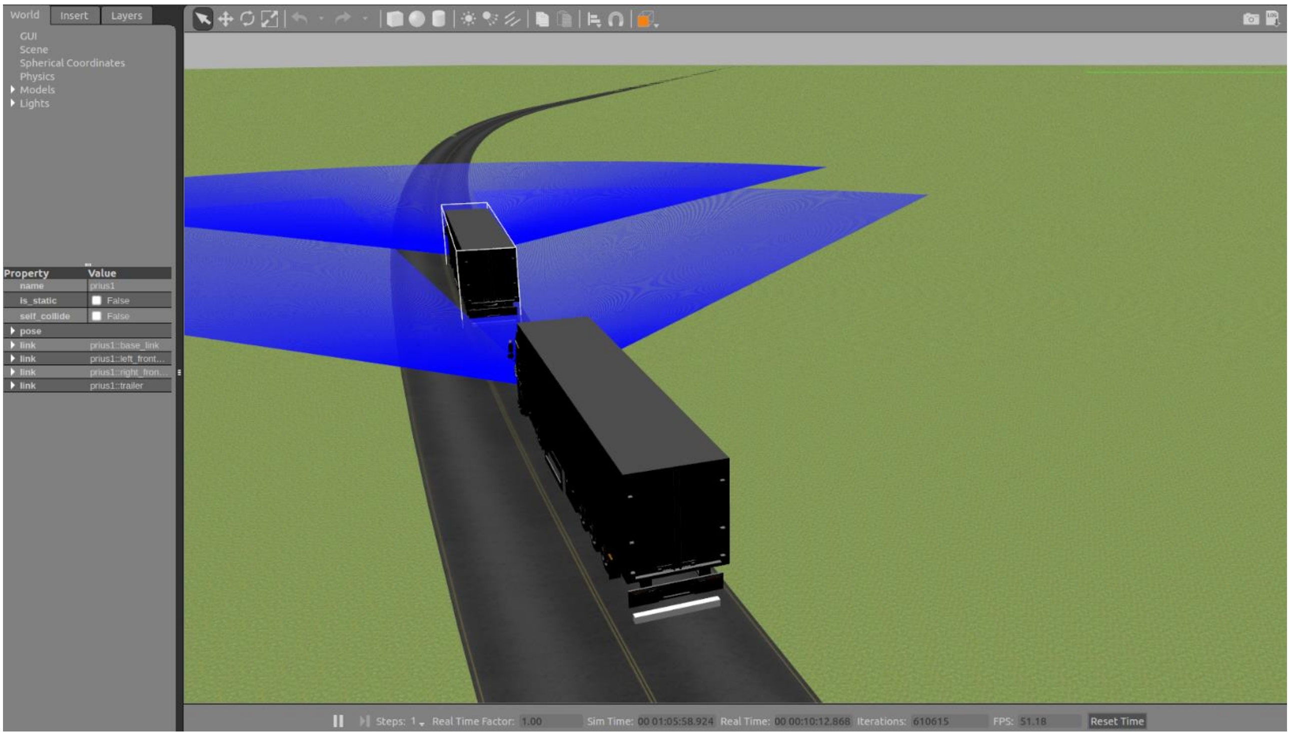

Figure 5: Simulation PC view with two trucks and the radar sensor range (in blue).

Figure 6: Shape of the road.

The vehicle PCs (PC1 and PC2) are used to mimic the white-label trucks and monitor the behaviour of the system from a single truck's perspective. To manually control the driving of a truck a steering wheel and pedals for throttle and braking are available. The steering wheel is also equipped with force feedback, allowing the steering wheel moment to be sensed while steering.

On the monitors of PC1 and PC2, different views can be opened. The most commonly used view is RViz. RViz consists of 3 panels, as can be seen in Figure 7. Firstly, the truck camera panel contains a view of the road from the driver’s perspective. When driving the trucks through the simulation this view is needed to be able to manoeuvre the trucks. Secondly, the control panel can be used to configure RViz to turn on/off different type of visualizations. Finally, the visualization panel is used to show certain output to evaluate the Platooning software under test.

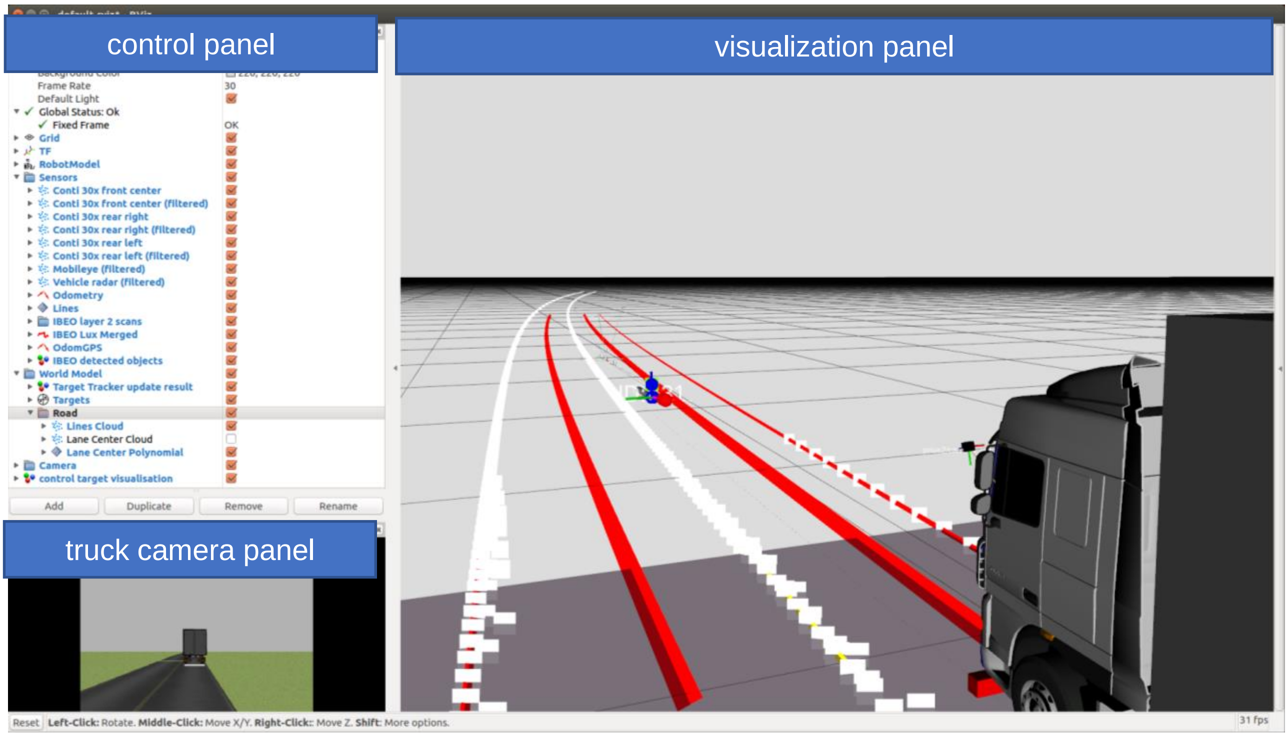

Figure 7: RViz screen on PC2.

In particular the visualization panel is helpful when debugging or identifying a problem. In this panel all positions and speeds used in internal interfaces can be plotted in the simulated world. An example is shown in Figure 8, where the following information is visualised:

- Detected lane markings;
- Identified targets with their speed, uncertainty and sensor source (radar detection location is shown, i.e. back of trailer);
- Platoonable targets with their ID;
- Reference coordinate system of the ego truck;
- Reference coordinate system of the trucks received via V2V communication.

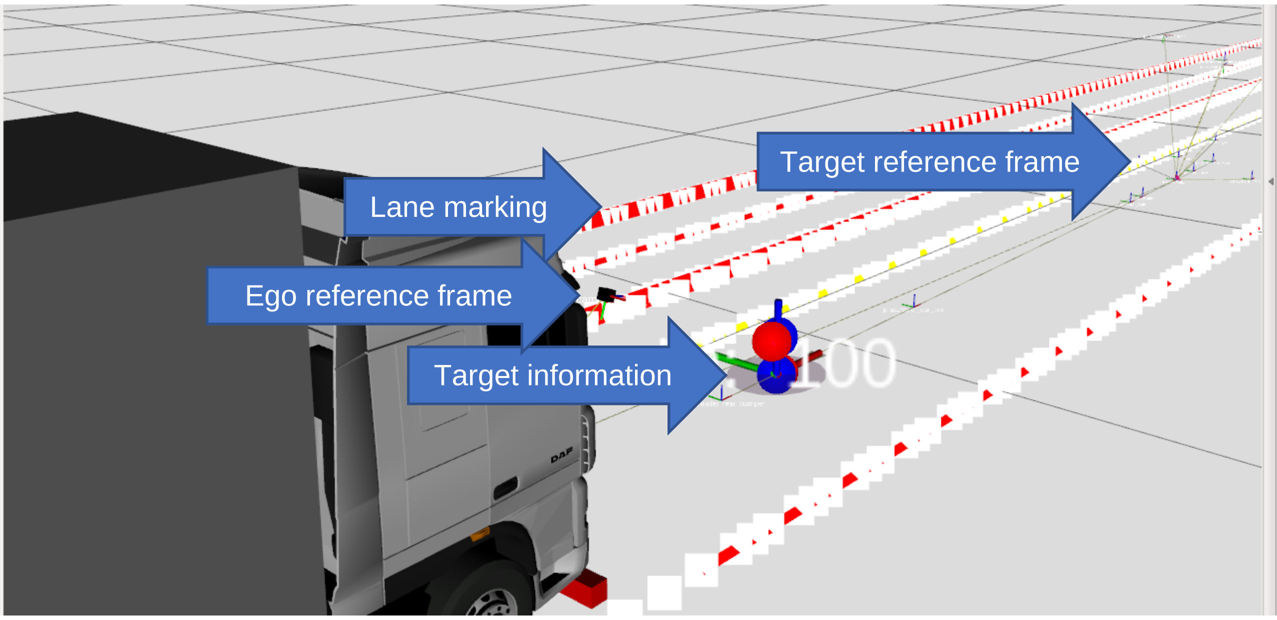

Figure 8: RViz visualization panel.

Next to the simulator PCs, also a small infrastructure network exists, which is used to connect all PCs and hardware together over ethernet. The ITS-G5 communication units, i.e. TNO's APU-based on-board units (OBUs) as described in D3.2 (de Jongh, 2019), are also attached. This is shown in Figure 9. All software under test is configured in such a way that communication with another truck's software can only be done via the ITS-G5 communication units and not directly via ethernet.

In addition to the HIL setup a small ether listening setup is used to read signals being sent in the room, see Figure 10. This setup consists of a third communication unit connected to a laptop. Using Wireshark, every signal which this units receives can be displayed. The signals are not decoded, but are only used to quickly identify if the communication units are transmitting or not.

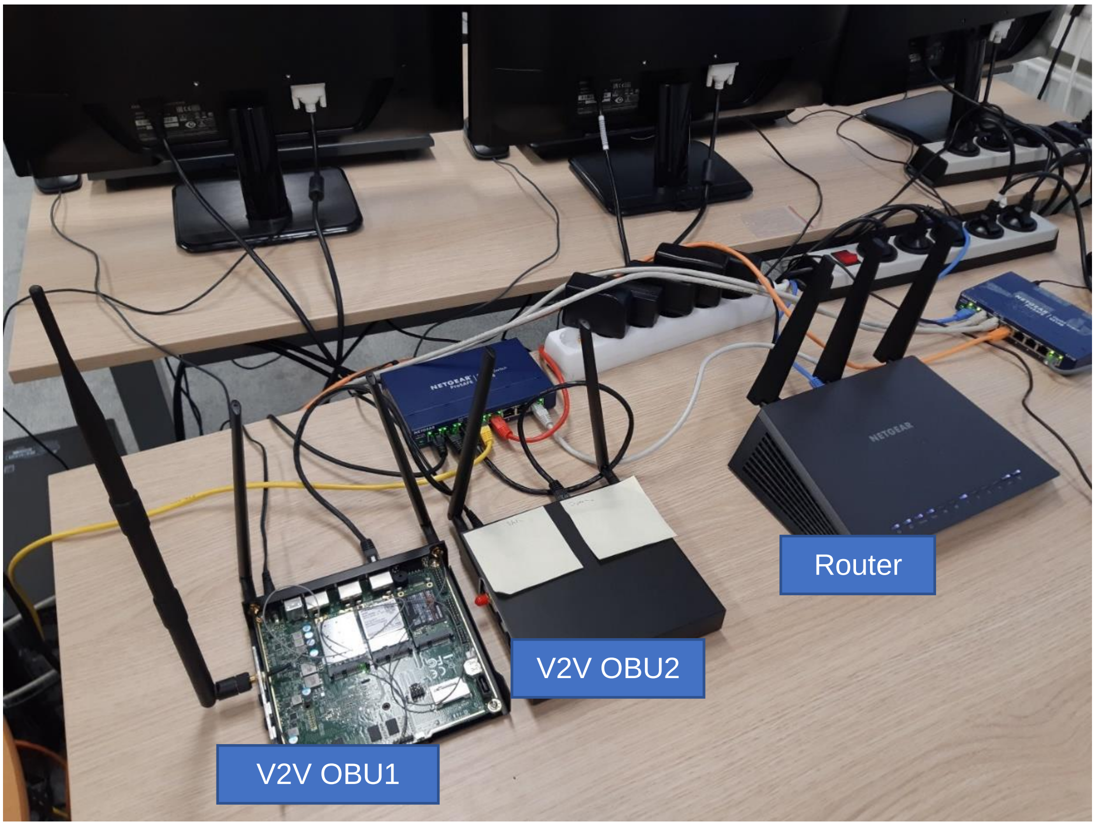

Figure 9: Communication hardware of the HIL.

Figure 10: HIL ether listener setup.

5. SUMMARY AND CONCLUSION

The aim of D3.3. is to demonstrate that the Tactical Layer, consisting of the Tactical Layer software modules (Platoon Coordinator), as described in D3.1, and the reference communication unit including the ENSEMBLE protocol, as described in D3.2, are functional in a rapid control prototyping environment, a.k.a. Hardware-in-the-loop (HIL) simulation, representing platooning white-label trucks.

For rapid control prototyping purposes of the Tactical Layer, platooning 'full' white-label trucks are simulated using a HIL simulation platform. In order to check the functioning of the developed ITS-G5 V2X communication, the (hardware) communication unit is included in the HIL simulation. In the HIL environment, the Platoon Coordinator functions as a standalone node in the network, meaning that the Platoon Coordinator runs independent of Simulink in which it was made. To achieve this standalone C++ code is generated from the Simulink model of the Platoon Coordinator. In this way, the functioning of the Platoon Coordinator as standalone code is tested in simulation on a dedicated hardware, in a network and with real input/output devices.

In order to simulate the 'full' white-label truck, next to the Tactical Layer also the Operational Layer is simulated. This means that all Operational Modules (e.g. Longitudinal Control, HMI, communication, sensors) and the hardware components such as actuators (e.g. engine, brakes), sensors (e.g. radar, camera) and HMI (e.g. buttons) are simulated. Further, in order to simulate platooning use cases, multiple white-label trucks are simulated in a virtual environment containing e.g. a road or other road users. Finally, the HIL simulation platform for ENSEMBLE is established using the Robot Operating System (ROS) ecosystem, which offers besides ROS's core functionality as middleware, a variety of development tools, e.g. for simulation of systems, visualisation and data recording.

D3.3 is a demonstrator type of deliverable, meaning that it consists of the demonstration that the Tactical Layer is implemented and functioning in the HIL simulation. In this context, this deliverable report is aimed to be a brief description of the setup and its usage.

Finally, the HIL simulation platform is a tool for developing and testing of the Tactical Layer in ENSEMBLE. In the future, the tool will be further used in the project for testing and improvement of the Tactical Layer, as it is likely that new insights obtained during implementation and testing in WP 3 and/or WP 5 will lead to future updates of the Tactical Layer, which would require updates of the HIL as well, e.g. interfacing, simulation of specific scenarios, etc.

.

6. BIBLIOGRAPHY

de Jongh, J. and van de Sluis, H.J.D. (2019). *Operational (non-automotive grade) ITS-G5 communication, supporting the multi-brand truck*. D3.2 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

ROS wiki (2019), http://wiki.ros.org/ROS/Concepts.

Schmeitz, A.J.C., Verhaeg, G.J.A., Fusco, M. (2019). *Detailed design of the unbranded Tactical Layer.* D3.1 of H2020 project ENSEMBLE, (www.platooningensemble.eu).

APPENDIX A

Definitions

Term	Definition
Convoy	A truck platoon may be defined as trucks that travel together in convoy formation at a fixed gap distance typically less than 1 second apart up to 0.3 seconds. The vehicles closely follow each other using wireless vehicle-to-vehicle (V2V) communication and advanced driver assistance systems
Cut-in	A lane change manoeuvre performed by vehicles from the adjacent lane to the ego vehicle's lane, at a distance close enough (i.e., shorter than desired inter vehicle distance) relative to the ego vehicle.
Cut-out	A lane change manoeuvre performed by vehicles from the ego lane to the adjacent lane.
Cut-through	A lane change manoeuvre performed by vehicles from the adjacent lane (e.g. left lane) to ego vehicle's lane, followed by a lane change manoeuvre to the other adjacent lane (e.g. right lane).
Ego Vehicle	The vehicle from which the perspective is considered.
Emergency brake	Brake action with an acceleration of <-4 m/s2
Event	An event marks the time instant at which a transition of a state occurs, such that before and after an event, the system is in a different mode.
Fail-safe	A fail-safe in engineering is a design feature or practice that in the event of a specific type of failure, inherently responds in a way that will cause no or minimal harm to other equipment, the environment or to people.
Following truck	Each truck that is following behind a member of the platoon, being every truck except the leading and the trailing truck, when the system is in platoon mode.
Leading truck	The first truck of a truck platoon
Legal Safe Gap	Minimum allowed elapsed time/distance to be maintained by a standalone truck while driving according to Member States regulation (it could be 2 seconds, 50 meters or not present)
Manoeuvre ("activity")	A particular (dynamic) behaviour which a system can perform (from a driver or other road user perspective) and that is different from standing still, is being considered a manoeuvre.

Term	Definition
ODD (operational design domain)	The ODD should describe the specific conditions under which a given automation function is intended to function. The ODD is the definition of where (such as what roadway types and speeds) and when (under what conditions, such as day/night, weather limits, etc.) an automation function is designed to operate.
Operational layer	The operational layer involves the vehicle actuator control (e.g. accelerating/braking, steering), the execution of the aforementioned manoeuvres, and the control of the individual vehicles in the platoon to automatically perform the platooning task. Here, the main control task is to regulate the inter-vehicle distance or velocity and, depending on the Platooning Level, the lateral position relative to the lane or to the preceding vehicle. Key performance requirements for this layer are vehicle following behaviour and (longitudinal and lateral) string stability of the platoon, where the latter is a necessary requirement to achieve a stable traffic flow and to achieve scalability with respect to platoon length, and the short-range wireless inter-vehicle communication is the key enabling technology.
Platoon	A group of two or more automated cooperative vehicles in line, maintaining a close distance, typically such a distance to reduce fuel consumption by air drag, to increase traffic safety by use of additional ADAS-technology, and to improve traffic throughput because vehicles are driving closer together and take up less space on the road.
Platoon Automation Levels	In analogy with the SAE automation levels subsequent platoon automation levels will incorporate an increasing set of automation functionalities, up to and including full vehicle automation in a multi-brand platoon in real traffic for the highest Platooning Automation Level. The definition of “platooning levels of automation” will comprise elements like e.g. the minimum time gap between the vehicles, whether there is lateral automation available, driving speed range, operational areas like motorways, etc. Three different levels are anticipated; called A, B and C.
Platoon candidate	A truck who intends to engage the platoon either from the front or the back of the platoon.
Platoon cohesion	Platoon cohesion refers to how well the members of the platoon remain within steady state conditions in various scenario conditions (e.g. slopes, speed changes).
Platoon disengaging	The ego-vehicle decides to disengage from the platoon itself or is requested by another member of the platoon to do so. When conditions are met the ego-vehicle starts to increase the gap between the trucks to a safe non-platooning gap. The disengaging is completed when the gap

Term	Definition
	is large enough (e.g. time gap of 1.5 seconds, which is depends on the operational safety based on vehicle dynamics and human reaction times is given). A.k.a. leave platoon
Platoon dissolve	All trucks are disengaging the platoon at the same time. A.k.a. decoupling, a.k.a. disassemble.
Platoon engaging	Using wireless communication (V2V), the Platoon Candidate sends an engaging request. When conditions are met the system starts to decrease the time gap between the trucks to the platooning time gap. A.k.a. join platoon
Platoon formation	Platoon formation is the process before platoon engaging in which it is determined if and in what format (e.g. composition) trucks can/should become part of a new / existing platoon. Platoon formation can be done on the fly, scheduled or a mixture of both. Platoon candidates may receive instructions during platoon formation (e.g. to adapt their velocity, to park at a certain location) to allow the start of the engaging procedure of the platoon.
Platoon split	The platoon is split in 2 new platoons who themselves continue as standalone entities.
Requirements	Description of system properties. Details of how the requirements shall be implemented at system level
Scenario	A scenario is a quantitative description of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. From the perspective of the ego vehicle, a scenario contains all relevant events. Scenario is a combination of a manoeuvre ("activity"), ODD and events
Service layer	The service layer represents the platform on which logistical operations and new initiatives can operate.
Specifications	A group of two or more vehicles driving together in the same direction, not necessarily at short inter-vehicle distances and not necessarily using advanced driver assistance systems
Steady state	In systems theory, a system or a process is in a steady state if the variables (called state variables) which define the behaviour of the system or the process are unchanging in time. In the context of platooning this means that the relative velocity and gap between trucks is unchanging within tolerances from the system parameters.

Term	Definition
Strategic layer	The strategic layer is responsible for the high-level decision-making regarding the scheduling of platoons based on vehicle compatibility and Platooning Level, optimisation with respect to fuel consumption, travel times, destination, and impact on highway traffic flow and infrastructure, employing cooperative ITS cloud-based solutions. In addition, the routing of vehicles to allow for platoon forming is included in this layer. The strategic layer is implemented in a centralised fashion in so-called traffic control centres. Long-range wireless communication by existing cellular technology is used between a traffic control centre and vehicles/platoons and their drivers.
Tactical layer	The tactical layer coordinates the actual platoon forming (both from the tail of the platoon and through merging in the platoon) and platoon dissolution. In addition, this layer ensures platoon cohesion on hilly roads, and sets the desired platoon velocity, inter-vehicle distances (e.g. to prevent damaging bridges) and lateral offsets to mitigate road wear. This is implemented through the execution of an interaction protocol using the short-range wireless inter-vehicle communication (i.e. V2X). In fact, the interaction protocol is implemented by message sequences, initiating the manoeuvres that are necessary to form a platoon, to merge into it, or to dissolve it, also taking into account scheduling requirements due to vehicle compatibility.
Target Time Gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds, agreed by all the Platoon members; it represents the minimum distance in seconds allowed inside the Platoon.
Time gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds.
Trailing truck	The last truck of a truck platoon
Truck Platoon	Description of system properties. Details of how the requirements shall be implemented at system level
Use case	Use-cases describe how a system shall respond under various conditions to interactions from the user of the system or surroundings, e.g. other traffic participants or road conditions. The user is called actor on the system, and is often but not always a human being. In addition, the use-case describes the response of the system towards other traffic participants or environmental conditions. The use-cases are described as a sequence of actions, and the system shall behave according to the specified use-cases. The use-case often represents a desired behaviour or outcome. In the ensemble context a use case is an extension of scenario which add more information regarding specific internal system interactions, specific interactions with the actors (e.g. driver, I2V) and will add different flows (normal &

Term	Definition
	alternative e.g. successful and failed in relation to activation of the system / system elements).

Acronyms and abbreviations

Acronym / Abbreviation	Meaning
ACC	Adaptive Cruise Control
ABS	Anti-lock Braking System
ACSF	Automatically Commanded Steering Function
ADAS	Advanced driver assistance system
ADR	Agreement concerning the International Carriage of Dangerous Goods by Road
AEB	Autonomous Emergency Braking (System, AEBS)
ASIL	Automotive Safety Integrity Level
ASN.1	Abstract Syntax Notation One
BTP	Basic Transport Protocol
C-ACC	Cooperative Adaptive Cruise Control
C-ITS	Cooperative ITS
CA	Cooperative Awareness
CAD	Connected Automated Driving
CAM	Cooperative Awareness Message
CCH	Control Channel
CPU	Central Processing Unit
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
DITL	Driver-In-the-Loop
DOOTL	Driver-Out-Of-the Loop
DSRC	Dedicated Short-Range Communications

Acronym / Abbreviation	Meaning
ESF	Emergency steering function
ESP	Electronic Stability Program
ETSI	European Telecommunications Standards Institute
EU	European Union
FAD	Fully Automated Driving
FCW	Forward Collision Warning
FLC	Forward Looking Camera
FSC	Functional Safety Concept
GN	GeoNetworking
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRVA	Working Party on Automated/Autonomous and Connected Vehicles
GUI	Graphical User Interface
HAD	Highly Automated Driving
HARA	Hazard Analysis and Risk Assessment
HIL	Hardware-in-the-Loop
HMI	Human Machine Interface
HW	Hardware
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITL	In-The_Loop
ITS	Intelligent Transport System
IVI	Infrastructure to Vehicle Information message
LDWS	Lane Departure Warning System

Acronym / Abbreviation	Meaning
LKA	Lane Keeping Assist
LCA	Lane Centring Assist
LRR	Long Range Radar
LSG	Legal Safe Gap
MAP	MapData message
MIO	Most Important Object
MRR	Mid Range Radar
MVC	Modular Vehicle Combinations
OBD	On-Board Diagnostics
OS	Operating system
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OOTL	Out-Of The-Loop
OTA	Over The Air
PAEB	Platooning Autonomous Emergency Braking
PMC	Platooning Mode Control
QM	Quality Management
RCP	Remote Control Parking
ROS	Robot Operating System
RSU	Road Side Unit
SA	Situation Awareness
SAE	SAE International, formerly the Society of Automotive Engineers
SCH	Service Channel
SDO	Standard Developing Organisations
SIL	Software-in-the-Loop
SOTIF	Safety of the Intended Function
SPAT	Signal Phase and Timing message

Acronym / Abbreviation	Meaning
SRR	Short Range Radar
SW	Software
TC	Technical Committee
TF	Task Force
TOR	Take-Over Request
TOT	Take-Over Time
TTG	Target Time Gap
UNECE	United Nations Economical Commission of Europe
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to any (where x equals either vehicle or infrastructure)
VDA	Verband der Automobilindustrie (German Association of the Automotive Industry)
VECTO	Vehicle Energy Consumption Calculation Tool
VMAD	Validation Method for Automated Driving
WIFI	Wireless Fidelity
WLAN	Wireless Local Area Network
WP	Work Package