



Fifth Generation Communication Automotive Research and innovation

Deliverable D5.2

The 5GCAR Demonstrations

Version: v1.0

2019-07-31

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761510. Any 5GCAR results reflects only the authors' view and the Commission is thereby not responsible for any use that may be made of the information it contains.



This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 761510

5G PPP

<http://www.5g-ppp.eu>

Deliverable D5.2

The 5GCAR Demonstrations

Grant Agreement Number:	761510
Project Name:	Fifth Generation Communication Automotive Research and innovation
Project Acronym:	5GCAR
Document Number:	5GCAR/D5.2
Document Title:	The 5GCAR Demonstrations
Version:	v1.0
Delivery Date:	2019-07-31
Editors:	Bastian Cellarius (Ericsson) Tobias Frye (Bosch) Jérôme Tiphène (PSA Group) Jürgen Otterbach (Nokia)
Authors:	Nadia Brahmi, Tobias Frye (Bosch) Xabier Parcerero, Fernando Fernández, José Luis Rodríguez, Sebastián Dadín, Hadrián Grille (CTAG) Bastian Cellarius, Massimo Condoluci (Ericsson) Hanwen Cao (Huawei) Omar Nassef (King's College London) Karine Berger, Rémi Theillaud (Marben Products) Jürgen Otterbach, Stephan Saur (Nokia) Sylvain Allio, Mathieu Lefebvre, Jean-Marc Odinet, Elias Salam, Frédéric Gardes (Orange) Jérôme Tiphène, Alain Serval, Antonio Fernández (PSA Group) Hellward Broszio, Kai Cordes (VISCODA) Björn Bergqvist (Volvo Cars)
Keywords:	5GCAR, Testbeds, Use Cases, Interfaces, Integration Plan, Lane Merge Coordination, See-Through, Sensor Sharing, VRU Protection, Cooperative Maneuver, Cooperative Perception, Cooperative Safety
Status:	Final
Dissemination level:	Public

Abstract

The 5GCAR project successfully delivered four demonstrations on the 5GCAR final event on the 27th of June 2019, each representing different automotive use cases in corresponding environments. To that end, various 5G concepts were applied to the demonstration design of these use cases and implemented in a realistic environment on an automotive test track. More importantly, an end-to-end implementation for each use case was done, enabling a detailed analysis of the feasibility of said solutions, focusing on the identification of challenges to be addressed in a real deployment.

For the Lane Merge Coordination demonstration, a complete system capable of combining input from connected vehicles and roadside cameras and devising individual maneuvers was implemented and evaluated. Several KPIs were evaluated, from both the automotive and the communication domain, where the focus was set on the latency performance and dependencies in the system.

For the See-Through Sensor Sharing demonstration, a 5G-NR prototype was used to support a rear vehicle to enhance its range of vision by streaming a video from a front vehicle corresponding to the region occluded by the front car. Application KPIs were evaluated and mapped onto communication KPIs, namely high throughput, low latency, and high reliability. Furthermore, several antenna patterns were tested and evaluated in simulations.

For the Long-Range Sensor Sharing demonstration, an on-board LiDAR sensor was used to detect and localize unconnected vehicles. By combining this information with the shared status of connected vehicles, collisions were predicted, and corresponding warnings were distributed to affected vehicles. The evaluation focused on the enhanced driving comfort with respect to reduced deceleration in a collision case due to timely warnings, and on the collision prediction timing.

For the Vulnerable Road User demonstration, a 5G-NR positioning prototype was used for enhanced positioning of both a vehicle and a pedestrian dummy. Positioning information was used to predict likely collisions, based on an advanced tracking algorithm. Here, the evaluation focused on the positioning accuracy of the 5G-NR prototype, and on the collision prediction performance including the reliability of alarm messages.



Executive Summary

The role of the demonstrations in the 5GCAR project is to showcase and assess the potentials of 5G and understand the impact and challenges when combining 5G communication with foreseen use cases in autonomous driving. Also, selected technological advances of the 5GCAR concept work were adopted and successfully implemented in the demonstrations.

The 5GCAR demonstrations were showcased in June 2019, offering a complete implementation of the following use cases, based on the use case classes identified in [5GC19-D21].

1. Lane Merge Coordination, from the Cooperative Maneuver use case class.
2. See-Through Sensor Sharing, from the Cooperative Perception use case class.
3. Long Range Sensor Sharing, also from the Cooperative Perception use case class.
4. Vulnerable Road User Protection, from the Cooperative Safety use case class.

Each demonstration was realized as a joint effort among partners of the automotive and telecommunication industry, academia, and smaller or medium sized enterprises focused on specific expert topics. Aside from implementation of individual software components and optimization of existing products, a key focus and challenge has been the integration of such components from different partners for a complete ecosystem, combining the components of all demonstrations. As a result, a representative end-to-end concept is presented, with a focus on individual aspects for the different use cases, which is designed for autonomously driving cars, but also capable of supporting human drivers, as it was done on the 5GCAR final event.

The lane merge coordination use case deals with the orchestrated creation of gaps for cars entering a motorway, using cellular communication and a centralized lane merge coordination function. In the implemented scenario, a fixed camera installation near the merging point is used to detect vehicles that are not connected, and thus cannot receive instructions or communicate information about themselves. In the demonstration work of the project, a complete system capable of combining input from connected vehicles and roadside cameras and devising individual maneuvers was implemented and evaluated. Communication-wise, all road users were connected using an experimental cellular network, implementing 5G deployment models, namely edge computing, network slicing, and Quality of Service. Several KPIs were evaluated, from both the automotive and the communication domain, where the focus was set on the latency performance and dependencies in the system. It was shown that the latency contribution by the radio access network is significant, but other contributors are at least as significant, mainly the deployment of components (should be coordinated, preferably in an edge cloud), the aggregation of data from multiple inputs for the data fusion, and the proper configuration of higher layer protocols for low latency operation.

The see-through demonstration is a cooperative perception application that showcases the extension of the vehicle's awareness using real-time video shared by a nearby vehicle to provide support during overtaking maneuvers. The main idea is to use a stereo vision system mounted



on a front vehicle to generate a local 3D map of the environment, in which a rear vehicle could localize itself using a feature tracking algorithm. This means the rear vehicle compares its own camera features with the ones received from the front vehicle. Based on the information about the relative pose (position and orientation) of the rear vehicle, the front vehicle generates a new synthetic image with the perspective of the rear vehicle, from which only the region of interest is transferred to the rear vehicle to be displayed. In the demonstration work of the project, a 5G-NR prototype was used for a see-through application, which supports a rear vehicle to enhance its range of vision by streaming a video from a front vehicle corresponding to the region occluded by the front car. Application KPIs were evaluated together with the relevant communication KPIs, namely image quality, fit of the see-through overlay and latency of the video stream. These result in the following communication KPIs: high throughput, low packet error rate and low latency. For the demonstration, the performance of the 5G-NR prototype was able to meet the requirements of the see-through application.

The cooperative perception use case class targets the extension of each vehicle's perception range based on sensor data collected from different sources. Such information can be used to detect critical situations further in advance, either by estimating such situations inside the vehicles (which was done in the See-Through use case), or in an infrastructure component. The infrastructure-based approach offers more computational power in the edge cloud and more dynamic learning skills thanks to the multi-vehicle and multi-scenario input and is pursued in this use case. In the demonstration work of the 5GCAR project, an on-board LiDAR sensor was used to detect and localize unconnected vehicles. By combining this information with the shared status of connected vehicles, collisions were predicted, and corresponding warnings were distributed to affected vehicles. The evaluation focused on the enhanced driving comfort with respect to reduced deceleration in a collision case due to earlier warnings, and on the collision prediction timing. It was shown that the LiDAR produced useful detections up to distances over 100 m, and therefore, collisions could be predicted approximately 2 s ahead of collision, with a confidence of over 80%. The dynamics between collision forecast time and prediction confidence was presented in more detail.

In the era of autonomous driving and smart mobility, the protection of vulnerable road users (VRU) like pedestrians, cyclists, scooters, etc. becomes increasingly important. With the aid of communication, vehicles and VRUs can be informed by upcoming dangerous situation and reduce risk of accidents. To that end, the 5G radio network serves as additional sensor for the vehicle, complementing vehicle internal sensors e.g., video and radar, offering enhanced position estimation for connected road users. In the demonstration work of the 5GCAR project, a communication network was used to exchange status, sensor data and alarm messages between involved traffic users. By tracking such road users and extrapolating their trajectories, collisions have been predicted and affected road users have been warned using alert messages. The evaluation focused on the positioning accuracy of the 5G-NR prototype, and the collision prediction performance including the reliability of alarm messages. The positioning accuracy was found to be superior to GPS at higher quantiles in the concrete given scenario, but it should be noted that this is not necessarily universally valid. For the reliability of alarm messages, the dynamics between collision forecast time and collision warning reliability was presented, and it



was shown that a time window between one or two seconds was identified as a good compromise, allowing a collision warning reliability of approximately 90%.

Aside from presenting the technical outcome of the 5GCAR demonstrations, technical and organizational learnings from the project are summarized in this document. On a general notice, a key learning was that combining the demonstrations in a common ecosystem was a great benefit when it comes to reusing common development, integration, and testing support tools, flexibly sharing vehicles among different demonstrations, as well as sharing insights on performance issues and possible improvements across demonstrations. Furthermore, a central coordination and detailed logging of the executed tests was essential for efficient test execution and post-processing of results. Also, collecting and organizing logs with rich information (component, ingress/egress, timestamp, location, etc.) as well as a millisecond-accurate clock synchronization are key to facilitating the evaluation. Finally, the challenge of managing a high number of prototypes, and especially multiple levels of dependencies between them, can be a big blocker in the development process, which should be avoided or at least addressed when planning such demonstrations.



Contents

1	Introduction	11
1.1	Objective of the Document	11
1.2	Structure of the Document	11
2	5GCAR Demonstrations.....	12
2.1	Lane Merge Coordination.....	14
2.2	See-Through Sensor Sharing	17
2.3	Long Range Sensor Sharing.....	20
2.4	Vulnerable Road User Protection.....	23
3	Architecture and Key Concepts.....	25
3.1	Lane Merge Coordination.....	27
3.1.1	Detection of Vehicles by Intelligent Camera System	28
3.1.2	Data Fusion	30
3.1.3	Traffic Orchestrator	32
3.1.4	Optimized Message Delivery	34
3.2	See-Through Sensor Sharing	35
3.2.1	Details on Application	35
3.2.2	Details on the Connectivity.....	37
3.3	Long Range Sensor Sharing.....	39
3.3.1	Detection of Unconnected Vehicle	39
3.3.2	Prediction of Collision	44
3.4	Vulnerable Road User Protection.....	45
3.4.1	Real-Time Localization of Road Users with the 5G Radio Network	46
3.4.2	Trajectory Estimation with Sensor Fusion	48
3.4.3	Collision Prediction and Release of Warning Messages	49
3.4.4	Execution of the Demonstration	51
3.5	In-Vehicle Setup	53
3.5.1	On-Board Unit.....	53
3.5.2	Vehicle Sensors.....	54
3.5.3	Tablet HMI	54
3.6	Tools.....	56



3.6.1	Cloud Deployment	56
3.6.2	Data Injector	57
3.6.3	Tablet Monitoring Mode	59
3.6.4	KPI Evaluation Platform	62
3.6.5	Post-Processing with MATLAB for Parameter Tuning	64
3.6.6	Hardware Tools for VRU Positioning System	64
4	Demonstration Evaluation	65
4.1	Lane Merge Coordination.....	65
4.1.1	Camera System Localization Accuracy	65
4.1.2	Automotive KPIs	70
4.1.3	Communication KPIs	73
4.1.4	Learnings.....	82
4.2	See-Through Sensor Sharing	83
4.2.1	Application KPIs.....	83
4.2.2	Communication KPIs	85
4.3	Long Range Sensor Sharing.....	86
4.3.1	Automotive KPIs	86
4.3.2	Communication KPIs	88
4.3.3	Qualitative KPIs	89
4.3.4	Evaluation and Conclusions.....	90
4.4	Vulnerable Road User Protection.....	93
4.4.1	Key Performance Indicators.....	93
4.4.2	Evaluation Results	94
5	Conclusions and Learnings.....	97
6	References	99
A	Technical Component Details	101
A.1	Vehicle Position Extrapolation in Data Fusion.....	101
A.2	ELK Stack Interfaces	101
A.3	Data Injector Mechanisms.....	104
B	PSA Car Integration	109
B.1	VBOX Definition.....	109
B.2	CAN Bus Interface	110



B.3	HMI Interface	110
C	PSA Car Definitions	113
C.1	DS 7 Crossback	113
C.2	Peugeot 5008	114
C.3	Peugeot 3008	115
D	Volvo Car Integration	116
D.1	CAN Signal Definition	117
D.2	Volvo CAN Gateway	117



List of Abbreviations and Acronyms

5G	Fifth Generation
5GAA	5G Automotive Association
AoA	Angle of Arrival
CDD	Cyclic Delay Diversity
CNN	Convolutional Neural Network
CV	Constant Velocity
DQN	Deep Q-Network
E2E	End to End
ECU	Electronic Control Unit
EKF	Extended Kalman Filter
ETSI	European Telecommunications Standards Institute
FR	Frequency Range
F-TDoA	Fair-Time Difference of Arrival
gNB	Next Generation NodeB
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
HMI	Human-Machine Interface
IP	Internet Protocol
ITS	Intelligent Transport System
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LiDAR	Light Detection And Ranging
LOS	Line-Of-Sight
LTE	Long Term Evolution
mPCI	Mini Peripheral Component Interconnect
MQTT	Message Queuing Telemetry Transport
MRC	Maximum-Ratio Combining
NGSIM	New Generation Simulation
NLOS	Non-Line-Of-Sight
NN	Neural Network
NR	New Radio

OBU	On-Board Unit
OTA	Over-The-Air
P-GW	PDN Gateway
PC5	Proximity Services (ProSe) direct Communication interface 5
PDN	Packet Data Network
PF	Particle Filter
PSS	Primary Synchronization Signal
QAM	Quadrature Amplitude Modulation
QCI	QoS Class Identifier
QoS	Quality of Service
RL	Reinforcement Learning
ROS	Robot Operating System
RSRP	Reference Signal Received Power
RTK	Real-Time Kinematic
RTT	Round Trip Time
Rx	Receiver
S-GW	Serving Gateway
SSS	Secondary Synchronization Signal
TCP	Transmission Control Protocol
TEQMO	TEQnology for MObility
ToA	Time of Arrival
TTI	Transmission Time Interval
Tx	Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UKF	Unscented Kalman Filter
UPS	Uninterrupted Power Supply
UTAC	United Test and Assembly Center
UTDOA	Uplink Time Difference Of Arrival
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VRU	Vulnerable Road User
WiFi	Wireless Fidelity



1 Introduction

The 5GCAR project demonstrated four use cases from the use case classes defined in [5GC19-D21] in a demonstration event on June 27, 2019. This document presents the demonstrations that were executed in detail and discusses the respective technical realizations. Furthermore, learnings gathered in the demonstration work of 5GCAR are shared.

1.1 Objective of the Document

This document summarizes the outcome of the demonstration work in 5GCAR. As part of this, the general use cases behind the four demonstrations developed within the project are described, together with the technical realization, and a corresponding evaluation. The objective is to evaluate the implemented use cases and assess the feasibility of the chosen approaches.

Finally, another purpose is to share insights and learnings gathered in the demonstration work of the project, covering tools and ways of working that proved to be helpful, but also technical challenges that should have been identified and addressed earlier. This point covers both technical and organizational aspects.

1.2 Structure of the Document

The document describes each demonstration, as well as common aspects, on different levels, as explained in the following.

Chapter 2 introduces the individual demonstrations, as well as the common demonstration framework, on a high level, covering also information on how the demonstrations were showcased in the 5GCAR event. The four demonstrations are presented in Sections 2.1, 2.2, 2.3, and 2.4.

Chapter 3 elaborates on the demonstration architecture in more detail, starting with a functional description of the common framework that was used by multiple demonstrations, and outlining the deployment of the introduced components on the test track. The specifics of the four demonstrations are described in Sections 3.1, 3.2, 3.3, and 3.4, following the same section pattern as in Chapter 2. Furthermore, a common in-vehicle setup is described in Section 3.5, which was used by multiple demonstrations, and a set of tools that proved helpful for the integration and testing of the demonstrations are introduced in Section 3.6.

Chapter 4 discusses the evaluation of each demonstration in Sections 4.1, 4.2, 4.3, and 4.4, following again the section pattern from the other chapters.

Chapter 5 summarizes the work of all demonstrations, and describes general technical and organizational learnings, including recommendations for other projects.

2 5GCAR Demonstrations

The 5GCAR project identified five use case classes in [5GC19-D21], which are illustrated in Figure 2.1. As representatives of these use cases, four different demonstrations were implemented as field tests and showcased in an event on June 27, 2019. These use cases are:

1. Lane Merge Coordination, from the Cooperative Maneuver use case class
2. See-Through Sensor Sharing, from the Cooperative Perception use case class
3. Long Range Sensor Sharing, also from the Cooperative Perception use case class
4. Vulnerable Road User Protection, from the Cooperative Safety use case class

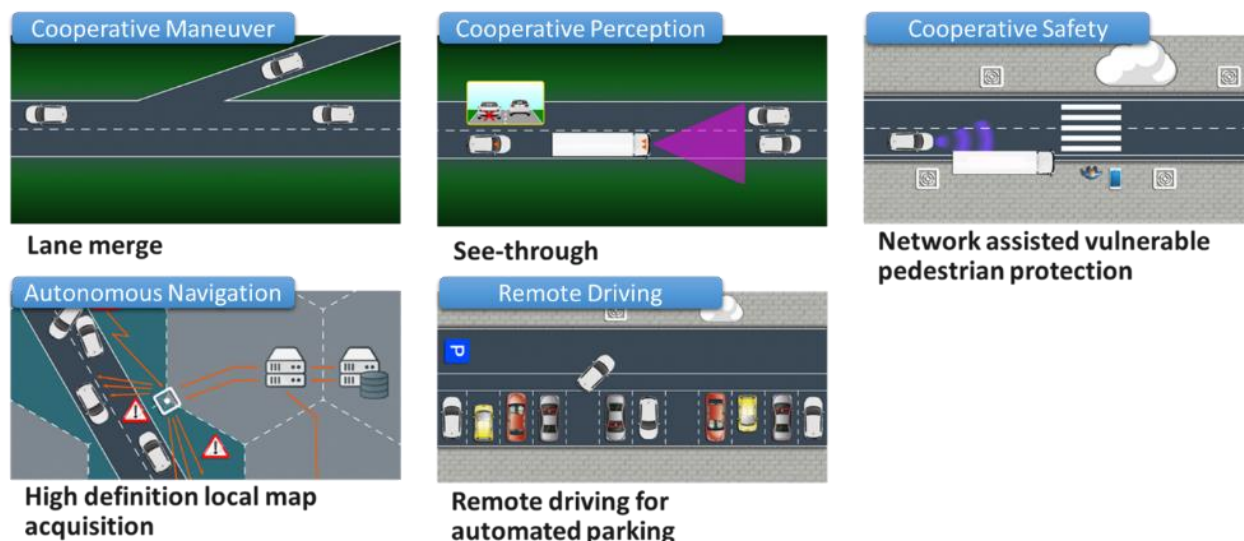


Figure 2.1: Five use case classes from [5GC19-D21], based on which four demonstrations were showcased and evaluated in field tests.

The demonstrations were showcased on various areas of the UTAC TEQMO test track [UTAC-TEQMO] (cf. Figure 2.2), each fitting the target environment of the respective use case. Autonomous cars were not used in the demonstrations, because they are still at the forefront of automotive research, and not available for communication projects. Instead, human drivers were driving the cars in the demonstrations.

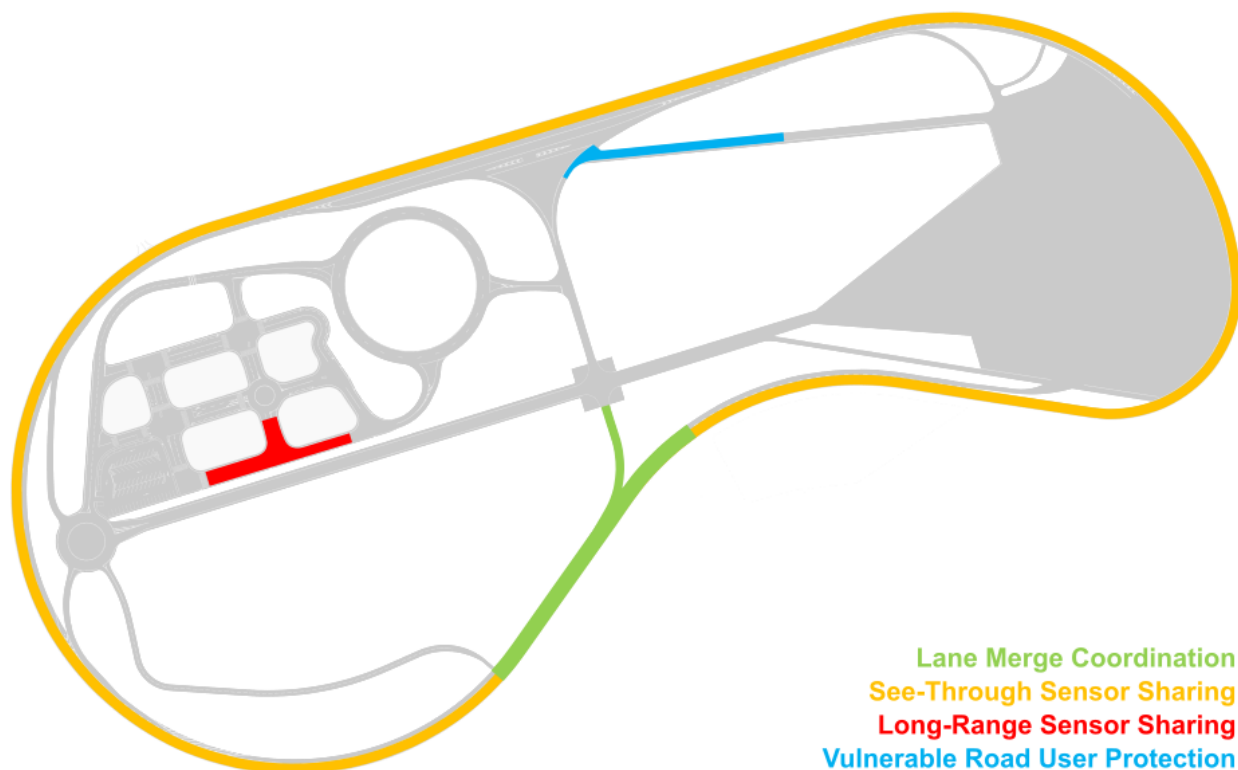


Figure 2.2: Different areas of the test track were used for the four 5GCAR demonstrations, each representing the target environment of the respective use case.

The purpose of the field tests was to showcase and evaluate the value of 5G for automotive use cases. Specifically, the 5G aspects highlighted in Figure 2.3 were used in the demonstrations.

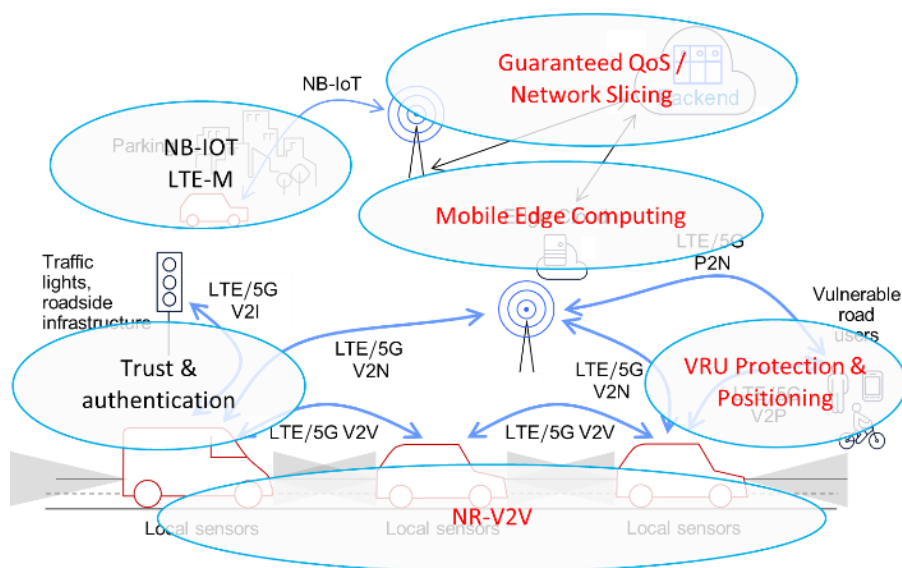


Figure 2.3: Different 5G aspects (highlighted in red) were implemented and evaluated in the demonstrations.

2.1 Lane Merge Coordination

The Lane Merge Coordination use case deals with the orchestrated creation of gaps for cars entering a motorway, using cellular communication and a centralized lane merge coordination function. In the implemented scenario, a fixed camera installation near the merging point is used to detect vehicles that are not connected, and thus cannot receive instructions or communicate information about themselves, as illustrated in Figure 2.4.

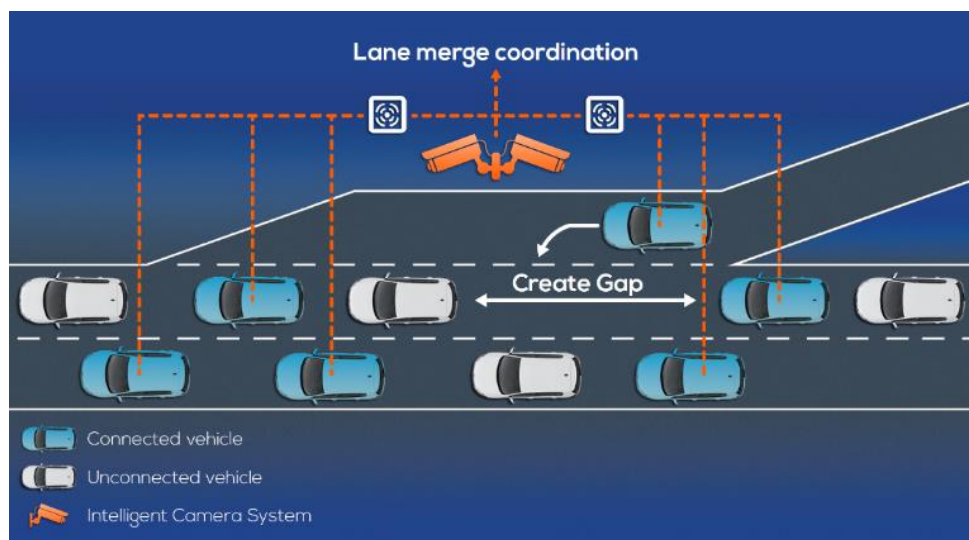


Figure 2.4: In the lane merge coordination use case, a central coordination entity plans the creation of a gap for vehicles entering a motorway.

The demonstration was executed on the highway part of the UTAC TEQMO test track (cf. Figure 2.5), with one vehicle joining the highway coming from the central crossing in the center of the test track, and four vehicles driving on the highway, two of which are connected.

Technically, the setup was designed for autonomously driving cars, but the demonstrations were executed with human drivers getting instructions on the Human Machine Interface (HMI), based on received maneuver recommendations, as shown in Figure 2.6.

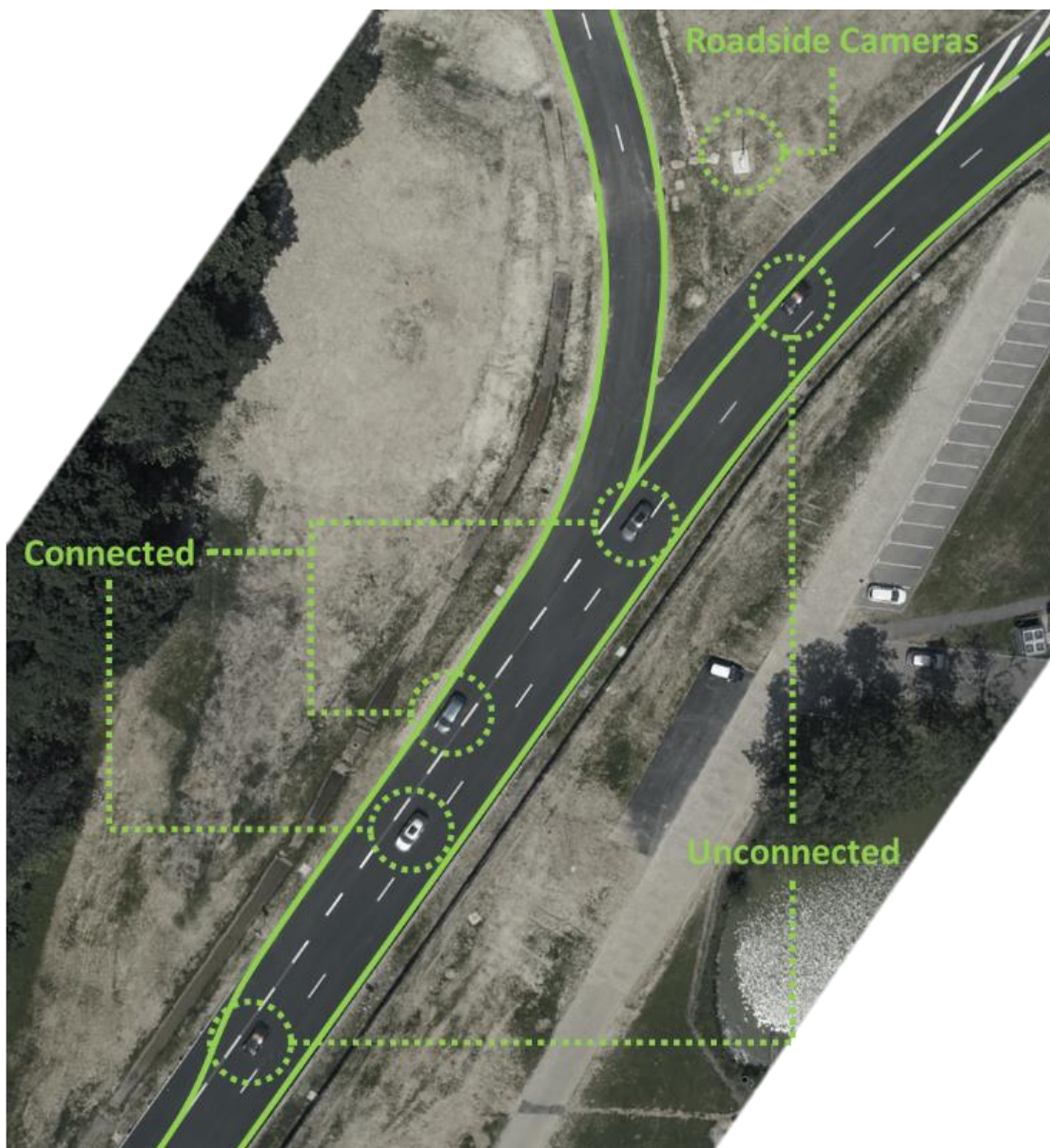


Figure 2.5: The lane merge area used for the demonstration consists of a section of the highway, and an entering lane coming from the central crossing.



Figure 2.6: The in-vehicle HMI, as well as the tablet HMI, were used to provide warnings and instructions to the drivers.

While lane merges already work without the use of inter-vehicle communication, coordinating the vehicles this way can greatly increase traffic efficiency, comfort, and safety. By keeping the communication latency small and stable, more time is available to aggregate data, and for various algorithms that are part of the use case, while still being reactive to unforeseen changes in the driving behavior on the road. This is enabled by 5G communications with edge computing, network slicing, and quality of service, in particular in scenarios like highway ramps in large cities, where a lot of vehicles need to be coordinated at the same time. For such scenarios, 5G brings the needed scalability with more and more cars being connected and coordinated in the future.

2.2 See-Through Sensor Sharing

The see-through demonstration is a cooperative perception application that showcases the extension of the vehicle's awareness using real-time video shared by a nearby vehicle to provide support during overtaking maneuvers. The main idea is to use a stereo vision system mounted on a front vehicle to generate a local 3D map of the environment, in which a rear vehicle could localize itself using a feature tracking algorithm. This means the rear vehicle compares its own camera features with the ones received from the front vehicle. Based on the information about the relative pose (position and orientation) of the rear vehicle, the front vehicle generates a new synthetic image with the perspective of the rear vehicle, from which only the region of interest is transferred to the rear vehicle to be displayed.

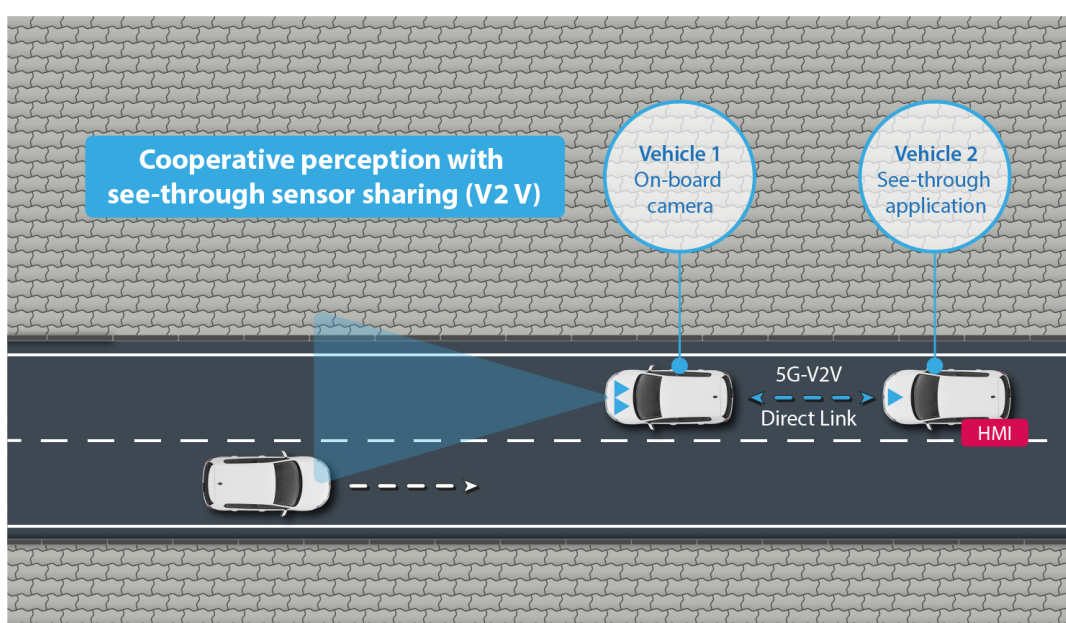


Figure 2.7: In the see-through use case, a front view captured by the vehicle camera is shared with a rear vehicle using a direct V2V link to assist during an overtaking maneuver.

The see-through sensor sharing scenario is shown in Figure 2.7. A vehicle (Vehicle 1) equipped with a front-facing stereo camera system drives along a road, and a second vehicle (Vehicle 2) equipped with the same camera system and a see-through HMI follows behind. The stereo camera is only necessary for the vehicle driving in front, but in order to allow a changed order of the vehicles, the second vehicle is also equipped with a stereo camera.

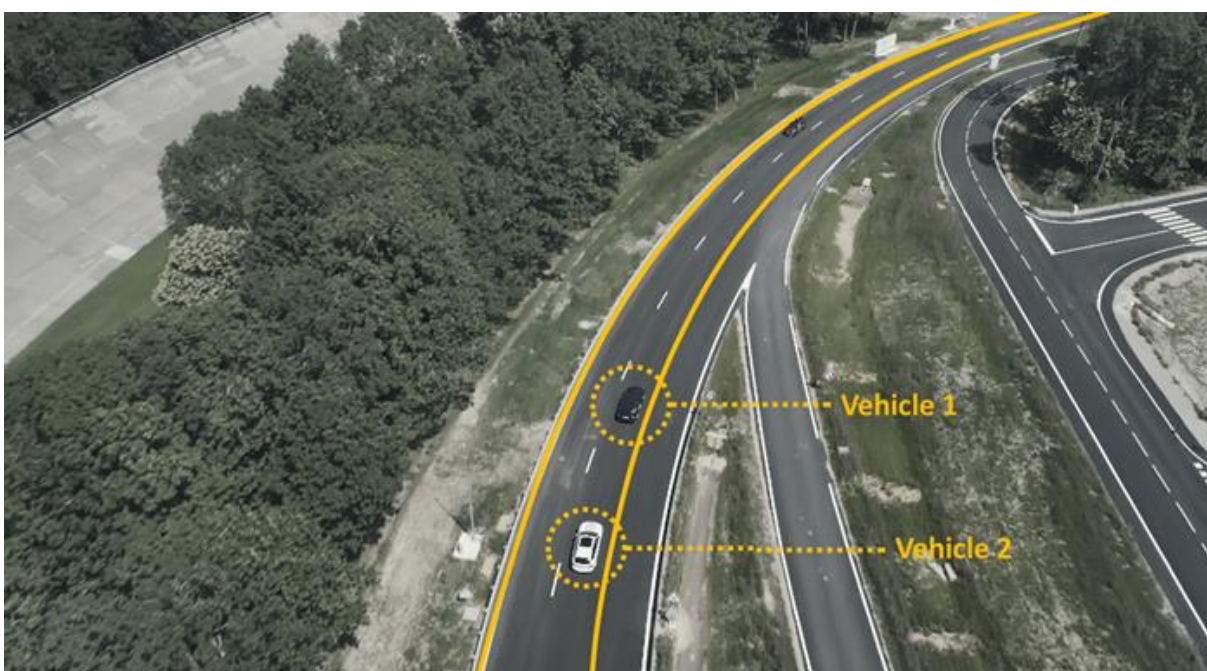


Figure 2.8: Scene to demonstrate the benefits of the see-through application before an overtaking maneuver.

In Figure 2.8, the scene is seen from the top. Vehicle 1 is providing a see-through video sharing to Vehicle 2, which can show this video information in order to assist the driver while overtaking to prevent dangerous situations. During the demonstration, the HMI in the rear vehicle showed a seamless 'see-through' effect in which the front vehicle was transparent on the rear vehicle's HMI, as illustrated in Figure 2.9.



Figure 2.9: See-through HMI inside the rear vehicle.

The see-through application is only possible with the use of a communication link between the involved vehicles. Without communication, a technically simpler solution would be possible, where



a truck displays the video stream of its own cameras on its rear [SAM15], but the result will not fit to the perspective of the rear car.

To ensure the seamless “see-through” operation, a highly reliable and low latency direct communication link between the two vehicles is required. This low latency is particularly needed because the real-time video representing the front scene can only be transmitted after an accurate relative pose (position and orientation) between the two vehicles is calculated. For this reason, a direct communication link is necessary, as a cellular link will induce additional delays and not give extra benefits. For this project, the sidelink or PC5 interface is used and is evaluated in Section 4.1.4.

2.3 Long Range Sensor Sharing

The cooperative perception use case class targets the extension of each vehicle's perception range based on sensor data collected from different sources. Such information can be used to detect critical situations further in advance, either by estimating such situations inside the vehicles (which was done in the see-through use case), or in an infrastructure component. The infrastructure-based approach offers more computational power and more dynamic learning skills thanks to the multi-vehicle and multi-scenario input and is pursued in this use case.

The cooperative perception increases the line of sight range based on the data exchanged from different sensors. The infrastructure-based algorithm can use more computational power and better learning skills thanks to the multivehicle and multi-scenario perception.

The collision detection alert from the infrastructure shall:

- Increase safety by avoiding collisions.
- Increase the comfort by predicting risk in advance and adapting softly the speed.

The infrastructure gets enhanced information thanks to the on-board sensors and the vehicle benefits from the off-board computational capability.

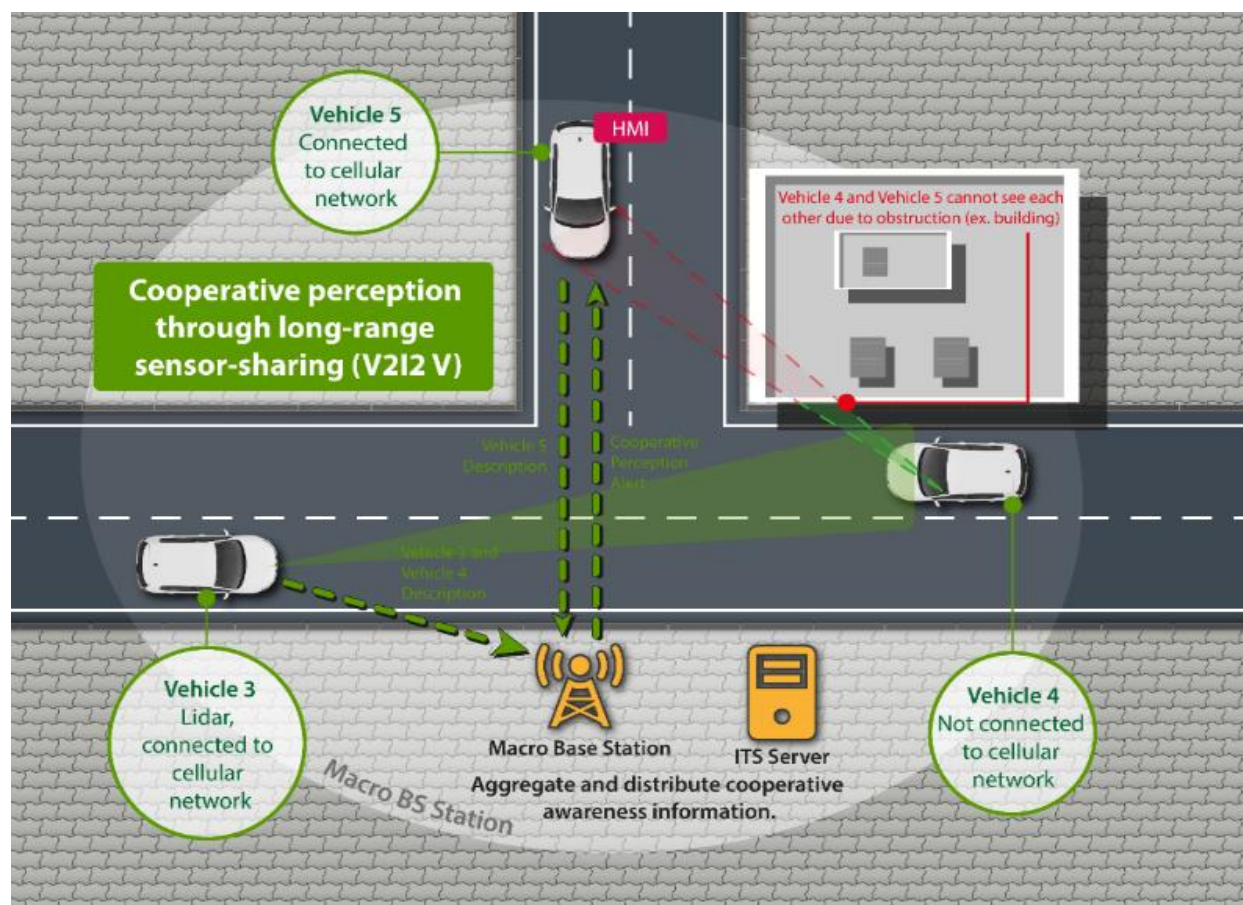


Figure 2.10: Illustration of the Long-Range Sensor Sharing demonstration scenario.

In the defined scenario (see Figure 2.10), the perpendicular intersection has reduced visibility.

- Vehicle 5 has the priority to turn right.
- Vehicle 4, a non-connected vehicle, may not respect the other lane preference.
- Vehicle 3 will detect and share the available on-board information about Vehicle 4 with the infrastructure.
- The infrastructure will compute in real time the trajectories of both vehicles, sending an alert to Vehicle 5 when a collision risk is detected.

This scenario is evaluated in the City area of the TEQMO test site, as shown in Figure 2.11.



Figure 2.11: Part of city area used for this demonstration.

In this demonstration, two scenarios were tested:

- Non-collision scenario: Vehicle 4 drives at 50 km/h, Vehicle 5 at 40 km/h. Vehicle 4 crosses the intersection before Vehicle 5, no risk, no alert.
- Collision scenario: both Vehicle 4 and 5 are at 40 km/h. The probability of collision is high, and Vehicle 5 must brake to avoid collision. An alert is sent to Vehicle 5, as shown in Figure 2.12.



Figure 2.12: Alert message in the embedded HMI.

Without communication, human driver or autonomous vehicle are limited by the line of sight. The benefit of communication is to improve security and comfort giving some extra delay to react to a dangerous situation.

The benefit of 5G for this use case is that the architecture natively supports the deployment of compute and networking resources on the edge of the network, enabling the processing of data close to the final users, thereby reducing latency down to the very tight requirements imposed by such a safety-critical use case.

The improved reliability of 5G prevents critical alerts from being lost or delayed due to necessary retransmission.

2.4 Vulnerable Road User Protection

In the era of autonomous driving and smart mobility, the protection of vulnerable road users (VRU) like pedestrians, cyclists, scooters, etc. becomes increasingly important. Life threatening in traffic situations by more and more smartphone user generate an increase the number of traffic accidents. New radio technology 5G NR will support high accurate location services of communications devices, that can be tracked in higher precision compared to previous radio technologies; new countermeasures are under discussion e.g. in ETSI-ITS and 5GAA to increase humans' safety in the future using new 5G NR features.

By the present network layer there is a communication between different entities to exchange status, sensor data and alarm messages between involved road users. With the aid of the common network the driver in the vehicle or the VRU can be direct informed by upcoming dangerous situation and reduce risk of accidents.

The 5G radio network serves as additional sensor for the vehicle, complementing vehicle internal sensors e.g., video and radar. In the network layer these functions are implemented in the so-called Location Server that tracks the movement of a vehicle and a VRU and anticipates potentially critical situations. A schematic view is shown in Figure 2.13, where a VRU is about to cross the street behind an obstacle. The demonstration was executed on the UTAC TEQMO test track in the area highlighted in Figure 2.14.

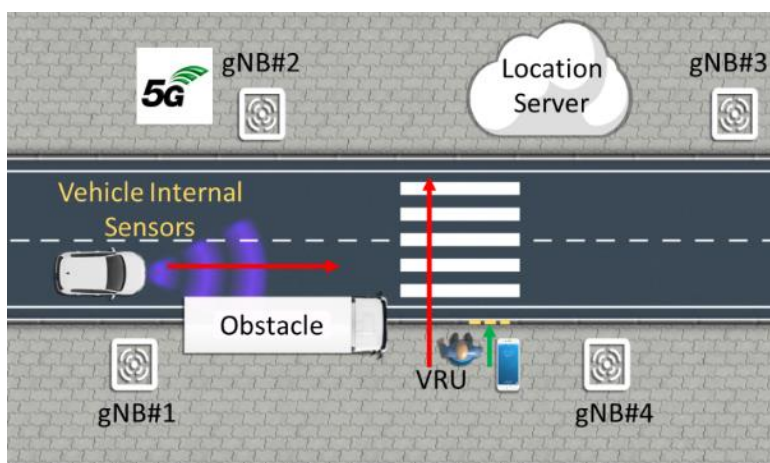


Figure 2.13: The 5G radio network serves as additional sensor. In case of a critical situation (red arrows) warning messages are sent to protect the vulnerable road user.

Only in case of a critical situation indicated by the red arrows, a warning message is released to the vehicle and the VRU. The Human Machine Interface (HMI) for this is shown in Figure 2.15.

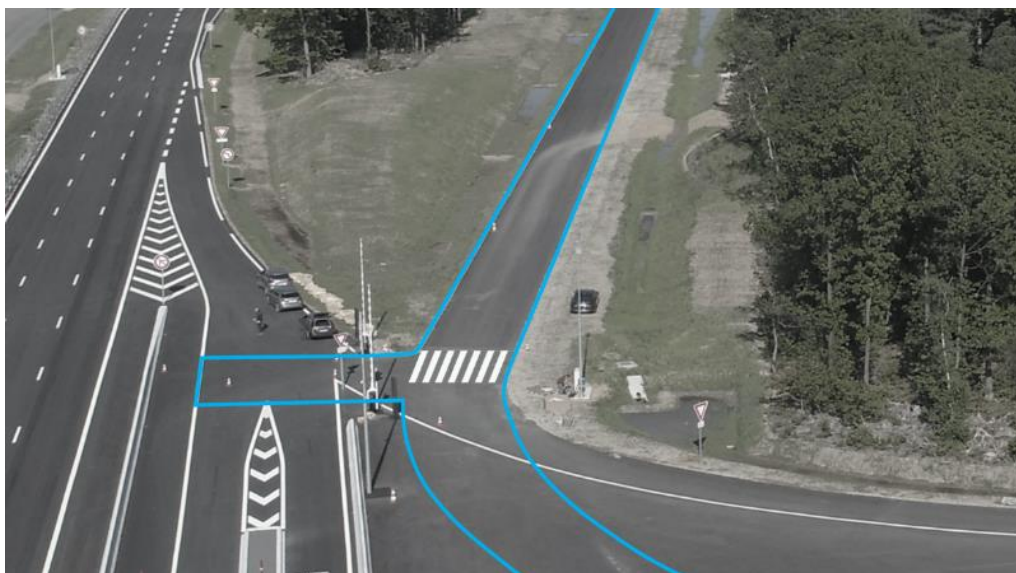


Figure 2.14: VRU protection demonstration area on the test track.



Figure 2.15: Collision alert shown on the HMI in the vehicle.

The main benefit of 5G for radio-based positioning is the bigger system bandwidth of up to 100 MHz in Frequency Range 1 (FR1) and up to 400 MHz in Frequency Range 2 (FR2) that can be aggregated to a maximum bandwidth of 800 MHz [3GPP19-38101]. With increasing RF bandwidth, the time resolution for the individual propagation path becomes better, so the time of arrival of the signals can be determined more accurately. Moreover, with higher carrier frequencies the size of antenna arrays becomes smaller, i.e. panels with a high number of antenna elements can be utilized. This enables accurate Angle of Arrival (AoA) measurements that can complement time measurements for positioning.

3 Architecture and Key Concepts

A common ecosystem to host all implemented demonstrations was developed. It is built around a context-based messaging system, the context-aware V2X gateway, reachable via cellular communication, but also includes sidelink communication between vehicles. Figure 3.1 illustrates the functional architecture of this ecosystem, where connected vehicles and other components ingest information on road users, be it vehicles or VRUs, into the messaging system. Applications can subscribe for information at the V2X Gateway, and depending on subscription parameters, information context, and authorization, the V2X Gateway forwards such information to different applications. Optionally, a common Data Fusion step can be used that combines redundant input information, in conjunction with a dynamic map that maintains a knowledge base, handling e.g. timeout of information.

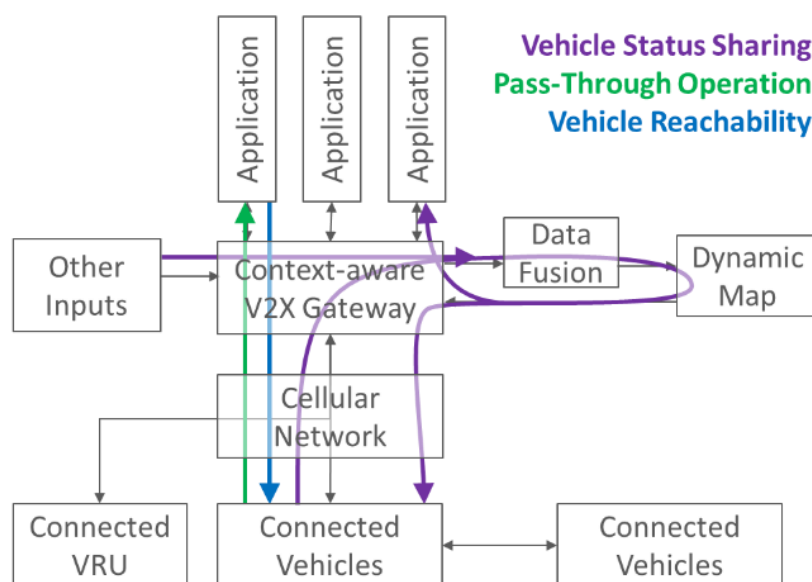


Figure 3.1: The common ecosystem for all demonstrations shares a lot of functionalities for exchanging message-type information, supporting different message flows based on context.

The 5GCAR demonstrations use this system as illustrated in Figure 3.2. Some parts are shared between various demonstrations, where information shared by a vehicle can be consumed by multiple applications at once, offloading the cellular network.

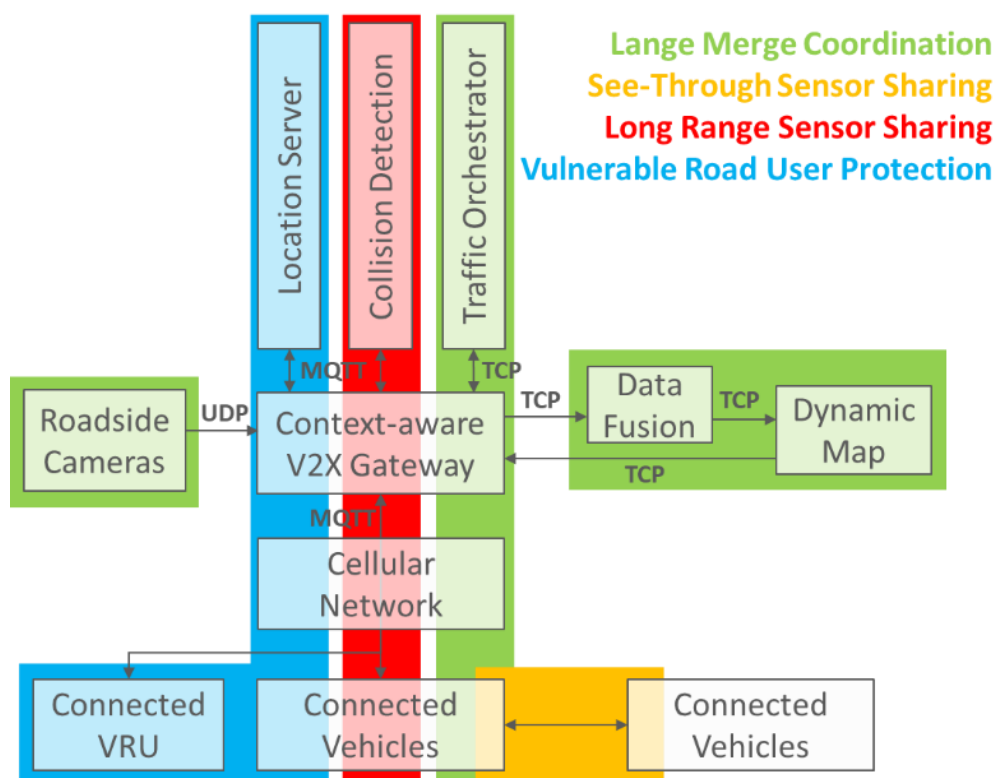


Figure 3.2: The four 5GCAR demonstrations made use of the functional elements as highlighted.

For the different demonstrations, separate applications are connected to the V2X Gateway and consume the road user information. Namely:

1. The Lane Merge Coordination demonstration uses the Traffic Orchestrator, which is further described in Section 3.1.
2. See-Through Sensor Sharing demonstration has all needed applications inside the vehicles and is further described in Section 3.2.
3. The Long-Range Sensor Sharing demonstration uses a collision detection application, which reuses parts of the Location Server, and is further described in Section 3.3.
4. The Vulnerable Road User Protection demonstration uses the Location Server, which is further described in Section 3.4.

The coverage of the cellular network on the test track is provided from an antenna mast next to the test track. On network-side, several sites on the test track are connected for the overall demonstration system, as shown in Figure 3.3. The fiber network of the test track is used to connect roadside cameras and 5G positioning gNBs with corresponding servers in the UTAC chalet, while the cellular network and the other 5GCAR software components are deployed in the Orange Shelter next to the test track. The Orange shelter and the UTAC chalet are connected via a direct fiber link, to inter-connect both systems and aggregate all information in a common place for the respective demonstrations. Finally, only the 5GCAR applications and the user plane(s) of

the cellular network are instantiated locally, while the control plane is hosted in a remote location, in Aachen.

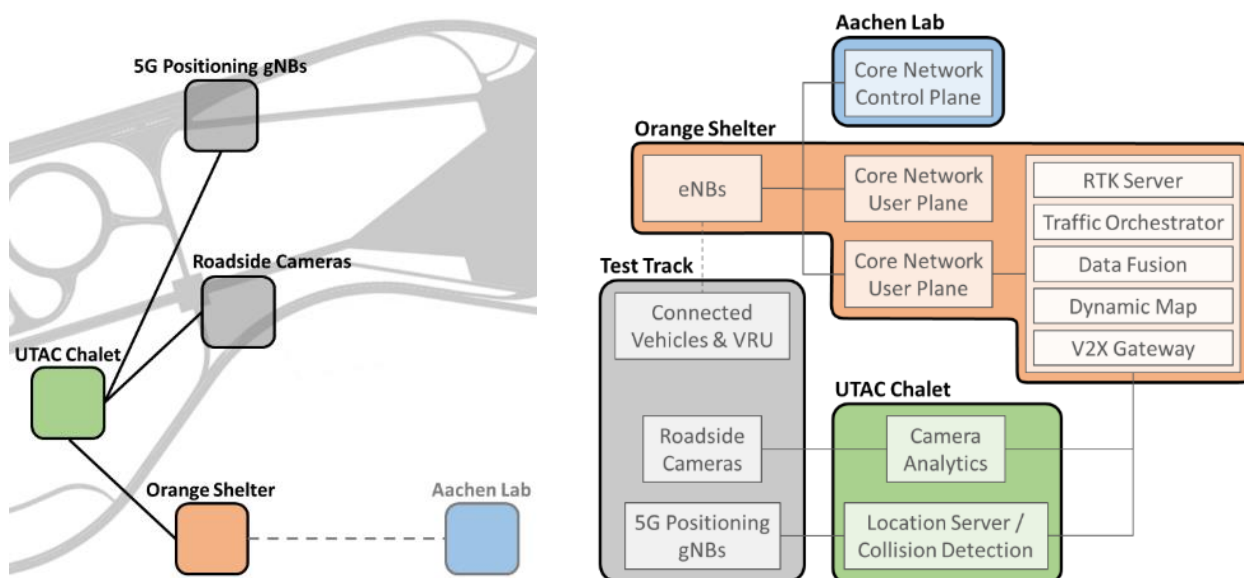


Figure 3.3: The different 5GCAR components were deployed in different locations on or next to the test track, aside from a few non-critical components that were deployed in remotely in Aachen.

On client-side, an On-Board Unit (OBU) is used to connect to the cellular network and integrate with the in-vehicle system and the tablet HMI, as further described in Section 3.5. Aside from a stand-alone Global Navigation Satellite System (GNSS) module, the OBU also has a GNSS Real-Time Kinematic (RTK) module that gets correction signals from an RTK server in the Orange Shelter offered over IP.

More details on the configuration of the cellular network are given in Section 3.1.4.

3.1 Lane Merge Coordination

Figure 3.4 illustrates the functional architecture of the Lane Merge Coordination demonstration, and marks the following key concepts:

1. The intelligent camera system incorporates a mechanism to detect vehicles in a live video sequence and estimate their real-world positions, as well as other parameters of interest.
2. The data collected from the intelligent camera system and from the connected vehicles is partially redundant. Thus, status information that refers to the same vehicle are identified and merged in a Data Fusion step.
3. Based on the collected information on vehicles in the area of interest, the central lane merge coordination entity devises a cooperative maneuver for the lane merge and sends out individual maneuver recommendations to affected vehicles.
4. For communicating all the information between the different entities sufficiently fast and reliably, the cellular network is optimized in several ways, while the Context-aware V2X

Gateway provides a scalable solution for communication between different groups of interest and networks (road authorities, vehicle manufacturers, mobile operators, and others).

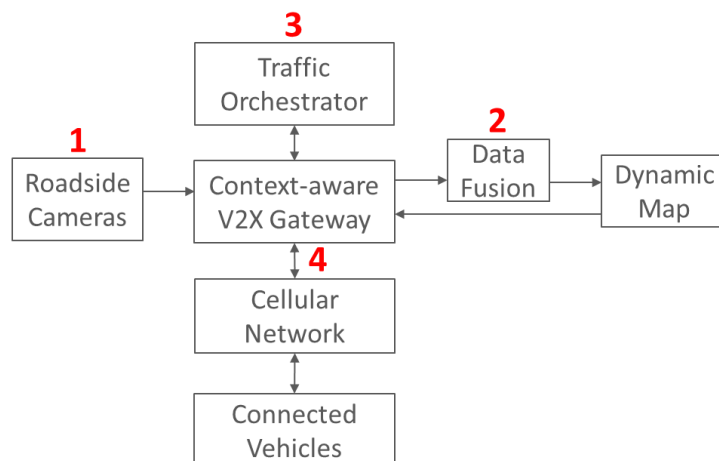


Figure 3.4: The lane merge demonstration is supported by four key concepts, as highlighted.

3.1.1 Detection of Vehicles by Intelligent Camera System

For the lane merge coordination service, it cannot be assumed that every vehicle is connected to the network. Thus, non-connected vehicles must be detected, and corresponding descriptions must be communicated. To fulfill this task, roadside cameras are installed next to the road (cf. Figure 3.4). The camera system consists of monocular video cameras observing the relevant scene content which is required for the lane merge. The observed area includes main and entering lanes within a visual range of up to 200 m. To cover the required visual range in the camera images, three cameras are used. They are mounted on two masts at 10.15 m height (cf. Figure 3.5, left). Two cameras observe the main lane (one for small and one for large distances, mounted at mast AP12) while the third camera observes the entering lane (mounted at AP14). The cameras are calibrated to the GNSS coordinate system [VERM02] to enable representation of measurements in a common coordinate system. To achieve highly-accurate 2D to 3D mapping, the camera calibration procedure incorporates measured distances between the cameras and their height in a constrained optimization technique [VISAPP19]. For visualization, the calibrated cameras and input images are stitched to an overview image, with a top view as shown in Figure 3.6.



Figure 3.5: Left: two of the four cameras are mounted at 10.15 m height. Three camera streams are processed in real-time. The fourth camera serves as a backup. Right: the positions of the cameras: Camera 0 (f=8 mm) and Camera 1 (f=25 mm) on AP12, Camera 2 (f=8 mm) and backup camera on AP14.

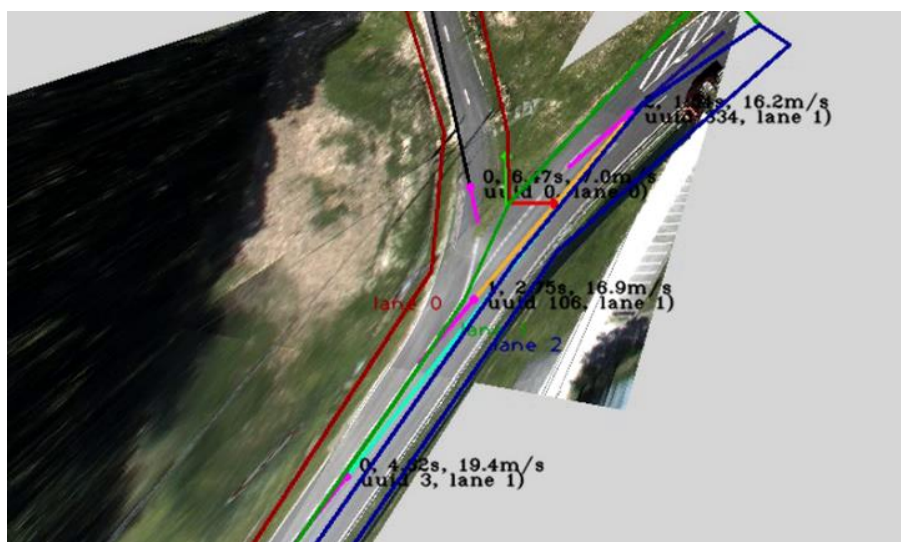


Figure 3.6: Top view of the ground plane using three camera views with five tracked objects shown as pink arrows. Their 3D positions are used for determining the vehicle lane 0 (red polygon), 1 (green polygon), or 2 (blue polygon).

The image signal is processed frame by frame in order to detect and track the vehicles continuously. The processing pipeline is shown in Figure 3.7, left. For the localization of the vehicles, a state-of-the-art CNN-based detector is used [YOLOv3]. The detections in each image are assembled to trajectories based on a feature tracker [FEAT94]. The tracked feature points are classified using the image regions of the detected vehicles. Compared to approaches using the bounding box center for tracking [MOT17], feature based object tracking is more robust, especially in case of occlusions and missing detections. The results from each of the three processed cameras are merged to the final tracking result, providing one localization for each vehicle in the whole scene. The merging between the cameras is done using their overlapping regions. Vehicles from different views are associated by their minimal 3D distance in the global coordinate system. The localization of a vehicle is determined by the projection of a reference 2D image location (the center of the bounding box) into a reference plane of height h_r in the global coordinate system. The merged camera views are visualized in Figure 3.6.

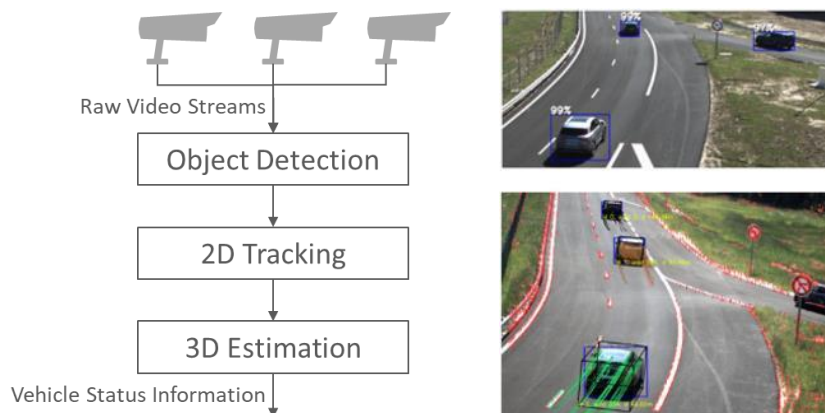


Figure 3.7: The intelligent camera system extracts vehicle status information, including 3D position and orientation, based on 2D image detections (cf. images on the right).

The extracted vehicle status information including position, heading, speed, and lane are transmitted to the Context-aware V2X Gateway. Connected vehicles and unconnected vehicles are not distinguished by the camera system. This is done in a Data Fusion step as explained in Section 3.1.2.

3.1.2 Data Fusion

The Data Fusion function evaluates vehicle descriptions from different sources with various update rates, based on information carried in the descriptions. The aim is to provide the optimal fusion of incoming information which is used for the lane merge maneuver planning. The *Data Fusion* consists of three main steps as shown in Figure 3.8, namely *Data Synchronization*, *Data Association*, and *Tracking*.

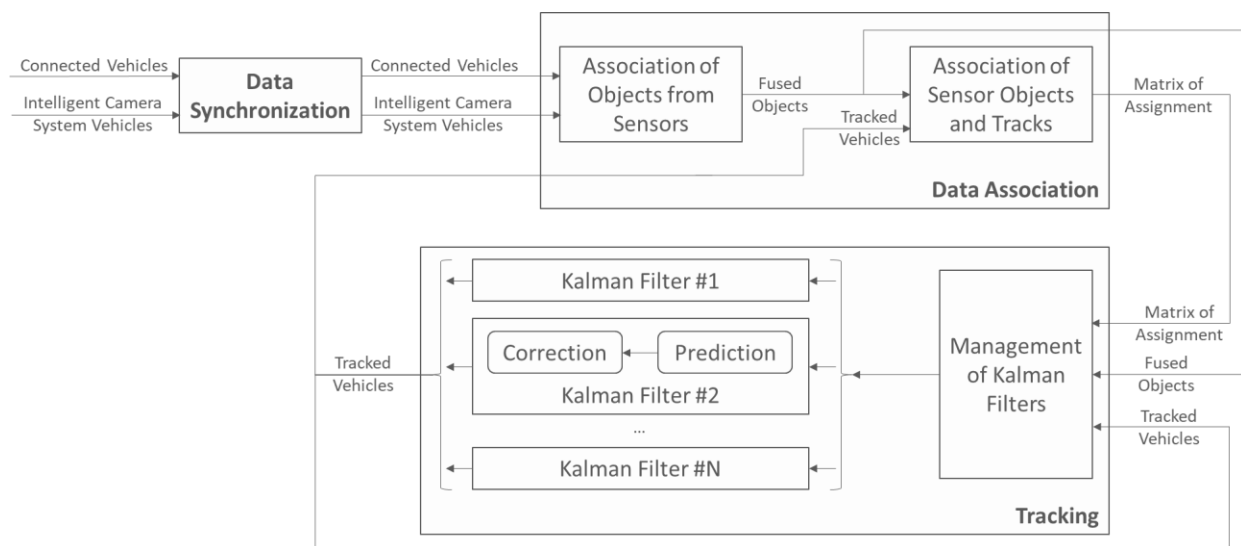


Figure 3.8: The Data Fusion function is composed of three interlinked steps, as illustrated.

The *Data Synchronization* aligns vehicle descriptions from different sources to the same point in time, eliminating the effect of transport delays and data processing durations in the rest of the system, in order to process all information in the same temporal reference. This is achieved by extrapolating each vehicle status to equidistant times (e.g. every 100 milliseconds), updating the GPS positions and the timestamp, as illustrated in Figure 3.9. More detailed information on the position extrapolation can be found in Annex A.1.

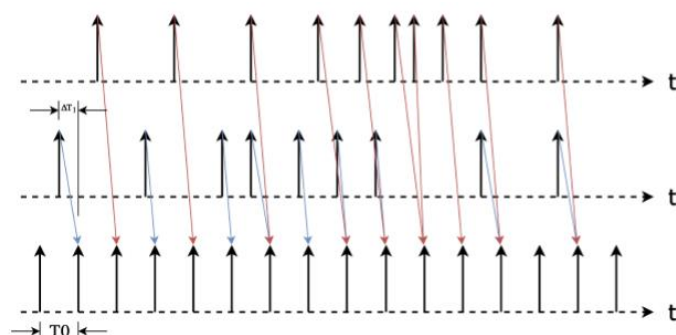


Figure 3.9: The Data Synchronization aligns the timestamp and the GPS point of each vehicle before passing it to the data association.

The *Data Association* matches objects from different sources. This is important since the intelligent camera system provides descriptions for connected and non-connected vehicles. A process of cost minimization is followed, using a cost function based on Euclidean distance or another cost function involving more variables than position (for example, orientation of objects), as illustrated in Figure 3.10.

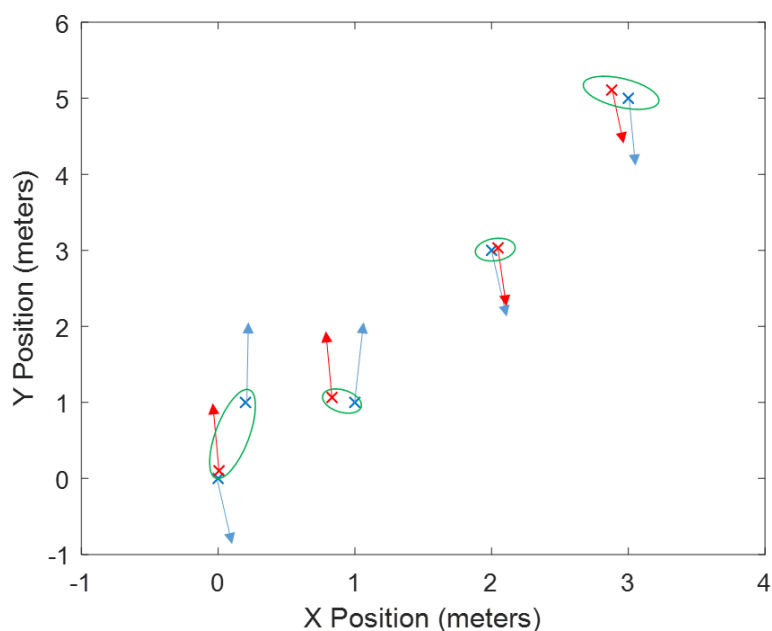


Figure 3.10: In the Data Association step, different parameters (such as position, speed, and heading) are used to associate vehicle status information from different sources.

Tracking is the last step of the data fusion, in which temporal information is added, including a step of estimation and step of correction, where the output from data association is used. This step aims to detect and neglect temporarily bad sensor data (see Figure 3.11), caused by occlusions, wrong operation of sensors, problems with network (in case of connected cars), different field of view of sensors, etc.

For prediction or estimation, different dynamic models are used (based on the kind of object tracked; in this case, all of them are vehicles, and a subset of dynamic models can be used). For tracking, Kalman Filter is the first approach, because of its simplicity and good results. If needed, Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), or Particle Filter (PF) can be used. One Kalman Filter (or variant) is needed for each object, so a process managing creation and destruction of filters for new and obsolete objects was implemented.

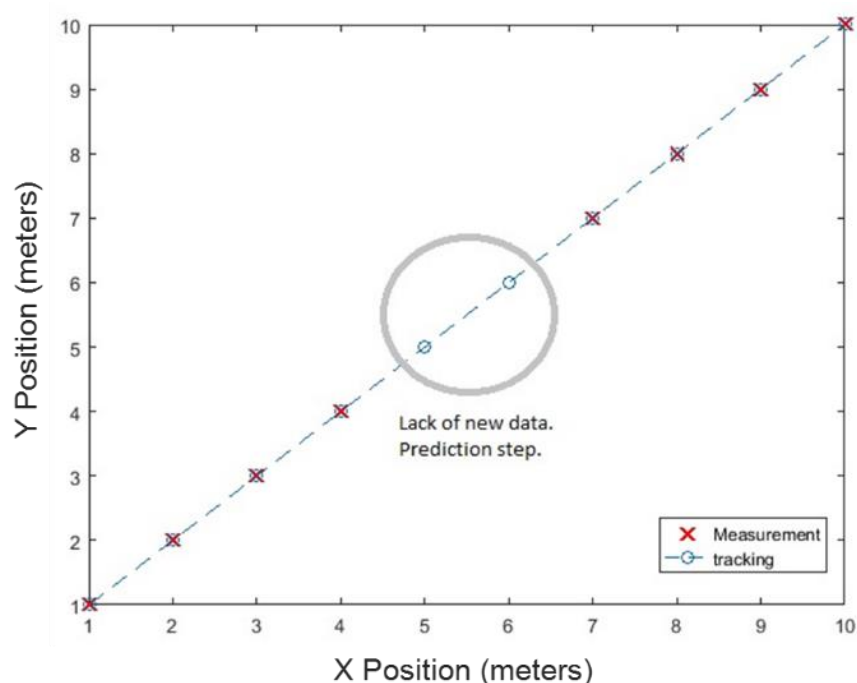


Figure 3.11: The Tracking step in the data fusion is used to filter out measurement outliers, such as missing or implausibly different samples.

The resulting vehicle descriptions are used for the planning of the lane merge maneuver as explained in Section 3.1.3.

3.1.3 Traffic Orchestrator

The Traffic Orchestrator has the purpose of devising individual maneuvers for connected cars in the lane merge demonstration, based on vehicle status information provided by the intelligent camera system and the connected vehicles. The unconnected vehicles influence the traffic flow in the vicinity, and thus need to be considered in the maneuver planning. In this context, the maneuver planning function considers the information provided by the Data Fusion function,

which contains position and movement information, and the vehicle connection status (connected or unconnected) to derive the corresponding actions.

In the demonstration, each driver receives a maneuver recommendation. However, the decision of accepting or refusing a recommendation is solely reliant on the OBU, Traffic Orchestrator and on-board HMI, removing driver input from the scenario. If the merge recommendation is rejected for any reason, including safety distances, speed/acceleration values or unhuman-like trajectory, the Traffic Orchestrator automatically handles a new cooperative maneuver for the scenario. This paves a way to instantly incorporate autonomous vehicles on the road, with minimal to zero changes when autonomous vehicles mature past prototype phase.

The Traffic Orchestrator receives information about every car that enters the scenario area it is orchestrating from the V2X Gateway. This information is processed to match the Deep Reinforcement Learning Algorithm (Deep Q-Network (DQN) and its derivatives) trained inputs, by separating the merging, closest following and preceding cars from the rest as illustrated in Figure 3.12. The reinforcement learning algorithm provides waypoints to the merging car to successfully merge in-between the two cars. The Traffic Orchestrator dynamically decides and assigns the closest cars, depending on the current road traffic. Therefore, the closest and following car may differ over multiple iterations of the same merging scenario.

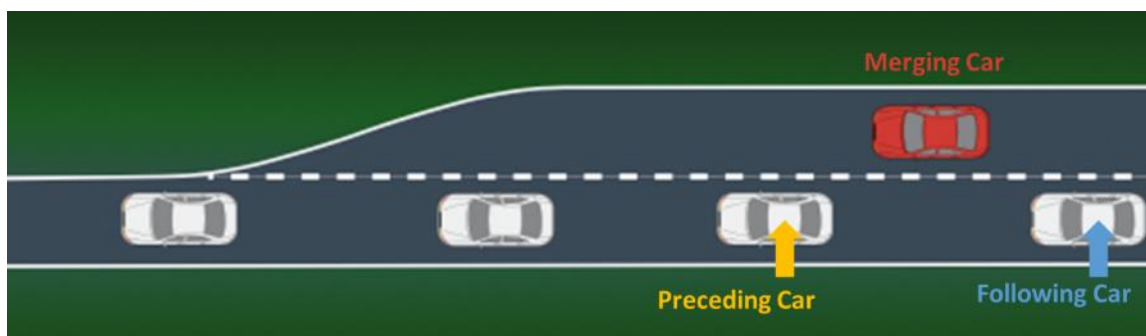


Figure 3.12: Vehicle states during maneuver planning.

The Traffic Orchestrator additionally provides safety actions to other autonomous cars in the merging scenario, this can be in form of slowing down cars to creating a larger merge gap, to merging other cars on neighboring lanes to make way.

In this work, the New Generation Simulation dataset [NGSIM18] was used in conjunction with real life data taken from the 5GCAR tests, to train and test the models. Collected in 2005 by the United States Federal Highway Administration, the NGSIM data set is one of the largest publicly available sources of driving data being widely studied in the literature. One immediate known limitation of the US101 data set is that vehicle trajectories were obtained from video-based analytics with a considerable amount of position noise. As a result, velocities obtained from differentiation with respect to position suffer from significant inaccuracies.

The data set was filtered by hand, in order to once again isolate the three important vehicles considered in the lane merge scenario, following, preceding and merging vehicle, removing all unrelated content.



The three aforementioned vehicles provide the state space for the reinforcement learning algorithm to calculate the most optimum waypoint for the merging vehicle to change lane. For the Deep Neural Network to provide high-quality output, live input data about the involved vehicles needs to be pre-processed in real time to shape the data for the expected input data of the DNN, and as well as to filter out disturbances.

Although it can be argued that with the use of 5G that mathematical models of a lane merge scenario would be provided with a longer time to carry out calculations. Having the Traffic Orchestrator incorporate Deep Neural Networks, specifically Dueling Deep Q-Network Architecture, allows for a greater generalization of the causes and effects that lead to a successful merge. This makes the Traffic Orchestrator adaptable to various lane merge scenarios it has not encountered before.

The Traffic Orchestrator implementation of the Deep Reinforcement Learning is unique in the sense that the environment processing of the merging scenario is carried out by the architecture at lower levels, leading to a reduction in neural network layers, optimizing power and memory consumption.

3.1.4 Optimized Message Delivery

As already described in the introduction of the Chapter, all demonstration-related components were instantiated in an edge cloud next to the test track. An OpenStack environment was used to instantiate both the user plane of the packet core, as well as virtual machines hosting docker containers with the various software components.

In order to isolate the use case traffic from artificial background traffic, two separate Serving Gateways (S-GWs) and PDN Gateways (P-GWs) as specified in [3GPP19-23401]) were used, and dedicated bearers with QCI 3 were triggered for V2X-related traffic, based on IP 5-tuples, as shown in Figure 3.13. The radio access is LTE-based, and uses 20 MHz in Band 7, specifically around 2645 MHz (downlink) and around 2525 MHz (uplink).

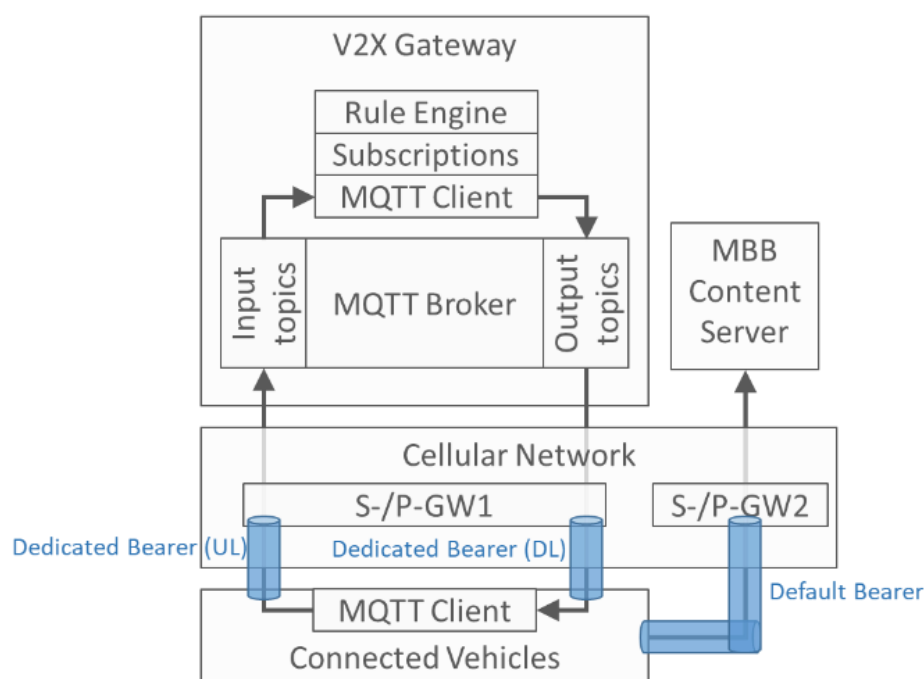


Figure 3.13: The V2X Gateway (and other demonstration-related components) is served via a dedicated user plane of the core network, and dedicated bearers are for prioritizing use case traffic.

The higher layers of the optimized message delivery addressed the scalability and deployment flexibility of such a setup. The Context-aware V2X Gateway and the OBUs communicate using the Message Queuing Telemetry Transport (MQTT), where the corresponding MQTT broker is functionally a part of the V2X Gateway. Furthermore, a set of custom functions in the V2X Gateway take care of implementing the context-specific routing and other 5GCAR custom functionality. This is achieved using context information in an outer layer of the messages, a subscription framework, and a rule engine sitting on top, used for defining that context-based application layer routing.

3.2 See-Through Sensor Sharing

For the see-through sensor sharing the architecture and key concepts can be separated into an application and a communication part. The Details on the application itself are stated out in Section 3.2.1 and the details on the communication in Section 3.2.2.

3.2.1 Details on Application

The see-through concept is built based on a four steps approach as shown in Figure 3.14.

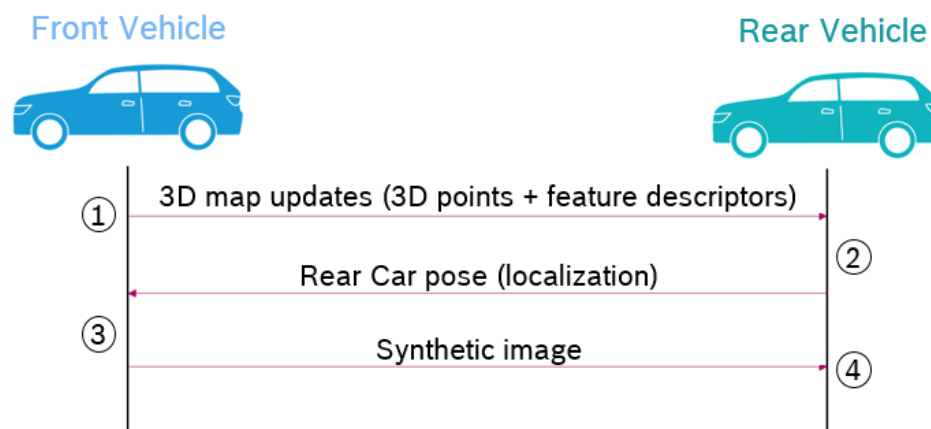


Figure 3.14: Steps to send the see-through overlay.

Step 1: Local 3D map generation

This step consists of the usage of visual odometry approach [AND08] to generate 3D map points and their feature descriptors at every new stereo image acquisition on the front vehicle. The generated 3D map points and the descriptors are transmitted regularly to the rear vehicle to update its 3D map of the environment.

Step 2: Localization of the rear vehicle

Using the new 3D map updates, the rear vehicle will update its 3D map and perform its localization in parallel, i.e., compute its relative pose localization, within this map. This information is transferred back to the front vehicle.

Step 3: Image synthesis

Based on the relative pose provided by the rear vehicle, the front vehicle computes a new synthetic image where the front vehicle is completely transparent from the rear vehicle's view. From this image, only a region of interest corresponding to the occluded part of the image is then transferred to the rear vehicle. The size of the image can be adapted based on the distance between the two vehicles. The closer the distance is, the higher is the number of pixels required to cover the occluded area. The synthetic image is then sent to the rear vehicle where it can be displayed for a human driver or used by computer vision algorithms to detect hidden objects in the case of an automated vehicle.

Step 4: Image stitching and display

An appropriate cropping of the received synthetic image is performed by the rear vehicle in order to stitch it onto the current image on the rear vehicle's front camera and display it overlaying the occluded area. This step is also beneficial if the vehicle is driven by an automated system as the stitching will tell the computer vision algorithms where exactly e.g. detected objects are located. The displaying of the see-through image is of course only necessary for a human driver but can also be interesting to show for passengers in an automated vehicle.

The video processing algorithms are developed based on the Robot Operating System (ROS) and the computational load is distributed between two computers running one on each vehicle and synchronized in time. All the processes corresponding to the 4 steps mentioned above are running in parallel. The computer of the front vehicle is responsible for running steps 1 and 3 related to the visual odometry for the 3D map generation and the image synthesis, while the computer on the rear vehicle is responsible for the execution of steps 2 and 4, related to the localization, stitching, and displaying of the image.

3.2.2 Details on the Connectivity

The communication link between the vehicles is based on a direct communication link which is not dependent on an infrastructure or cellular connectivity. The Modem is connected to the vehicle computer with Ethernet and the See-Through Application is running inside each vehicle computer, as depicted in Figure 3.15.

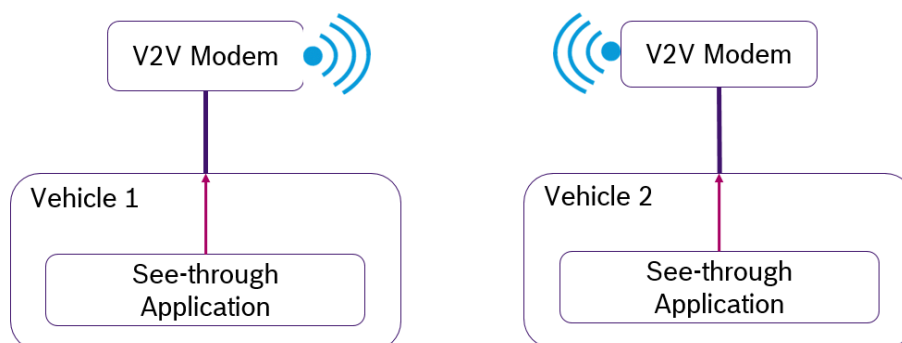


Figure 3.15: Connectivity between application and modem in the two vehicles.

The vehicle computers are located in the trunk of the vehicles used for the demonstration. The V2V prototype is located on one backseat of each vehicle (see Figure 3.16), because the V2V hardware could not fit in the trunk next to the built-in experimental hardware and a display was necessary to operate the system.



Figure 3.16: V2V hardware integrated into vehicle,

The V2V research platform implements the main concept of the 5G NR sidelink which is still under definition in 3GPP Release 16. The major radio layer parameters of it is listed in Table 3.1. The short TTI down to 0.25 ms lays the foundation of short latency connectivity while the 60 kHz subcarrier spacing allows for higher reliability against the Doppler effect in the highly mobile V2V channel. Both transmit and receive diversities with two antennas are enabled for effective reliability enhancement. Besides, the OTA synchronization enables the system to be independent of GNSS reception for the V2V link.

Table 3.1: Radio layer parameters of the 5G V2V research platform.

Parameters	Values
Carrier Frequency	3,692 MHz
Bandwidth	10 MHz
Tx Power	8 dBm
Multi-Antenna Mode	2 x 2 CDD at Tx and MRC at Rx
Slot Length	0.25 ms
Subcarrier Spacing	60 kHz
MCS	Adaptively selected between (16 QAM, 2/3 coding rate) and (64 QAM, 2/3 coding rate)
Tx / Rx Processing Delay	0.1 ms / 0.2 ms
Synchronization	OTA sync via PSS/SSS, optionally GNSS

The data interface for connecting the see-through sensor-sharing system supports two modes, the L3 IP tunneling mode and the L2 Ethernet transparent mode. Based on the experience from preparation tests performed earlier in Germany, the Ethernet transparent mode is selected for its better compatibility at the cost of transferring also the Ethernet header of 14 bytes over the sidelink.

3.3 Long Range Sensor Sharing

The connected vehicles share their own information, namely GPS/RTK based position, speed, acceleration and yaw rate, every 200 ms. The information about detected vehicles issued from the embedded sensors, a LiDAR in this precise use case, is sent in a dedicated message containing the speed and position of the detected vehicle, also every 200 ms. The context-aware V2X Gateway forwards this information to the Collision Detection algorithm, which predicts the trajectories of the different vehicles in the area every 100 ms and estimates a Probability of Collision based on these trajectories, every 200 ms. When the Probability of Collision reaches a given value, an alert message is sent to the connected vehicle. The overall infrastructure can be seen in Figure 3.17. The goal is to prevent an emergency brake when the incoming vehicle will be in line of sight for a human driver or even for the sensors of an autonomous vehicle, with this information given some seconds in advance. The driver or the autonomous vehicle can make a soft brake to prevent a collision before reaching the intersection.

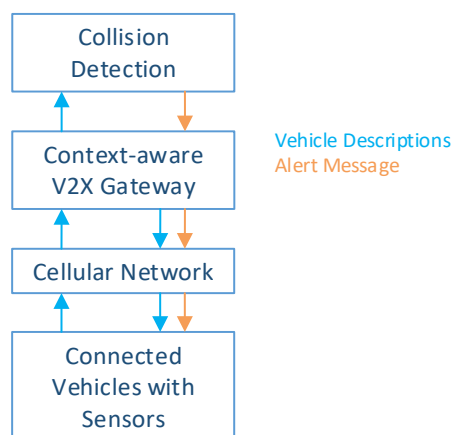


Figure 3.17: Generic description of the infrastructure.

3.3.1 Detection of Unconnected Vehicle

The first approach for detecting the unconnected vehicle (Vehicle 4, cf. Figure 2.10 in Section 2.3) at medium / long range made use of a camera mounted behind the windshield of Vehicle 3 and it would effectively detect Vehicle 4 and provide a rough estimation of its relative position with respect to Vehicle 3. This is how this use case was done at the beginning for the very first tests.

Bearing in mind that a precise positioning is crucial to determine an accurate time-to-collision in this use case, the vision-based approach was abandoned, and a LiDAR-based solution was implemented instead. The purpose of the demonstration is to send almost raw data to the infrastructure to simplify the on-board data processing. This is the reason why, no data fusion

technique was applied combining camera, LiDAR and other sensors before sending the data to the infrastructure.

A LiDAR system is based on a sensor that fires light beams in different directions and detects their reflection with integrated photodetectors. By a simple measurement comparing time of emission and time of arrival, the distance to an object in a certain direction can be inferred. With a high enough number of these detections, a point cloud is built, acting as a means of 3D sampled reconstruction of the environment. By properly processing this point cloud, objects can be detected with a highly accurate estimation of their relative position. Most sensors in the market provide a distance estimation accuracy of few centimeters. Tracked over time, the corresponding velocity is easily estimated.

Integration

The LiDAR sensor was integrated in the front bumper of a DS7 at approximately 45 cm from the ground, following sensor recommendations. A calibration was made to ensure a 0 degrees sensor pitch in order to have an optimal detection range. In order to avoid colliding with an already integrated radar sensor, it was shifted about 20 cm to the left of the car (as shown in Figure 3.18). As the sensor is not perfectly aligned with the longitudinal axis of the car, objects can appear rotated to the right or the left. In that case, yaw needs to be compensated so that the heading of the target is correctly calculated.



Figure 3.18: LiDAR integration in the front of the DS7.

LiDAR Model Used

In this case, the LiDAR sensor used was Scala 1, manufactured by Valeo. It is a time of flight LiDAR, i.e., distance is measured by calculating the time elapsed between the firing of a laser



beam and its return. Alternatively, to this measuring principle, other LiDAR sensors in the market use frequency modulation, which enables to estimate not only the distance but also the relative radial velocity of a target. But considering the characteristics of the use case and its requested precision, it was assumed that velocity could very well be calculated by tracking the position over time.

Scala 1 has one laser which fires pulsed beams periodically. These laser beams are vertically expanded by a lens so that they have their final 3.2 degrees aperture and are deflected by a rotating mirror which directs them horizontally. These beams come out of the sensor and travel until they hit a physical object which reflects them back but diffuses their energy almost in all possible directions. A part of this energy gets back to the sensor, is again reflected by the mirror and is concentrated by the receiving lens so that it fully hits 1 of its photodetectors (depending on the incidence vertical angle). These photodetectors are wired to a circuit which estimates the elapsed time between the pulses and outputs a voltage level according to the received pulsed energy, deciding if it justifies a warning about a physical target at that angle and distance or not.

LiDAR is a rapidly evolving technology. Generation by generation LiDAR sensors are becoming more accurate, reliable and above all, more abundant in the provided information volume. This sensor choice is probably aging quickly inside the market, but for this specific case there was a big confidence on its performance due to a long-lasting experience using it. The Scala 1 module is shown in Figure 3.19.

Its main characteristics are listed in Table 3.2.

Table 3.2: Characteristics of the LiDAR model used in the demonstration.

Characteristic	Value
Laser class	Class 1
Wavelength	905 nm
Technology	Time of flight, output of distance and echo pulse width
Horizontal field of view	145 degrees
Vertical field of view	3.2 degrees
Horizontal resolution	0.125 degrees between 15 and -15 degrees 0.25 degrees, otherwise.
Vertical resolution	0.8 degrees
Distance resolution	4 cm
Distance error	10 cm



Number of vertical layers	4 (2 permanent, 2 every second frame)
Scan rate	25 scans / s
Maximum range	around 350 m in the middle sectors
Number of echoes	3



Figure 3.19: The LiDAR module.

Let us note that the maximum range is theoretically given for highly reflective objects under ideal circumstances (highly reflective target, clear colors, low atmospheric attenuation, no destructive interference with neighbor reflections, etc.) and the best possible beam direction. Practically, the algorithm needs to have several reflections in a certain region to confirm that a detection is something physical and not mere sensor noise, and it's hard to have reliable detections over 120-140 meters away. In any case, this range is highly dependent on several factors that go beyond the scope of this work.

The LiDAR integrated in the front grille of the DS7 is shown in Figure 3.20.



Figure 3.20: Final LiDAR integration in the front grill.

In the next snapshots, Figure 3.21, the detections from the LiDAR are shown, as the target vehicle approaches with a first detection at around 120 m (in the following figure, the concentric dotted circles represent 30 m distance from the LiDAR). The speed estimation is calculated using the difference in time and position between two consecutive scans. This speed also helps to estimate the position of the tracked object in the next scan, which will lead to repeat the whole process in a cycle.

In this case, the object marked in the figure is consistently recognized as the target vehicle, which is very important as it means that the matching was successful across the different scans. Any error in this assignment would lead to difficulties in tracking the vehicle, which is essential for this use case.

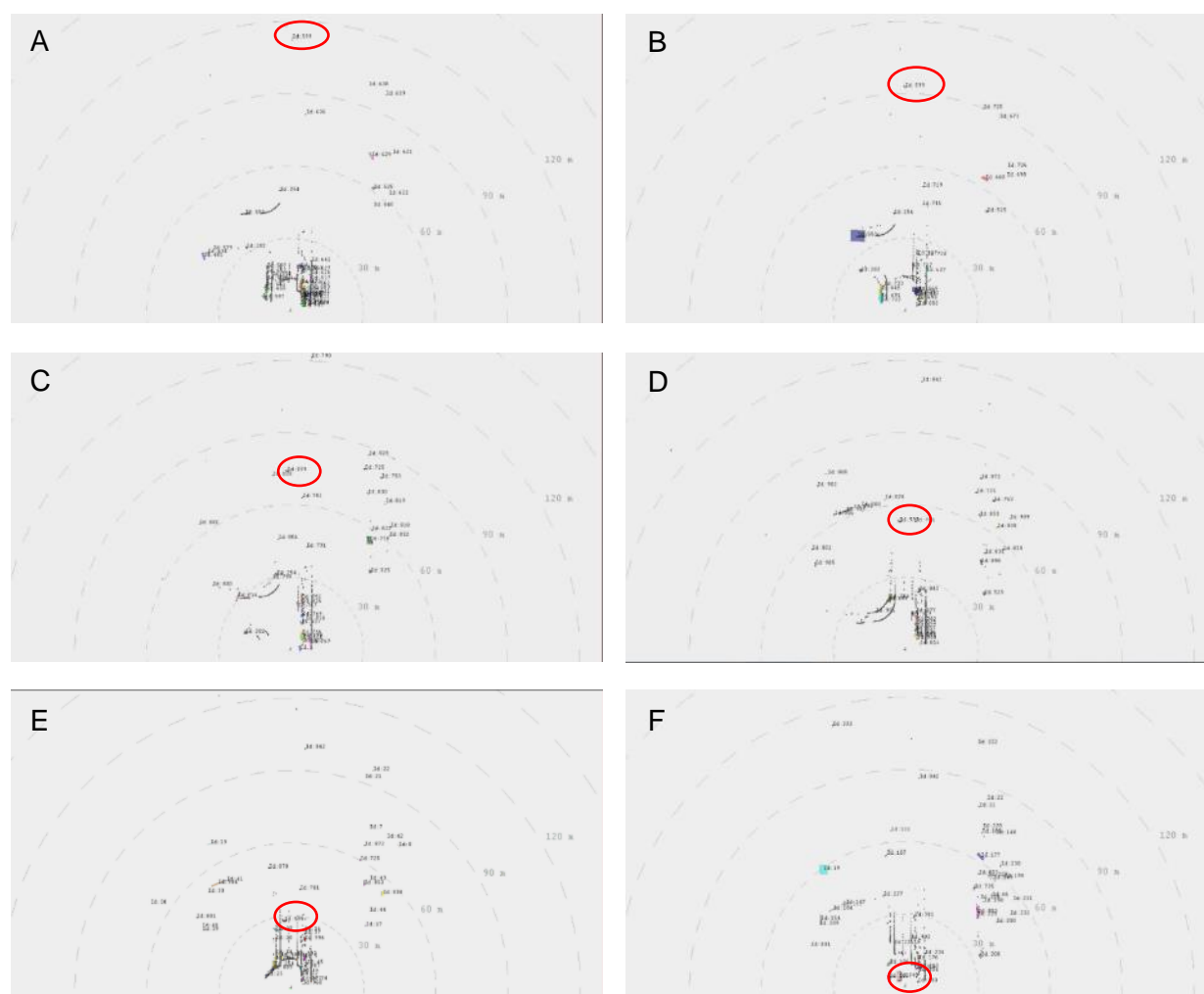


Figure 3.21: Detection and tracking of the target over the time (read from top left to bottom right, A to F).

The LiDAR communicates with a VBOX (see Annex B.1 for further information) by an Ethernet link, providing every 200 ms the relative position and heading of Vehicle 4 (or any other tracked object), as well as its speed and dimensions. Meanwhile, the VBOX has received Vehicle 3's GPS

information (latitude, longitude, timestamp, speed, heading, etc.) from the OBU. With this information, the VBOX is able to perform the following sequence of tasks:

- Convert Vehicle 4's position to the GPS reference system.
- Use Vehicle 3's GPS position to calculate Vehicle 4's absolute position (longitude, latitude).
- Transform Vehicle 4's relative heading (provided by LiDAR) into absolute heading using the GPS-provided Vehicle 3's absolute heading.

Add its own ID for the tracked object and complete the rest of the fields with the available information. VBOX is also in charge of sending it to the OBU, so that it is forwarded to the server. The period of the message is driven by the position update of Vehicle 3. So even if the LiDAR estimates the position and speed of the target every 40 ms, the message is sent every 200 ms, after acquiring the updated position of the ego vehicle as shown in Figure 3.22.

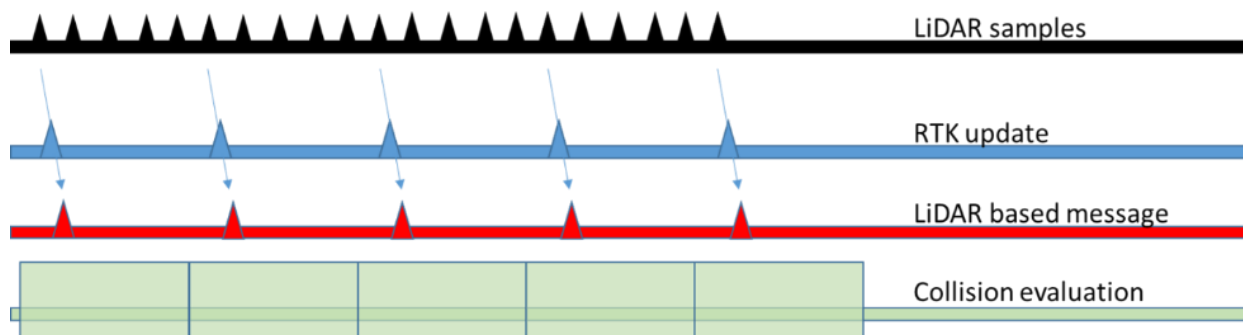


Figure 3.22: Periodicity of messages and collision evaluation.

3.3.2 Prediction of Collision

The state of the art of trajectory estimation of road users with particle filters and the prediction of collisions between road users have been intensively studied in 5GCAR [5GC18-D32, 5GC19-D33]. The concepts have been primarily developed for the vulnerable road user protection demonstration as described in more detail in Section 3.4. The same principles can be applied for the prediction of a collision between two vehicles as well, however the parameter configuration is different.

With a particle filter, the position of the vehicles, and optionally other state variables like speed and turn rate over time, can be tracked. The estimation of the future position of the vehicle is based on the constant velocity model. The probability density function of the state variables is represented by a swarm of particles. Important results have been published in [MMS+18].

Collision prediction is based on the anticipation of future states of the vehicles in the time window of a few seconds. The probability of a collision is the joint probability that the anticipated occupancy areas of two vehicles overlap in a certain point in time. Important parameters are the speed of the vehicles, their size, and the forecast time window. For a more detailed description we refer to Section 3.4.3. Compared to the vulnerable road user configuration in that Section, we observe the following differences:

- The future movement of a vehicle can be estimated much more accurately than the movement of a pedestrian or a cyclist. Consequently, the shape of the probability density function of the state variables, e.g., the position, has a more distinctive peak. Therefore, the number of particles can be reduced, which leads to less computational effort. In the demonstration the number of particles was set to 1,000.
- The speed of the vehicle is clearly higher than the speed of a pedestrian. This makes the collision prediction between two vehicles more difficult than the collision prediction between a vehicle and a vulnerable road user. Moreover, the size of the vehicle is larger. Finally, larger anticipated occupancy areas were got. In the demonstration, a vehicle size of 5 m x 2 m was assumed.
- The two effects described above partly compensate each other. Nevertheless, we adapted the collision probability threshold that must be exceeded before raising an alarm accordingly based on the perception of the car drivers during the demonstration preparation. In the demonstration the threshold was set to 40%.

3.4 Vulnerable Road User Protection

The overall architecture of the vulnerable road user protection demonstrator is shown in Figure 3.23. In the following we will describe the three main concepts implemented in the demonstrator and the required software and hardware components in detail.

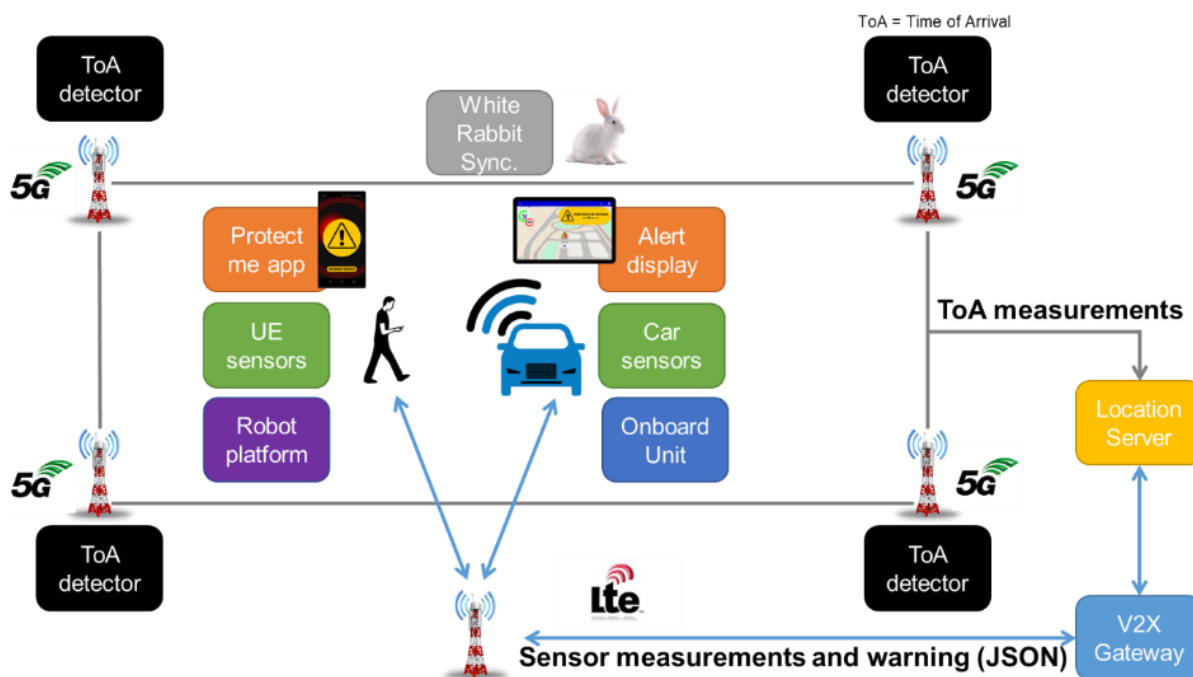


Figure 3.23: Overall architecture of the vulnerable road user protection demonstrator.

3.4.1 Real-Time Localization of Road Users with the 5G Radio Network

Real-time localization of road users is based on Uplink Time Difference Of Arrival (UTDOA). The basic principle is that transmitters in the vehicle and at the VRU periodically send pre-defined pseudo-noise sequences which are received at the base station antennas (gNB). Close to the antenna panel the main part is the Time of Arrival (ToA) detector that determines the ToA value for both users. The ToA detectors are connected to the location server provide new ToA measurements each 100 ms. The location server estimates the position of the two users from the ToAs. For the demonstration we used six passive cross-polarized antenna panels for the 5G radio band 3.6 GHz that are placed around the area of interest as shown in Figure 3.24. Four panels were mounted at stationary antenna masts, and two additional panels were installed at two temporary masts to improve the coverage in the area of interest.

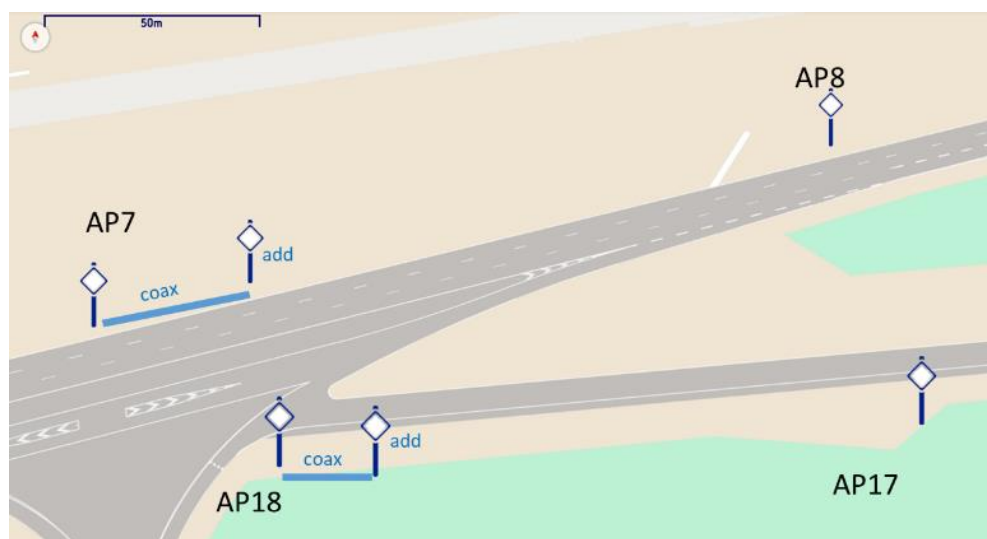


Figure 3.24: Positions of 5G receive antenna panels on the test track.

In the following we describe the components illustrated in Figure 3.23 which are required to implement the real-time localization.

5G Transmitters in the Car and the VRU

The car and the robot platform with the VRU dummy were equipped with transmitters of the pseudo-noise sequences. Figure 3.25 shows the block diagram of the transmitter and a photo of the hardware in a PSA vehicle. The 5G transmitters contain a signal generator in baseband and an upconverter unit for the specific 5G-NR band n78 [3GPP19-38.101] at 3.6 GHz, 3.628 GHz center frequency and 56 MHz channel bandwidth are allocated. Both users (vehicle and VRU) transmit the sequence alternately in two 50 ms slots. The synchronization of the two users is realized by GPS units that trigger the two 5G transmitters avoiding mutual interference.

For the 5G transmit units in the vehicle and at the VRU robot platform Uninterrupted Power Supply (UPS) is installed to guarantee permanent voltage supply and reliable transmission for around two hours, also in case the vehicle 12 V power supply is not always available.

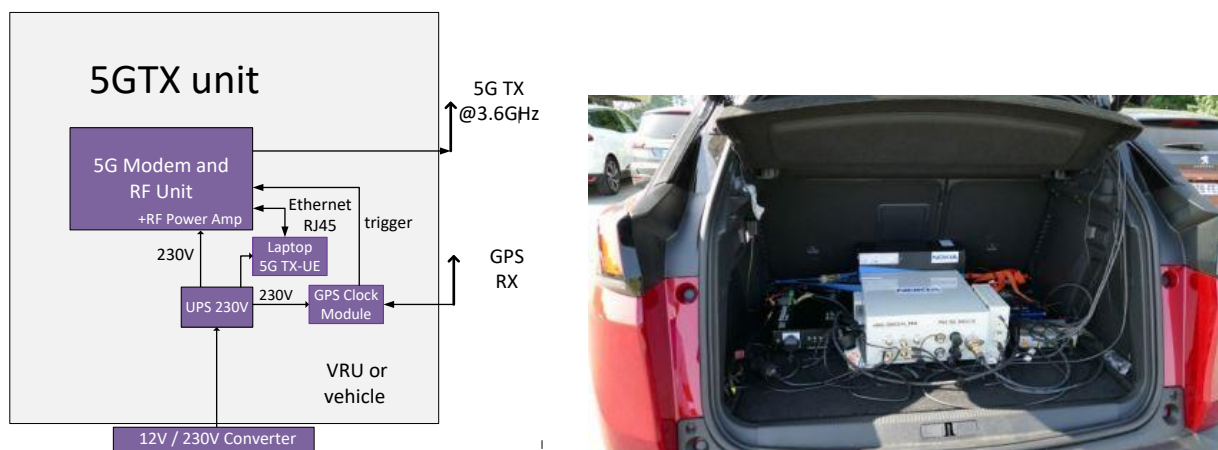


Figure 3.25: Block diagram and picture of 5GTX unit.

5G Receivers with ToA Detectors

The gNBs are located close to the antennas at the bottom of each stationary mast as shown in Figure 3.26. The passive antenna panel is not shown and on the top of the mast connected via cables to this gNB. The processing unit in the gNB receives the pseudo-noise sequences and correlates the received signal with the known transmitted signal. The main peak of the correlation output in Figure 3.26 represents the dominant line-of-sight propagation path. From this correlation peak the gNB determine the ToA value. As the ToA depends on the distance between transmitter and receiver, we implicitly measure the distances between the two users and all antenna panels. The measurements are provided to the location server via fibers. In next section, we describe how we compensate the additional offset due to processing time, cables and connectors.

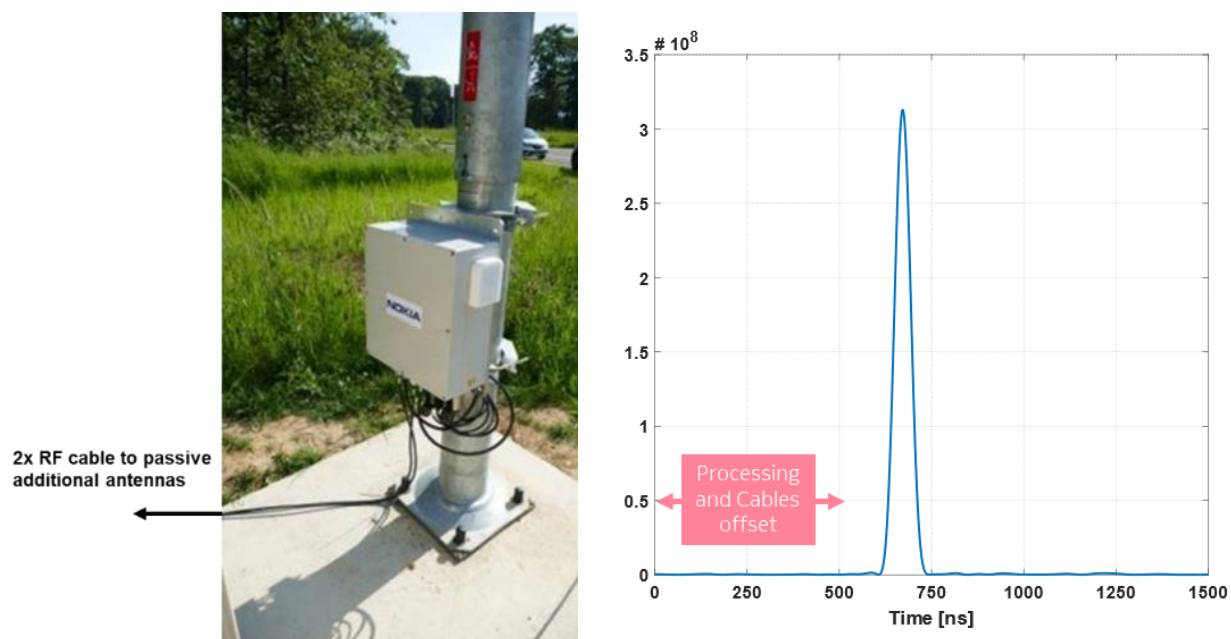


Figure 3.26: gNB at access point AP17 with ToA detector and correlation peak.



Synchronization with White Rabbit

One important enabler for the 5GCAR VRU demonstrator is the tight synchronization of gNBs through the White Rabbit protocol [WR09]. All gNBs are connected through multiple fibers and one of these fiber links is utilized for the synchronization of their clocks to below 1 ns offset with White Rabbit. Two-way exchange of precision time protocol messages allows the adjustment of clock phase and offset. The link delays are precisely known through hardware timestamps.

Location Server

The Location Server gets messages from overall twelve antenna signals (six cross-polarized antenna panels) provided by four gNBs through a UDP interface. Each message consists of a timestamp, the ToA, the peak amplitude, the average amplitude, and the user ID.

After some post processing to get rid of constant offsets in the ToA and to identify unreliable antenna signals we estimate the position of the users. We did not implement the concept for Non-Line-of-Sight (NLOS) compensation described in [5GC18-D32, 5GC19-D33] because the propagation conditions in the demonstration area were clearly dominated by Line-of-Sight (LOS). The single peak in Figure 3.26 is a strong indicator for pure LOS. Instead we used a Fair-Time Difference of Arrival (F-TDoA) method.

Finally, we have position estimates based on 5G localization with an update rate of 100 ms for the vehicle and the VRU available at the location server.

3.4.2 Trajectory Estimation with Sensor Fusion

The initial position estimate based on 5G localization as described in Section 3.4.1 can be significantly improved if sensor measurements from the vehicle and the VRU are considered as well. The basic concept is the application of a motion model as described in more detail in [5GC18-D32, 5GC19-D33]. For the demonstration we have implemented the Constant Velocity (CV) model, i.e. we track the two state variables position and velocity. The underlying estimation method is the Particle Filter (PF) [AMG+02]. A swarm of particles represents the probability density function of the state variables.

In the following we describe the components in Figure 3.23 which are required to implement this second concept.

The demonstration was performed with vehicles from Volvo and PSA. A detailed description of the PSA and Volvo cars and the specification of the available sensors can be found in the Annexes B and C for the PSA vehicle and Volvo vehicle in Annex D. For the demonstration we utilized only the speed with an update rate of 10 Hz.

The On-Board Unit (OBU) is described in detail in Section 3.5.1 and was used for vehicle and robot platform for the evaluation phase and the final demonstration. The OBU, equipped with an internal GNSS module with an update rate of 1 Hz and an external GNSS RTK module with an update rate of 5 Hz provided trace recording in the algorithm evaluation and measurement campaigns in May and June of 2019.



Trajectory estimation is done in the location server using PFs, and the motion models are implemented in MATLAB and Python. For the demonstration we used 1,000 particles. The inputs are 5G-based position estimates (over the 5G network), GPS measurements from the car and the VRU, as well as the speed of the car (from the V2X Gateway). Output is a more accurate and stable estimation of the position.

3.4.3 Collision Prediction and Release of Warning Messages

While the trajectory estimation described in Section 3.4.2 is based on tracking of the state variables by applying a motion model and fusing of measurements (5G-based localization, GPS and vehicle speed) to get an update of the current position, the collision prediction module tries to anticipate the future movement of the road users in the next few seconds. The theoretical background is described in [5GC19-D33]. The probabilistic approach that we implemented in the demonstrator adopts the motion model introduced in Section 3.4.2 to determine a mean value of the anticipated position in a certain point of time in the future. The uncertainty is represented by a probability density function around this mean value. The shape of this function is mainly influenced by three parameters:

- The time window: The longer we look into the future the less reliable is the prediction. Consequently, the probability density function of the future position will become flatter and less distinct.
- The speed of the road users: A relative increase or decrease of the speed will have more impact on the future position, the higher the speed is. Therefore, we will observe a flattening of the probability density function in the direction of the movement with increasing speed.
- The area that is covered by the user: It is intuitively clear that a big road user (e.g., a truck) will lead to a higher collision probability than a small road user (e.g., a pedestrian).

The probability of a collision is then simply the joint probability that the anticipated occupancy areas of the vehicle and the VRU overlap in a certain point in time.

A warning message is released when the collision probability exceeds a certain threshold. Our implementation observes several time windows in parallel and triggers an alarm based on the combination of collision probabilities for these time windows. The warning is sent to the V2X Gateway, where it is forwarded to both road users.

In the following we describe the components in Figure 3.23 which are required to implement this third concept.

Location Server

In the location server the collision prediction algorithms as described above are implemented in MATLAB and Python. For the demonstration we consider time windows between 0.5 and 5 s. We assume an area of 1 m x 1 m for the VRU, and 5 m x 2 m for the car. The alarm threshold was set to 60%.

VRU Alert Display

Upon reception of an alert message by the OBU, it is transmitted to the on-board HMI and displayed to the driver. It consists of two elements: a warning icon on the map showing the location of the danger, and a bigger alert message indicating the distance to the danger, as can be seen on Figure 3.27. Additionally, a double beep tone is played to attract the driver's attention. More information on the generic HMI can be found in Section 3.5.3.



Figure 3.27: Pedestrian on the road alert in VRU demonstration.

Protect-Me App

The “Protect-Me” Android application was developed specifically for the VRU Protection demonstration. The vulnerable user is equipped with a smartphone featuring this dedicated application that connects to the V2X Gateway. This application uses the GNSS module of the phone to create location messages that are sent to the Gateway. The different states of the Protect Me app that the VRU can chose are shown in Figure 3.28. Users can indicate whether they are walking or using a bicycle, and they can increase the position update rate from 1 Hz to 5 Hz when they are on the road.

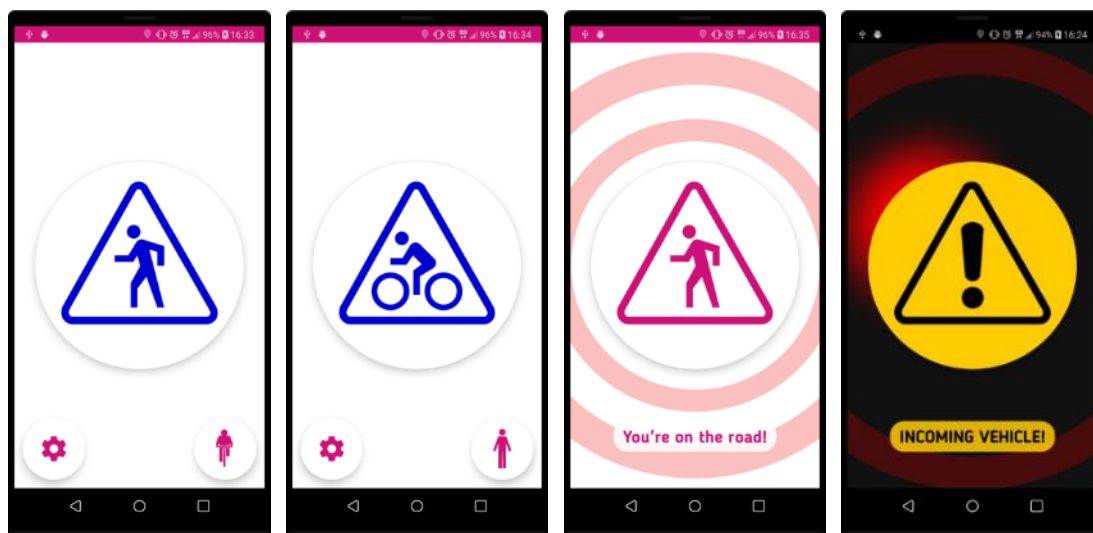


Figure 3.28: VRU app – pedestrian and cyclist mode, on the road mode and vehicle alert.

When the Location Server detects a collision risk, the application receives an alert message through the Gateway, triggering a strong alarm and indicating the direction of the incoming danger.

3.4.4 Execution of the Demonstration

Apart from the proof-of-concept of 5G radio-based positioning, the main target of the demonstration is to showcase the collision warning system. For that we have defined one drive route for the car, and two different routes for the movable robot platform with the dummy representing the VRU as non-human device. These different two routes are selected and used for the final demonstration. The route for a critical situation will stress the algorithm and the system will determine a collision probability above the threshold and release the warning message accordingly; the second route is for a non-critical situation without warning message generation. In the following we describe the exact execution of both scenarios.

Critical scenario: As shown in Figure 3.29, the red vehicle approaches a zebra crossing with around 40 km/h. At the same time the robot platform with the VRU dummy appears behind the parked white vehicle and crosses the street with walking speed around 3 km/h. A warning message appears on the HMI display in the Volvo vehicle (Figure 3.30), For the PSA a different integrated HMI was used, see Figure 3.31. The VRU smartphone generates an incoming alert messages, see Figure 3.28. and the vehicle stops at the crossing due to alert message indication in real time and avoid upcoming accidents successfully.

Non-critical scenario: The vehicle again approaches a zebra crossing with around 40 km/h. At the same time the robot platform with the VRU dummy moves towards the crossing area but stops at the right time. In this case no warning message will appear, and the vehicle drives through the crossing area without stopping.

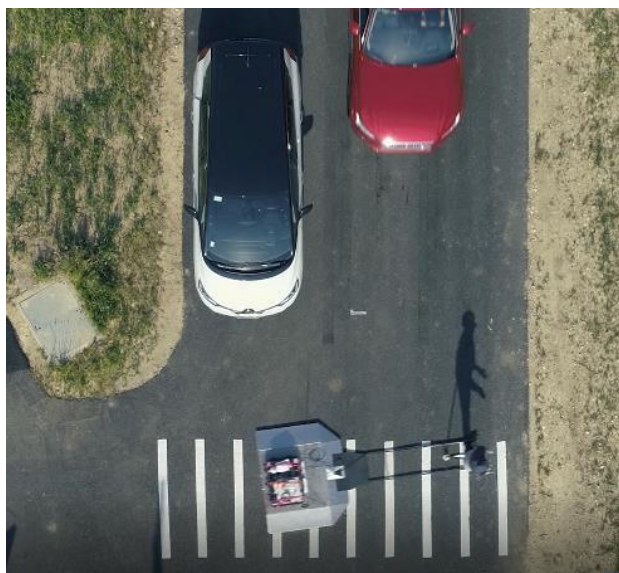


Figure 3.29: Crossing area of red vehicle VRU robot platform.



Figure 3.30: HMI display for Volvo vehicle with warning message.



Figure 3.31: Embedded HMI in PSA vehicle with alert message.

3.5 In-Vehicle Setup

In this section, the general in-vehicle setup is described, covering the connectivity unit (OBU), a tablet HMI used in the demonstrations, and the integration of these components with the in-vehicle systems of the used Volvo and PSA vehicles. The setup is described in Figure 3.32, and the key components are described in the following subsections.

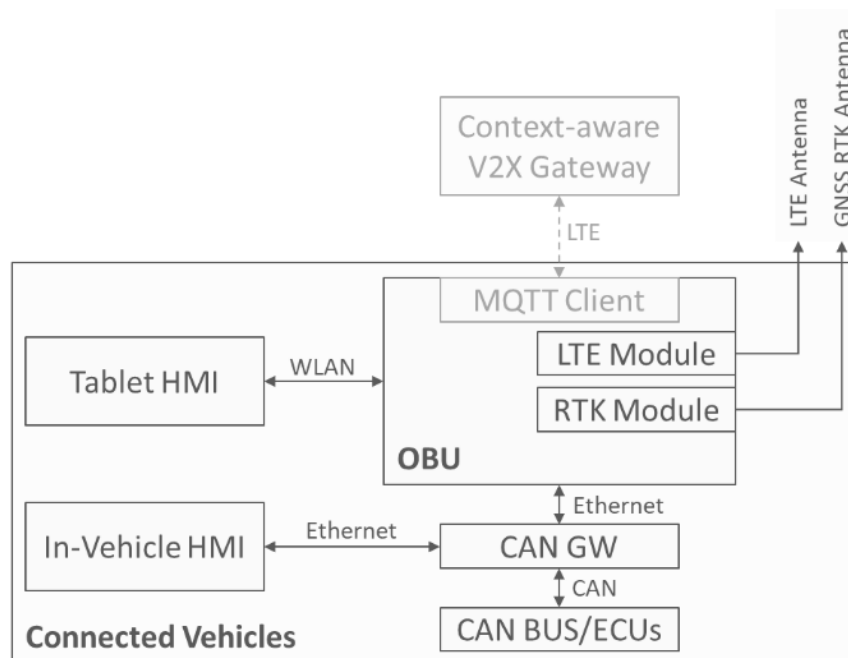


Figure 3.32: In-vehicle setup, including the different relevant components and interfaces. In in-vehicle HMI is only integrated in the PSA vehicles.

3.5.1 On-Board Unit

The On-Board Unit (OBU) is the interface between the car and the radio network (shown in Figure 3.33). It is based on a Sierra Wireless AirPrime MC Series Development Kit and uses a Sierra Wireless AirPrime MC7304 mPCI card as modem, combined with a magnetic rooftop antenna. The OBU is equipped with an internal GNSS chipset (uBlox m8n) with an update rate of 1 Hz, and an external GNSS RTK module (Emlid Reach M+ RTK) with an update rate of 5 Hz. Also, the OBU receives sensor information from the in-vehicle system, such as speed, acceleration, and yaw rate.

When the OBU reads a new GNSS measurement, it creates a vehicle status message, using the last data it received from the CAN Gateway. This message is then sent to the V2X Gateway.



Figure 3.33: OBU in the vehicles, connected to the in-vehicle system, a rooftop antenna, and a GNSS antenna.

3.5.2 Vehicle Sensors

The demonstration was shown with vehicles from Volvo and PSA. In both cases, a CAN gateway was used to read speed, acceleration and yaw-rate from the in-vehicle sensors, and to provide static information on size and color of the vehicle.

For the Volvo vehicles, the live information was provided with a periodicity of 100 Hz and sent to the OBU. To achieve this, two applications (signalbroker and car5g, a signalbroker client) are deployed in a RaspberryPi. The signalbroker acts as a server with the access to the CAN buses. After startup, the signalbroker reads a local JSON interface definition file to learn where to fetch the data when it receives a signal subscription from a client. The car5g application acts as a client, connects to the signalbroker and subscribes to the signals it would like to receive. It also automatically connects to the OBU after start-up and forwards the signals received to the. The software running in the Volvo vehicle has been using erlang/elixir and the corresponding code is available on GitHub at [VOLVO-BROKER]. More information is provided in Annex D.

For the PSA vehicles, the live information is collected from the in-vehicle system at 25 - 100 Hz and sent to the OBU with an update rate of 10 Hz. Beyond collecting and forwarding information from the CAN bus of the vehicle, the CAN gateway forwards relevant information to the main HMI of the vehicle. More information is provided in Annex B.

3.5.3 Tablet HMI

The tablet HMI is based on an Android application previously developed for other V2X projects and extended to handle 5GCAR use cases.

When used as an on-board HMI and as depicted on Figure 3.34, the tablet is connected to the OBU over a dedicated WiFi, and the application opens a TCP socket to receive messages from the OBU, such as its position, other vehicles and road users' positions, alert messages and so on.

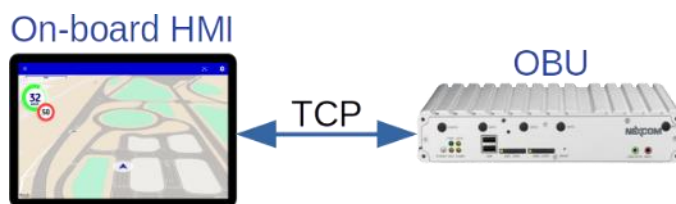


Figure 3.34: On-board tablet HMI connectivity setup.

The home screen of the application (Figure 3.35) allows drivers to select the use case that they want to demonstrate, filtering out the messages that are out of scope.

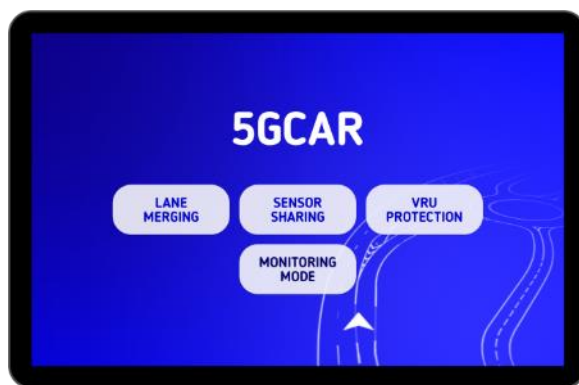


Figure 3.35: Home screen of the tablet HMI.

The main aspect of the HMI is its map-based navigation system style, depicting ego vehicle position and speed, alongside other road users' positions and alerts as they are received from the OBU.

To be able to illustrate the accuracy of the solutions developed within 5GCAR, a high-definition and accurately positioned overlay of the UTAC TEQMO test track has been added to the map. This can be seen on Figure 3.36.



Figure 3.36: Tablet HMI on-board view.

3.6 Tools

For supporting the development, integration, and testing of several demonstrations, various tools were developed and proved very useful. Some of these tools are described in the following.

3.6.1 Cloud Deployment

To facilitate integration and testing of the different demonstration components before the field tests, a cloud deployment was set up, where the core parts of the 5GCAR messaging system were hosted. All other components could then make use of this system 24/7, to accelerate the development and facilitate the cooperation between the partners. Figure 3.37 shows a picture highlighting the deployment of the different components, and the location of different test sites.

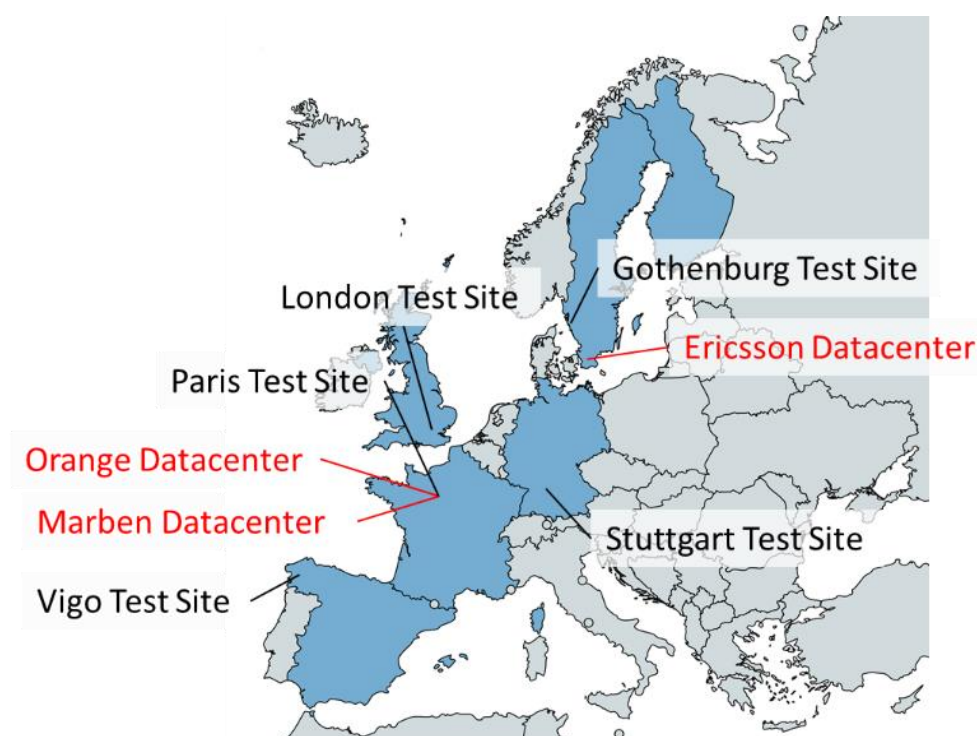


Figure 3.37: The core parts of the 5GCAR messaging framework was hosted in a set of data centers, and used from different test sites, for testing and integration work before the field tests.

While the Ericsson Cloud and the Marben Cloud only hosted manually updated instances of the V2X Gateway and the Dynamic Map, the Orange Cloud (called Flexible Engine) implemented a light but complete Continuous Integration and Continuous Delivery infrastructure for the other components, as illustrated in Figure 3.38.

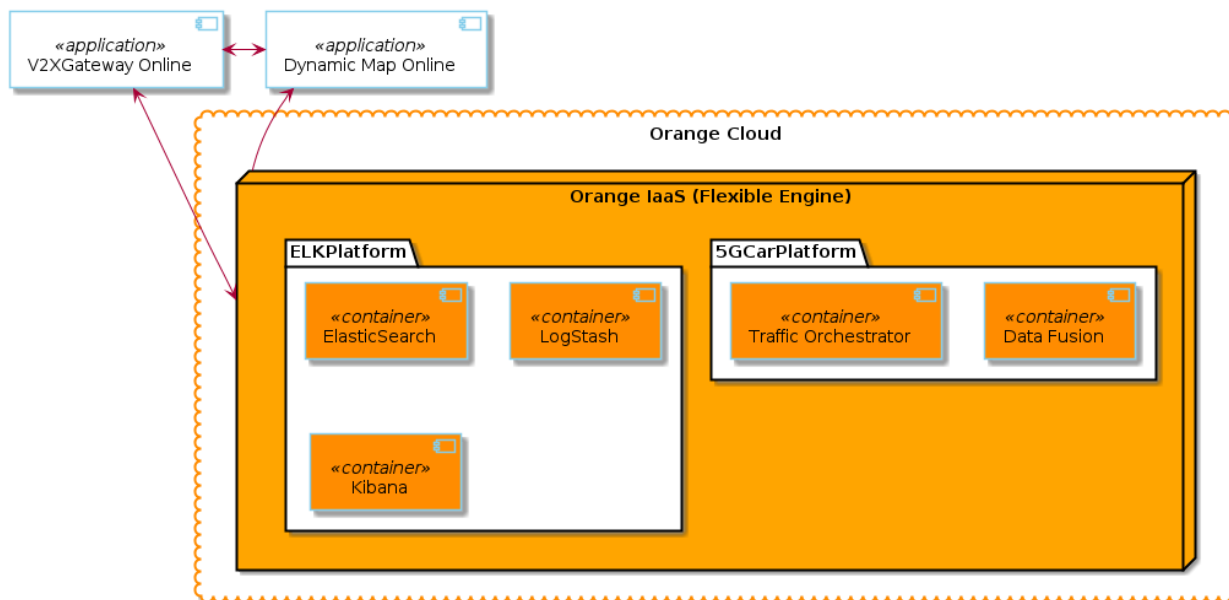


Figure 3.38: Integration over Internet using the online application or the nightly built container.

The complete regeneration was handled by Ansible scripts deploying virtual machines and then orchestrating containers through the docker compose manager.

For the lane merge coordination use case, we succeeded to obtain a fully operational exchange system, thus achieving a good level of robustness and maturity of the individual components before the integration tests and the event on the UTAC TEQMO test track.

3.6.2 Data Injector

The data injector was used to simulate live data, for the purpose of testing updates and improvements of the used components, as well as reproduce issues observed in other tests. It provides a Web GUI to select a test session, and specify the ingest point and parameters, as shown in Figure 3.39.



Data Fusion

This function will send data from files:

- [MessageTest_1_version_1_2.txt](#)
- [Monthery_1_trajectory_log_TEQMO_JSON_1.0.txt](#)
- [Monthery_2_lane_merge_01_ordered.txt](#)
- [Monthery_3_lane_merge_with_cam.txt](#)
- [Monthery_5_6_1.txt](#)
- [Monthery_5_8_1.txt](#)
- [Monthery_5_8_2.txt](#)
- [Monthery_5_8_3.txt](#)
- [Monthery_5_8_4.txt](#)
- [Monthery_5_cam_8_1.txt](#)
- [Monthery_5_cam_8_2.txt](#)
- [Vigo_1_VISCODA_visual_obu1.txt](#)
- [Vigo_1_VISCODA_visual_obu3.txt](#)
- [Vigo_1_VISCODA_visual_obu4.txt](#)
- [Vigo_1_VISCODA_visual_obu5.txt](#)
- [Vigo_1_obu10-00011.log](#)
- [Vigo_1_obu9-00011.log](#)
- [coop_perc_1_cleaned.csv](#)
- [coop_perc_2_cleaned.csv](#)
- [coop_perc_3_cleaned.csv](#)
- [coop_perc_4_cleaned.csv](#)
- [coop_perc_5_cleaned.csv](#)
- [notifications-vigo.txt](#)
- [notifications.txt](#)
- [notifications_v1_2_test.txt](#)

It send them to the URI provided:

host	<input type="text" value="90.84.192.152"/>	port	<input type="text" value="9203"/>	protocol	<input type="text" value="UDP"/>
test suite	<input type="text" value="Monthery 05 2019 [6] [1]"/>	number of lines (0 for all)	<input type="text" value="1000"/>		
method	<input type="text" value="extrapolate timestamp to now"/>				
<input type="button" value="send"/>					

Log received

Figure 3.39: Road User descriptions injector simulating a live input.

It also implements enhanced mechanisms from the messaging framework such as subscription mechanisms and sending/receiving Maneuver Recommendations and Maneuver Feedbacks. Finally, the data injector can inject data at different parts of the framework, e.g. in front of the V2X Gateway, the Data Fusion or the Traffic Orchestrator.

Aside from sending the data to other components, it can also be used to visualize the data contained in the locks, thus facilitating identification of malfunctioning in the components that produced the logs. An example is shown in Figure 3.40.

Road User description visualizer

Select a file and click on the play button to visualize how the cars are moving on the map.
Road user descriptions are read in order of appearance and not respecting their timestamp.

lane_merge_3_merged_2019-02-05T15-04-45.json

Info

- icon 1 is RU with id 10102
- icon 2 is RU with id 10101
- icon 3 is RU with id OBU5
- icon 4 is RU with id RSC01
- icon 5 is RU with id OBU11
- icon 6 is RU with id RSC05
- icon 7 is RU with id RSC07
- icon 8 is RU with id OBU12
- icon 9 is RU with id RSC08

Icons made by Roundicons from www.flaticon.com is licensed by CC 3.0 BY

Figure 3.40: Road User descriptions injector simulating a live map monitoring.

3.6.3 Tablet Monitoring Mode

Context

During the first test session in UTAC in February 2019, it appeared that it was very difficult for the people managing the server side of the system to comprehend what was happening on the test track and to detect issues from the server room. Meanwhile in the cars equipped with the tablet HMI, it was easy for drivers to spot unexpected behaviors, but complicated to report it and describe it to people in the server room.

Following this test session, it was decided to develop a Monitoring Mode in the tablet app, which would connect directly to the V2X Gateway through MQTT without the need of an OBU (available in limited number) as depicted in Figure 3.41, and display road users, alerts and trajectories on the map.

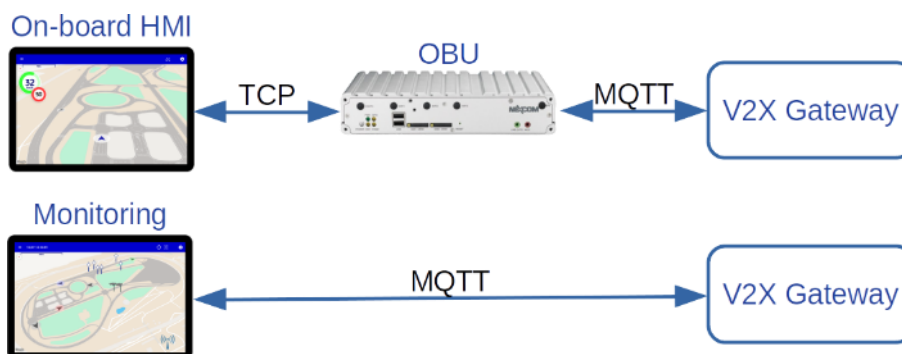


Figure 3.41: Difference of connectivity to V2X Gateway between on-board HMI and Monitoring Mode.

Overview

In addition to the three use cases used for the on-board HMI, Monitoring Mode can be selected on the home screen of the tablet application (see Figure 3.35). This mode can either be used on-board vehicles or remotely from a server room.

It is possible to choose the MQTT server address, to easily allow switching between over-the-internet test servers and on-site demonstration servers.

Only in Monitoring Mode, it is possible to filter messages by MQTT topic, as use case specific topics were used in the project. Road users being published on different topics are displayed with different colors on the map (original road user color for dynamic map, yellow for VRU protection, orange for sensor sharing, red for lane merge, black for direct camera output).

As shown in Figure 3.42 and similar to what is displayed in the vehicles using the tablet HMI, road users are represented by arrows with a slight tint hinting their color. A high definition image of the TEQMO test track has been added to the map, with points of interest such as the mast from Orange hosting LTE antennas, the Nokia 5G antennas used in the VRU demonstration and the Roadside Cameras used in the lane merge demonstration.



Figure 3.42: Monitoring Mode TEQMO view with road users and points of interest.

Selecting a road user allows to get some information, i.e. its unique identifier, road user type, speed and last update time (Figure 3.43).

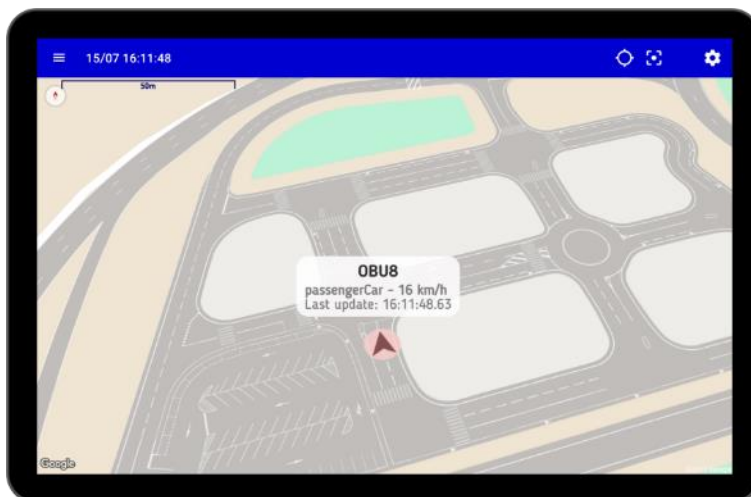


Figure 3.43: Road user information.

Finally, to ease troubleshooting and further help understanding what is happening from a road user's perspective, it is possible to simulate its view as it would be shown on the on-board HMI (Figure 3.44). In this mode, only alert messages and trajectories sent to this specific user are being displayed.



Figure 3.44: Simulation of a road user in monitoring mode.

This visual tool really helped the team to understand what was happening on the track in real time and to troubleshoot much faster. It also turned out useful to showcase and explain the use cases from the conference room on demonstration day.

3.6.4 KPI Evaluation Platform

In a distributed system with as many software components as used in 5GCAR, it can be a major overhead to collect logs for post-processing. To facilitate this, a platform was used that collects logs from all components and can be used to search, analyze, and visualize the combined information from all the components. The purpose of this is two-fold:

1. To address the KPIs defined in [5GC18-D51], certain measurements have to be performed.
2. It's convenient to check the health of the overall system, by verifying that the logs show the expected performance (e.g. processing duration by each component, and loss of messages between or inside components).

We chose the Elastic Stack solution, with Elasticsearch for storage and search, Logstash for data ingestion and transformation and Kibana for visualization. This allowed us to assess our KPIs on the runtime.

Each component sends logs in order to centralize, save, share and merge them with the other components, using any format. A common log format is used for some messages across the use cases, where each component sends a timestamp of ingress and egress, associated with a unique message ID and other differentiating features.

By combining this information from all components, we were able to compute delay, number of messages sent, received, lost or merged. Furthermore, this information could be analyzed and presented in different ways (over time, mean, min, max, percentiles, etc.), and filtered by different criteria (time window, vehicle ID, etc.), in order to track down unexpected behavior, and understand where certain observed characteristics originate from.

Figure 3.45 illustrates the various components sending logs to LogStash, as the ingest point of the elastic stack.

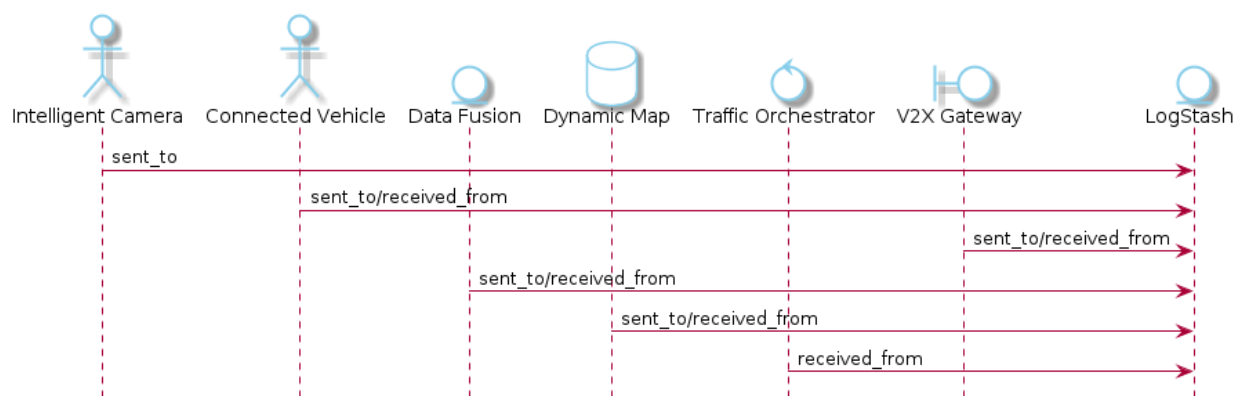


Figure 3.45: Each component sends and receives Road User descriptions and sends corresponding logs to LogStash.

In the end, we were able to follow the exchanges in real time and to evaluate them after the trials, as shown in Figure 3.46.

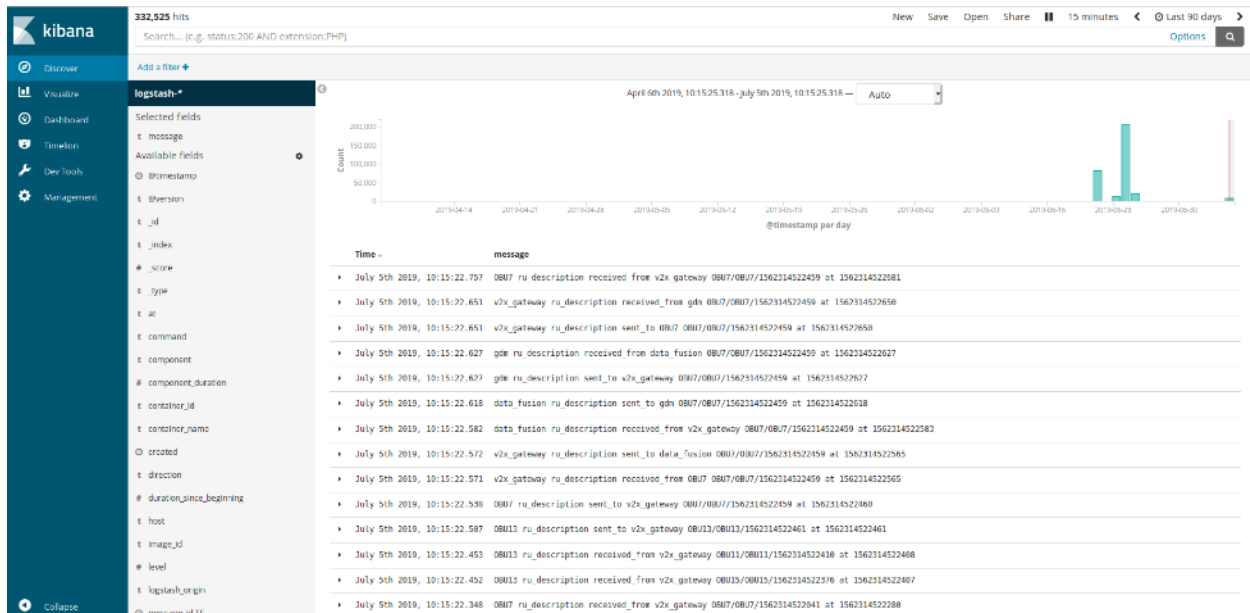


Figure 3.46: Discover page used to follow the logs from any component.

During the trials, we monitored the global system through visualizations clustered in dashboards, where Figure 3.47 shows an example.

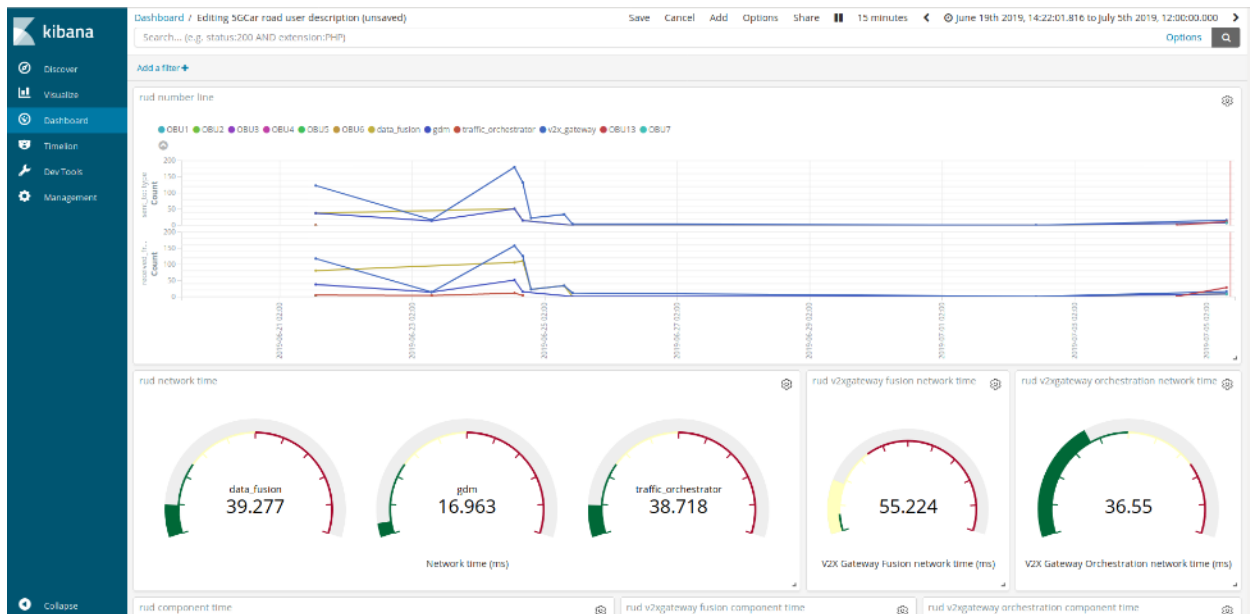


Figure 3.47: Dashboard used to monitor the Road User description number and online network times.

Here, we can see eight OBUs and four software applications communicating via public internet, with correspondingly high latencies.



3.6.5 Post-Processing with MATLAB for Parameter Tuning

During the preparation phase of the demonstration we evaluated ToA-samples measured at the twelve antennas. The transmitter was placed at known reference points with line-of-sight to the receive antennas. With this setting we could compare the expected ToA, which corresponds to the true propagation time depending on the distance between transmitter and receiver with the measured ToA. The difference between both values is a constant offset comprising delays through cables and connectors, processing time, and a clock offset between transmitter and receiver. This offset was determined offline and considered during the real-time localization during the demonstration.

We have found out that the ToA measured at some of the antennas are subject to significant fluctuations, whereas other antennas showed a more stable behavior. The main reason for that is the polarization of the transmitted signal. As we used cross-polarized antenna panels, the received power at some antennas was very weak. We could improve the positioning accuracy with post processing of the measurements by considering only 10 out of 12 antenna signals. The right selection of antenna signals was afterwards adopted for the real-time positioning.

A fine tuning of the collision prediction parameters (reasonable time windows in combination with collision probability thresholds) was done with post processing of a set of critical and non-critical trajectories. In the real-time demonstration we applied these optimized parameters.

3.6.6 Hardware Tools for VRU Positioning System

Spectrum analyzer: During the preparation of the demonstration we used a spectrum analyzer to ensure that the two road users are synchronized, and their alternating transmissions do not overlap. Otherwise mutual interference would mitigate the ToA measurements and accurate positioning would not be possible.

GPS system: We determined the precise location of the six antenna panels using a Leica Zeno20 measurement equipment. Knowledge of the exact 3D-location of the antenna panels is required for the 5G radio-based positioning. On the street marked reference points were measured to know a set of ground truth positions. These were utilized to determine the achievable accuracy of 5G radio-based positioning.

4 Demonstration Evaluation

The coverage of the cellular network on the test track is provided from an antenna mast next to the test track, as shown in the bottom right corner of Figure 4.1. As can be seen, the full track is in coverage: Table 7.3.3.1 in [3GPP19-36521] mandates -92 dBm reference sensitivity for UEs, corresponding to -119.8 dBm Reference Signal Received Power (RSRP), which is fulfilled on all measurement points. Furthermore, the RSRP is quite good in most parts of the test track.

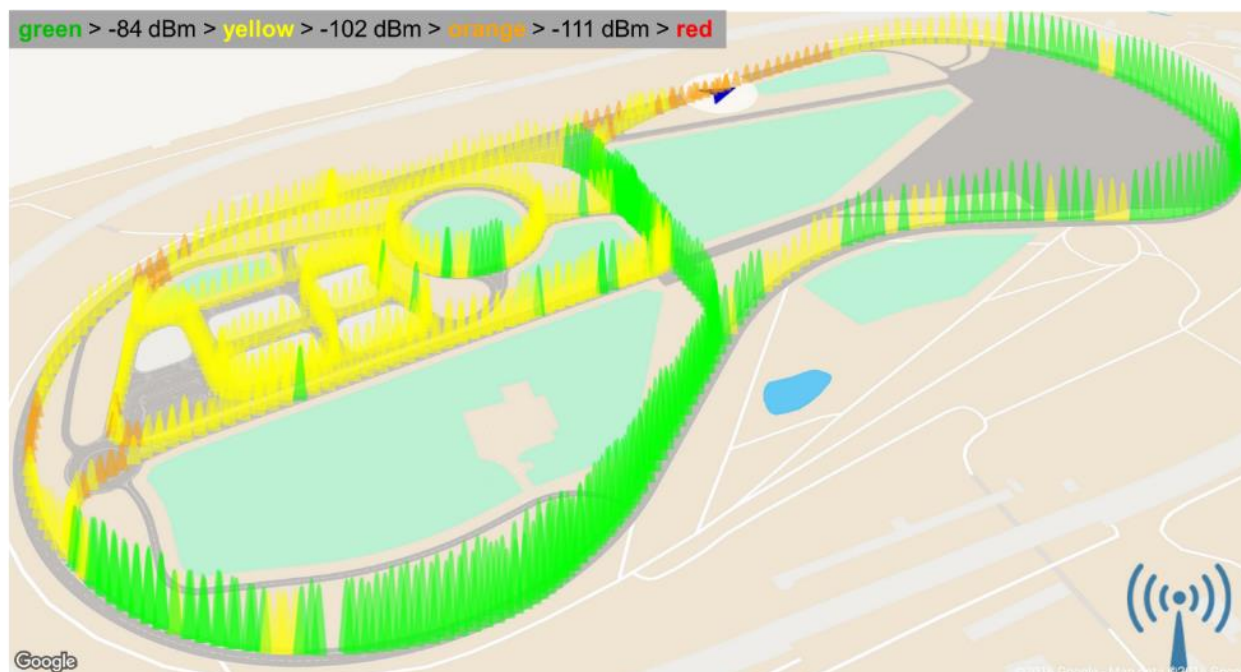


Figure 4.1: RSRP measurements across the test track, track show the circuit is for the most part covered with satisfactory received power, only showing few localized spots characterized by poor coverage.

The antennas are connected to a baseband unit and packet core user plane right at the bottom of the mast, where the components of the 5GCAR messaging system are hosted as well. More information on this setup is provided in Section 3.1.4.

4.1 Lane Merge Coordination

In the following, several aspects of the lane merge coordination demonstration are evaluated, and learnings are formulated as recommendations to other projects. The evaluation starts with the localization accuracy of the roadside camera system, then investigates automotive KPIs (inter-vehicle distance, acceleration, and maneuver length), before discussing communication KPIs, with a focus on latency.

4.1.1 Camera System Localization Accuracy

The goal of this section is to analyze the accuracy of the camera-based vehicle localization as described in Section 3.1.1. For this, the localization results of the camera system are compared



to the localization of the GNSS RTK measurements (cf. Section 3.4.2) of the vehicles. This is only possible for the connected cars. For the demonstration of the evaluation, the following evaluation diagrams are shown:

- The comparison of resulting coordinates (camera, GNSS RTK) with respect to the time index (Figure 4.3, Figure 4.6).
- The **difference** between the camera-based localization and the GNSS RTK (Figure 4.4, Figure 4.7) with respect to:
 - The time index (Figure 4.2, Figure 4.7).
 - The distance d_{VC} between vehicle and camera (since the camera system accuracy is expected to be dependent on d_{VC}).

Since the for localization (camera-based and from connected vehicle) sources are not synchronized, the data from the one with the lower update rate (vehicle GNSS RTK) is linearly interpolated.

Since the vehicle GNSS RTK does not provide ground truth, but is erroneous as well, we first decompose the observed error in an experiment using the nearly exact image positions of the GNSS antenna. Secondly, the camera system is evaluated using the GNSS RTK as if it was ground truth. To subsume, we evaluate:

- **GNSS RTK and Camera Calibration Accuracy:** Comparison between ground truth position in the images and GNSS RTK for one selected vehicle (offline experiment).
- **Camera System Accuracy:** Comparison between the localization of the camera system and GNSS RTK for the final demonstration (online processing).

GNSS RTK and Camera Calibration Accuracy

The GNSS RTK sensors, measurement units as used for the self-localization of the connected vehicles, provide accuracy in “the range of centimeters”. For evaluation, their localization is compared with the localization accuracy of the camera system for “nearly” ground truth image position. This is done by the manual selection and automatic tracking of the GNSS RTK antenna in two camera views (cf. Figure 4.2). These 2D positions are projected into the plane of known height (height of the antenna for the Volvo V-60: 1.43 m) to determine the 3D position of the antenna which is then compared to the GNSS RTK measurements [ICCVE19].



Figure 4.2: Visual tracking of the GNSS RTK antenna for evaluating the GNSS RTK accuracy (input data from May 10th, 2019).

The remaining error sources are:

- Localization error of the GNSS RTK units (independent of distance to the camera).
- Calibration error of the camera system, including the visual feature tracking (increases with distance to the camera).

The resulting positions in the global coordinate system are shown in Figure 4.3 for the 8 s where the antenna is sufficiently visible for 2D tracking. The diagrams show a good match between the 3D coordinates originating from the calibrated cameras and the GNSS RTK measurements.

The difference between GNSS RTK and the visual antenna tracking is shown in detail in Figure 4.4. Here, the difference is displayed with respect to the distance between the vehicle and the camera system d_{VC} (reference position between Camera 1 and Camera 0). Negative values show underestimated distances while positive values show that the distance is too large.

The localization error e is smaller than 2.5 m for a camera vehicle distance d_{VC} smaller than 140 m and increases with d_{VC} . The large variation in the error, especially for large distances, is due to mast shaking in the wind (wind speed: approximately 20 km/h). Thus, for the development of the vehicle localization, a compensation algorithm for the mast movement is applied to the system.

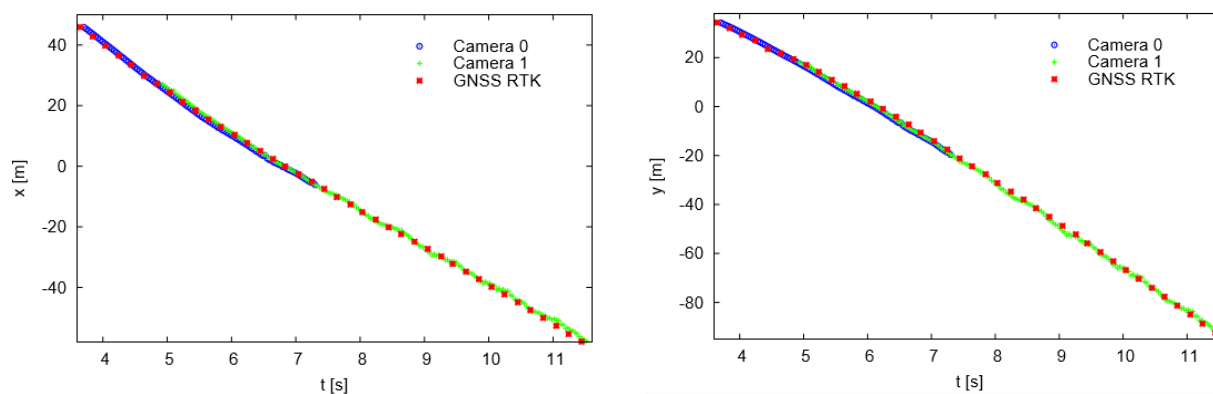


Figure 4.3: Antenna localization of GNSS RTK compared to the 2D antenna tracking for two cameras separately.

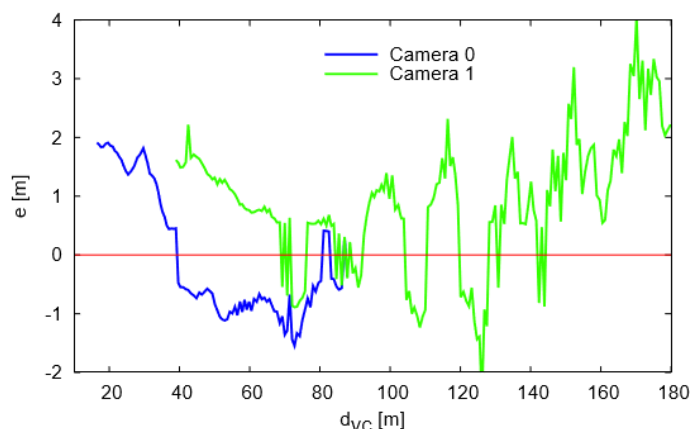


Figure 4.4: Distance between the visual localization of the antenna (cf. Figure 4.2) and the GNSS RTK localization.

Camera System Accuracy

The camera system-based online localization as demonstrated on the final event (June 27, 2019) consists of the following main features:

- Highly accurate multi camera calibration [VISAPP19].
- State of the art vehicle detector based on an extensively trained convolutional neural network (CNN) [YOLOv3].
- Compensation of mast motion based on tracked feature points [FEAT94].
- Intra-camera tracking and localization for each video stream.
- Inter-camera merge of objects using 3D positions in the global coordinate system.



Figure 4.5: Visualization of the three camera views for the same time index (input data from June 27, 2019).

The system processes three video streams in real-time (each with 30 frames per second) and transmitted vehicle status information with a latency of approximately 70 ms (i.e. one frame for capturing, one frame for processing) to the Data Fusion via the V2X Gateway.

A synchronized multi-image example is shown in Figure 4.5. The camera-based localization is compared to the GNSS RTK localization in Figure 4.6 and Figure 4.7. In Figure 4.6, the resulting positions of the camera system (five vehicles) and the connected vehicles (three vehicles) are displayed. The task of the data fusion is matching the vehicles Veh_i to the connected vehicles

(Peugeot 5008 (5008), Citroen DS7 (DS7), and Volvo V-60 (V-60)) based on their positions. Here, the correct mapping is: 5008 \leftrightarrow Veh₁ (merging vehicle, starting on lane 0), DS7 \leftrightarrow Veh₃ (lane 1), and V-60 \leftrightarrow Veh₄ (lane 1). Veh₂ is the first unconnected vehicle appearing in the camera view, Veh₅ is the last, and both appear on lane 1. Based on this mapping, the localization difference for the connected cars is evaluated in Figure 4.7. The difference is shown with respect to the time index (left) and with respect to the vehicle camera distance d_{VC} (right).

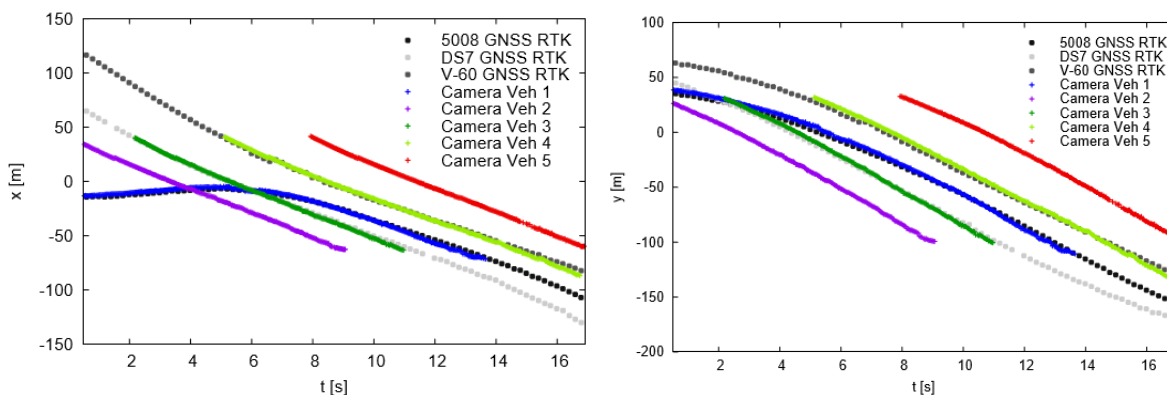


Figure 4.6: Camera-based localization of the five cars (3 connected, 2 unconnected), compared to the GNSS RTK vehicle localization for a lane merge maneuver with respect to the time index t . Here, the coordinates (left: x -coordinate, right: y -coordinate) for each position estimate on the output are shown.

Due to the perspective of the camera of relatively low height compared to the observed distance of the vehicle, the computation of the 3D location of vehicles tends to overestimate the object distance for large distances. By adjusting the height $h_r=1.2$ m of the reference projection plane, the estimation leads to the smallest errors in the main region of interest for the lane merge application (distances between 80 m and 180 m). Following from the perspective of the camera views, the errors are mainly in the longitudinal direction of the vehicles, which eases the tracking and the assignment for the Data Fusion.

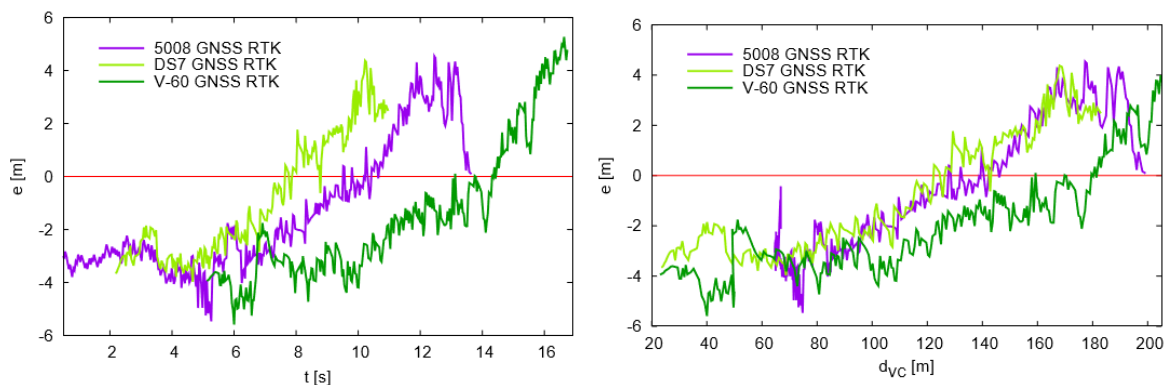


Figure 4.7: Difference between the camera-based localization of the three connected cars and the self-localization of the vehicle with respect to the time index (left) and the camera vehicle distance d_{VC} (right).



Summary

The implemented system for camera-based localization provides reliable and accurate results. The localization error is below 5 m up to a vehicle camera distance d_{VC} of 200 m. Due to perspective effects, the localization tends to underestimate the distance for smaller distance d_{VC} while overestimating it for larger distance d_{VC} . The height of the projection plane h_r provides a parameter for adjustment. Thus, it is adjusted such that the smallest errors are obtained in the target region of the lane merge maneuver preparation.

Future works shall exchange the 2D detection model with a 3D model for each vehicle, estimated from the image data. This approach solves the problem of overestimating the distance for large camera vehicle distances d_{VC} while underestimating it for small distance d_{VC} and, thus, improves the localization accuracy.

4.1.2 Automotive KPIs

In [5GC18-D51], the following automotive KPIs have been selected to evaluate the demonstration:

- Minimum inter vehicle distance shall be 2 s, with a possibility to be reduced to 0.9 s for a maximum duration of 3 s, as we are using human drivers in the demonstration, and this is a more challenging requirement compared to autonomously driving vehicles.
- Maximum acceleration or deceleration shall be 2 m/s² in longitudinal direction and maximum lateral speed shall be 1 m/s (maximal lateral acceleration: 1.5 m/s²) to ensure smooth maneuvers.
- The relevance area shall be 400 m to ensure a complete coverage of the lane merge area and the planning of a smooth maneuver, for all covered cases.

As the demonstration was only executed in a single scenario, these results only evaluate the basic feasibility of a lane merge coordination, but do not give indications on the universal applicability of the developed system. It should be noted that simulation-based studies and large-scale trials would be needed for such an assessment, covering a diverse set of scenarios and traffic regulations.

Inter-Vehicle Distance

In the field tests, the vehicles on the highway were driving with 70 km/h, which corresponds to approximately 19.5 m/s. Consequently, the mandated 2 s distance lead to a distance of approximately 39 m between the vehicles, and the possibility to reduce the distance to 17.55 m for a limited duration of 3 s. In the demonstrations, the entering vehicle accelerated to the target speed of 70 km/h before merging onto the highway, which is why these are the distances considered in the evaluation.

To measure the inter-vehicle distances in the field tests, the positions of vehicles on the same lane were interpolated to the same times, and then the distance between the back of a leading car and the front of a following car was calculated.

Figure 4.8 shows that for most cases, the inter-vehicle distance was above 45 m, with a small share of distances between 22 m and 29 m being present, corresponding to the point in time where the entering vehicle merged onto the highway into the still growing gap.

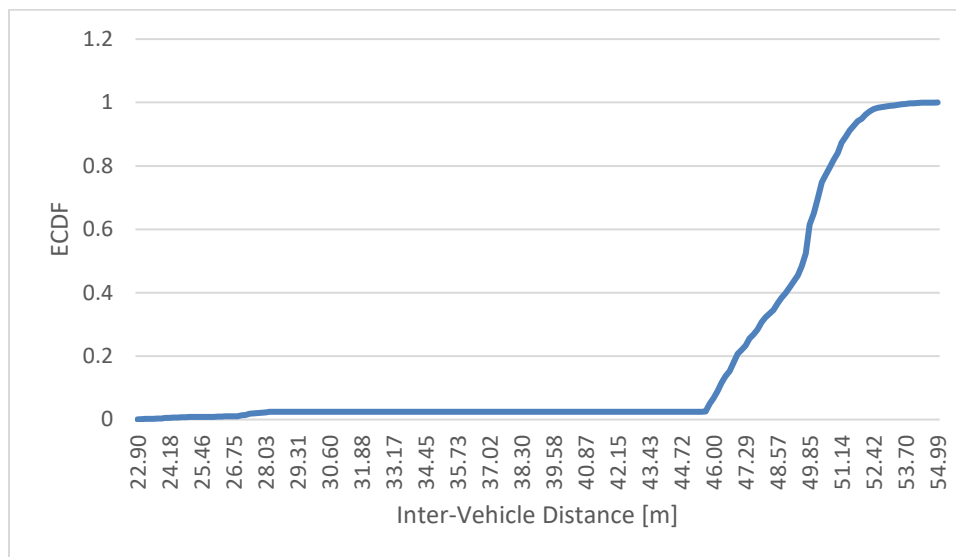


Figure 4.8: The distance between vehicles involved in the maneuver was kept above 2 s (approximately 39 m at 70 km/h) in most cases, and approximately halved for a limited time when the entering car merged.

Maximum Acceleration and Deceleration

To evaluate this KPI, the acceleration (deceleration is considered negative acceleration) of all connected vehicles was logged and analyzed in post-processing. Only the acceleration occurring in the lane merge area was considered, i.e. only the acceleration within the area that the traffic orchestrator is controlling. Figure 4.9 shows histograms of the leading, entering, and following vehicle in the scenario area (note the logarithmic scale). Only during 0.2% of the maneuver time was the leading vehicle outside the defined ± 2 m/s² constraint, during 0.3% for the entering vehicle, and 0.2% for the following vehicle.

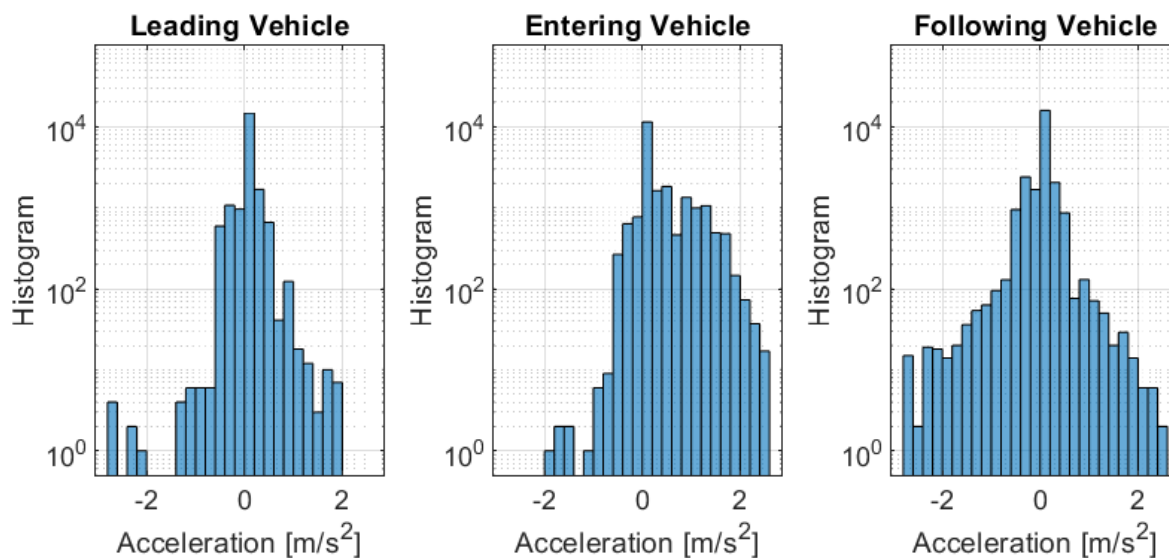


Figure 4.9: Histograms of acceleration values for leading, entering, and following vehicle, in logarithmic scale.

Relevance Area

The relevance area is defined as the stretch of road where the orchestrated maneuver is executed, i.e. the distance from the point of first instruction, to the point of maneuver completion.

This KPI is important because motorway ramps are typically (depending on the country/legislation) designed with a certain length corresponding to allowed and expected vehicle speeds in the respective scenario. Thus, this is a hard requirement to fulfill, but since the given thresholds are upper limits used for road construction, actual maneuver lengths usually stay far below this threshold by design. As can be seen in Figure 4.10, the maneuver length always was between 80 and 92 meters.

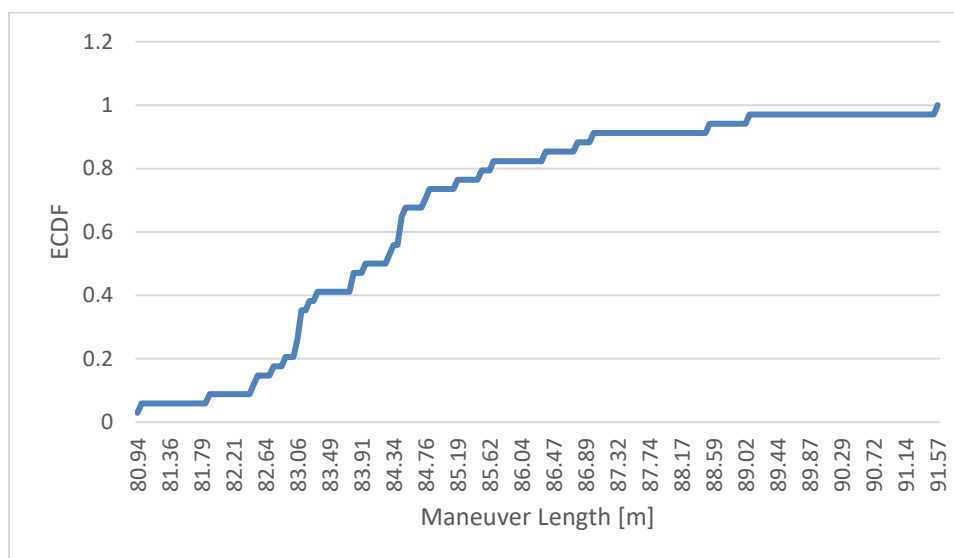


Figure 4.10: ECDF of the maneuver length.

4.1.3 Communication KPIs

In [5GC18-D51], the communication KPI requirements have been defined. A summary is given in Table 4.1, including an evaluation summary.

Table 4.1: Overview of KPIs and corresponding evaluation summary.

KPI Name	KPI Requirement	KPI Evaluation
Communication Range	Communication range around lane merge area shall exceed 400 m, in order to cover the whole relevance area.	As can be seen in Figure 4.1, the coverage, and thus communication range, was good in the target area, and in general is only a deployment question in cellular networks.
Availability	Availability shall be high, i.e. at least 99%.	This was not assessed in detail, as it requires long-term trials over several weeks or months and is not representative in a prototype-based implementation.
Reliability	Reliability shall be very high, i.e. at least 99.9%.	This is was achieved and is elaborated on below.
Latency	Latency shall be less than 30 ms per message. It corresponds to the elapsed time from devising a maneuver in the lane merge coordination system until all concerned vehicles receive the	We were not able to assess this precise one-way delay, as the time synchronization was not good enough. Instead, we did a more detailed latency study, presented below.



	corresponding instructions.	
SDU size	The SDU size shall be less than 1,200 bytes for safety messages and less than 16,000 bytes for maneuver messages, at application layer.	Although JSON messages were used in the demonstration, which are not very efficient with respect to SDU size, all messages were below 800 bytes.
Average data rate	Average data rate shall be lower than 96 kbps per vehicle in uplink, and lower than 128 kbps per vehicle in downlink.	In the demonstrations, vehicle status information was sent by each vehicle at a fixed rate of 5 Hz, each message having less than 800 bytes. Consequently, the average uplink data rate was 32 kbps, plus a negligible number of maneuver feedback messages. In the downlink, descriptions of all vehicles were sent to all vehicles with 10 Hz, resulting in 320 kbps. However, this information was sent for convenience and showcasing (to see which vehicles are where inside all vehicles) and is not needed for the use case itself. For the use case, only maneuver messages were sent, which are negligible in size.
Peak data rate	Peak data rate shall be at least 320 kbps per vehicle in uplink, and at least 4.27 Mbps per vehicle in downlink, for a latency requirement of 30 ms and Road User description messages being sent in the uplink, while maneuver messages are sent in the downlink.	As time synchronization was not working well enough to enable reliable one-way delay measurements, the one-way peak data rate could not be assessed based on the use case messages. However, in separate experiments, data rates of up to 10 Mbps in uplink and downlink for all vehicles in parallel were consistently achieved. Higher bitrates were not tested, but the requirement is fulfilled.

Latency

The latency was measured using logging by every component, including an ingress and egress timestamp for each message in every component it passes through. This way, all latency contributions can be evaluated in detail, however, time synchronization in the vehicles was not working well enough to do reliable one-way latency measurements to and from the vehicles, which is why the focus in the evaluation is on Round Trip Times (RTTs). Also, as a tool for analyzing and characterizing different latency contributions, some latency distributions were combined by convolving the probability density functions of the respective latency contributions, to get a combined latency contribution. It should be noted that a percentile of such a combined latency

contribution is not equal to the sum of the respective percentiles of the individual latency contributions.

All results presented in the following are based on more than 380,000 measurements, unless stated otherwise.

The E2E RTT of a standard measurement setup during the demonstration is illustrated in Figure 4.11, in the form of percentiles, meaning that e.g. 99.9% of all messages had an E2E RTT of less than 288 ms.

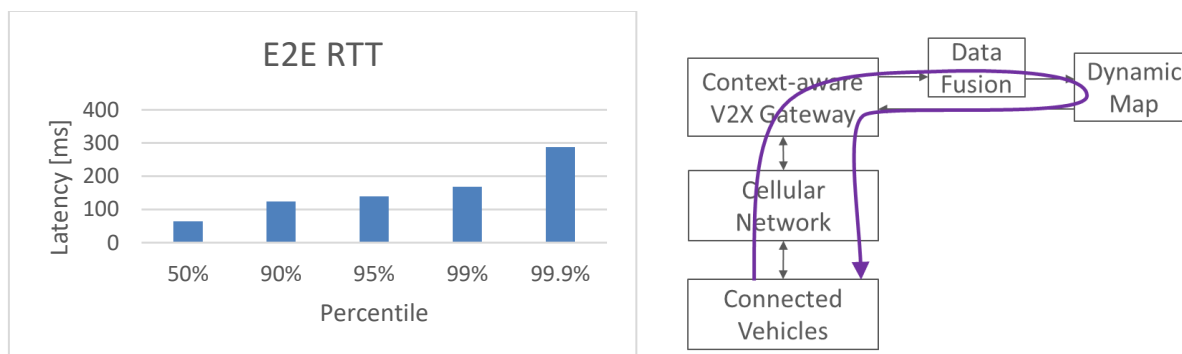


Figure 4.11: E2E RTT of vehicle status information as delivered in the demonstrations.

There are several significant contributors to that latency:

- Data Fusion.
- TCP buffering.
- Radio network latency measured as RTT between Connected Vehicles and V2X Gateway.

Processing delays of other components, as well as transport delays between the components were negligible in this edge cloud deployment.

Data Fusion

As described in Section 3.1.2, the Data Fusion operates at periodic intervals. In the evaluation shown in Figure 4.11, the Data Fusion was active every 100 ms, leading to a delay with uniform distribution between 0 and 100 ms. Figure 4.12 shows the corresponding measured processing time, resembling a uniform distribution with a bias to slightly higher latencies, while the 99.9-percentile is still at 100 ms. A few significantly larger values are also present, which is a normal artifact coming from multi-threaded operating systems.

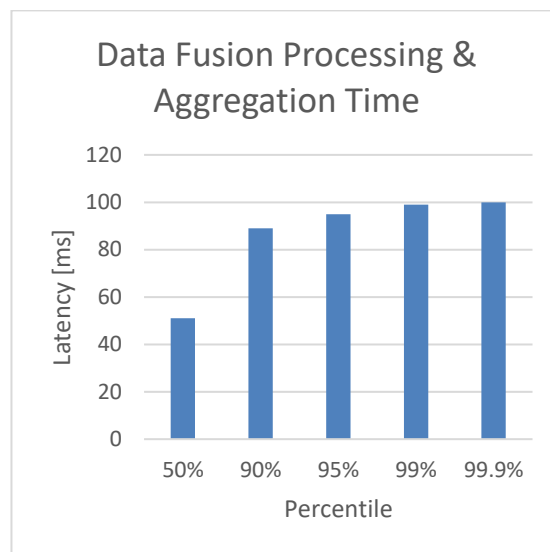
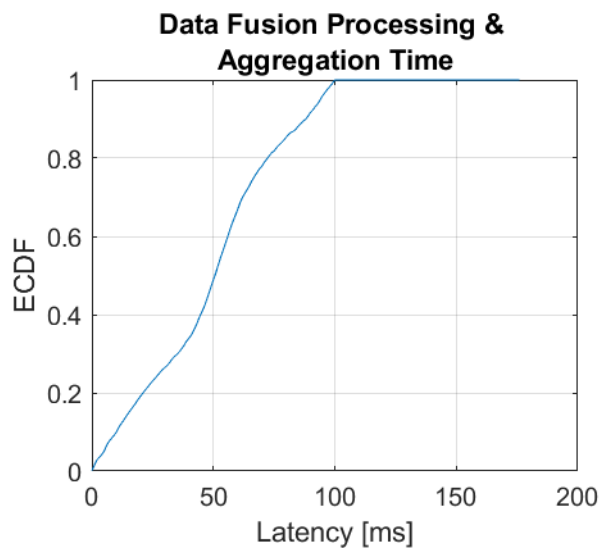


Figure 4.12: ECDF and percentiles of processing and input data aggregation delay of the Data Fusion component.

RTT between Connected Vehicles and V2X Gateway

The smallest latency contribution measured over the radio interface was the latency between OBU and V2X Gateway, in both directions. This was calculated from ingress and egress timestamps of a message in the V2X Gateway and a Connected Vehicle, as illustrated on the right side of Figure 4.13. Furthermore, the RTT is significantly higher than in raw LTE RTT measurements. The reason for this comes from missing latency optimization of the MQTT Broker.

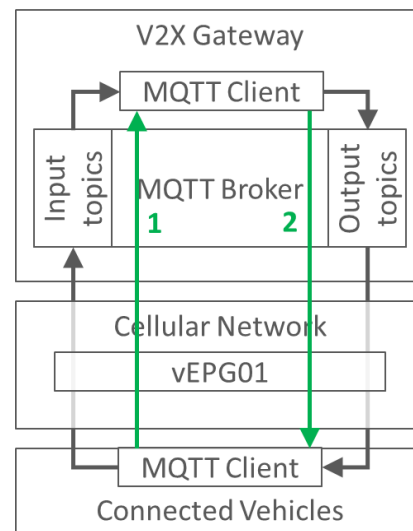
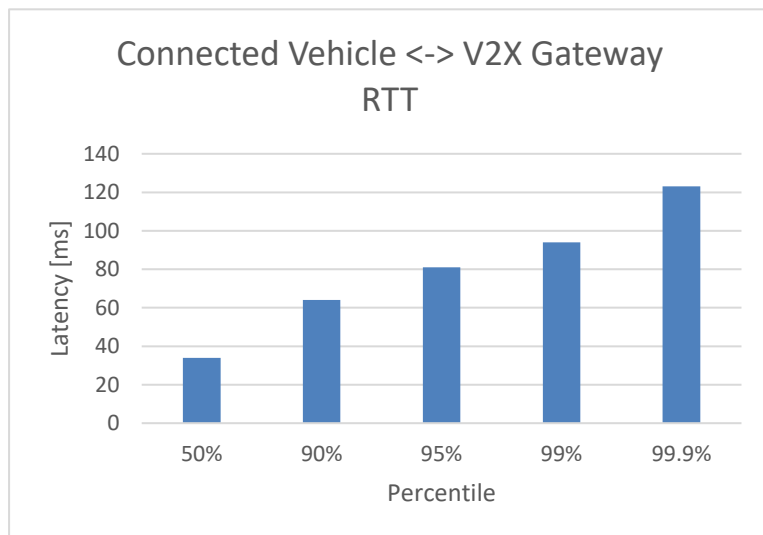


Figure 4.13: Left: latency percentiles of the RTT between Connected Vehicles and the V2X Gateway; Right: illustration of points of measurement.

TCP Buffering Latency

In a separate experiment, the latency generated by the MQTT Broker was measured, i.e. the latency of a single hop highlighted in the right part of Figure 4.13. While the latency is around a few ms in the measurement shown in the right figure, the latency is significantly higher in the experiment shown in the left figure. This is caused by Nagle's algorithm, which causes a TCP sender to buffer data for tens of milliseconds before sending it, in case more data arrives to fill an IP packet. While this optimizes the payload to header ratio, it has a significant drawback on latency, which is why it is best practice to turn this off for low latency applications such as gaming [SKS+10]. Furthermore, in the demonstrations, each message passed the MQTT Broker twice, causing the latency to accumulate twice. Figure 4.14 shows the latency percentiles of messages passing the MQTT Broker one and two times, with and without Nagle's algorithm.

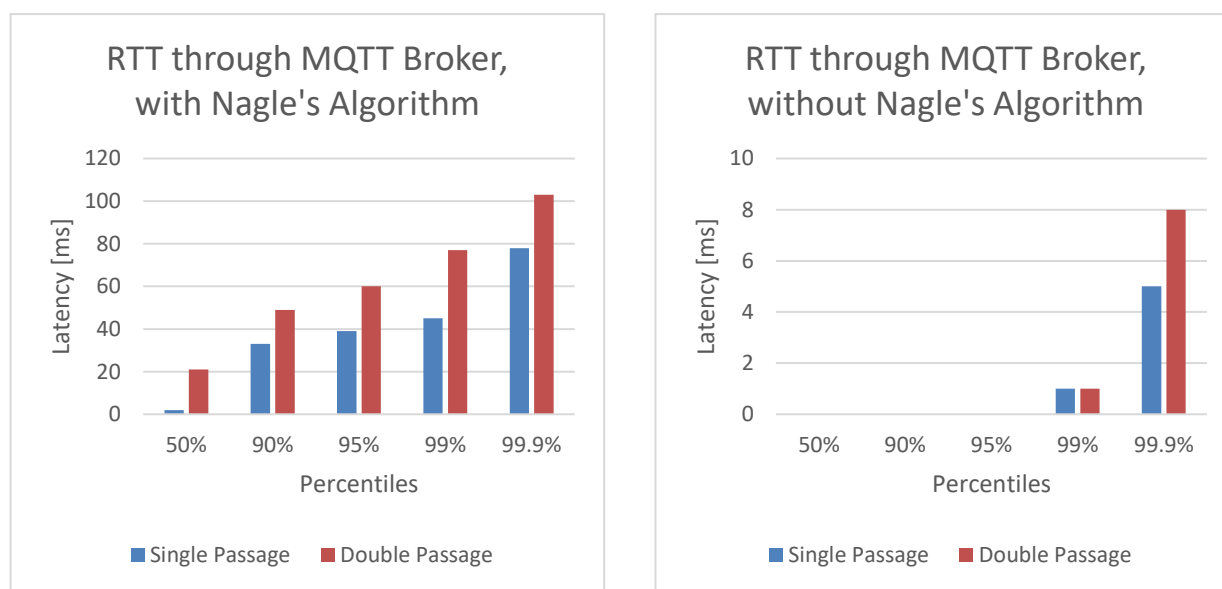


Figure 4.14: RTT through MQTT Broker (more than 100,000 measurements).

Edge Cloud versus Central Cloud

An experiment was done to highlight the latency impact of deploying and co-locating all the messaging components in the edge cloud, versus having a deployment on public internet. Different hardware was used in both cases, and more importantly different computing power was available to the instantiated applications (more compute power in the edge cloud), and finally the edge cloud was used exclusively, while the central cloud was shared with other content. Figure 4.15 shows the distribution of the components in the central cloud setup, where the V2X Gateway was instated in Lund (Sweden), while the Data Fusion and Dynamic Map were instantiated in Paris (France), where also the Connected Vehicles were located. Furthermore, the travel path of vehicle status information is sketched in the figure.

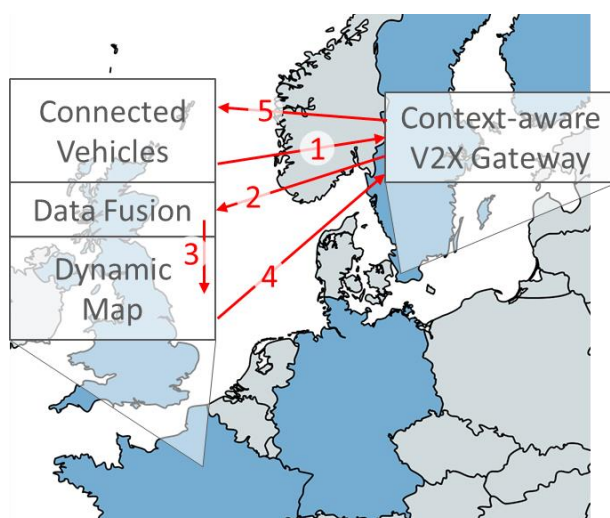


Figure 4.15: Deployment sketch of the central cloud deployment, where components were instantiated in Lund (Sweden) and Paris (France), with the steps how each messages is delivered.

As expected, the E2E RTT is significantly lower in the edge cloud deployment, where most of the delay reduction comes from reduced transport delays, as shown in Figure 4.16. Also, the case of having all components deployed in Lund is evaluated, by assuming the latency distribution of step 3 also for steps 2 and 4 (cf. Figure 4.15). As can be seen, this brings a large gain for the lower percentiles (traveling distance is lower), while the advantage is weaker for the higher percentiles (caused by latency jitter over shared links).

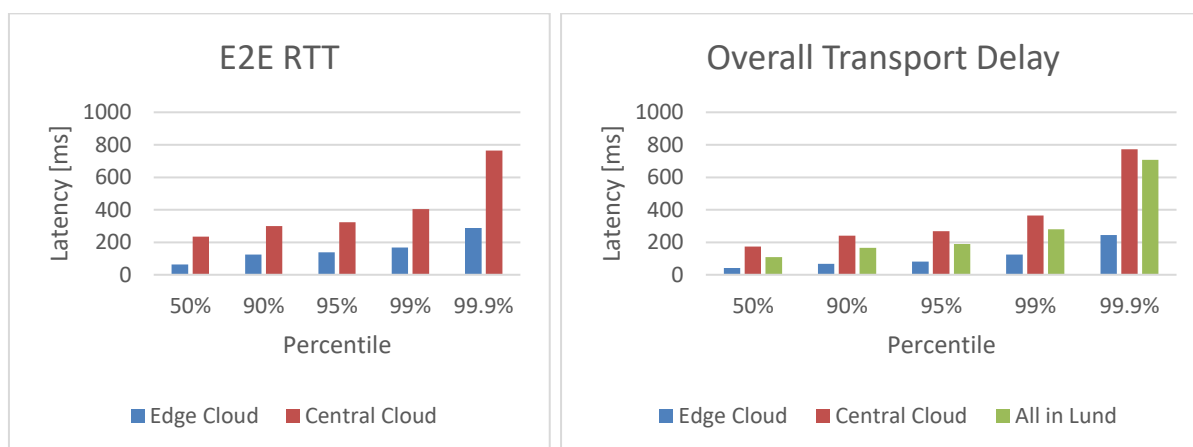


Figure 4.16: E2E RTT and the sum of all transport delays (i.e. steps 1-5 from Figure 4.15), for the edge deployment, the central deployment case, and the case of having all network-side components instantiated in Lund.

Figure 4.17 shows that the overall processing delay (combined processing delay of V2X Gateway, Data Fusion, and Dynamic Map) is similar for both cases. The processing delay is slightly better in the edge cloud deployment because more compute capacity was available.

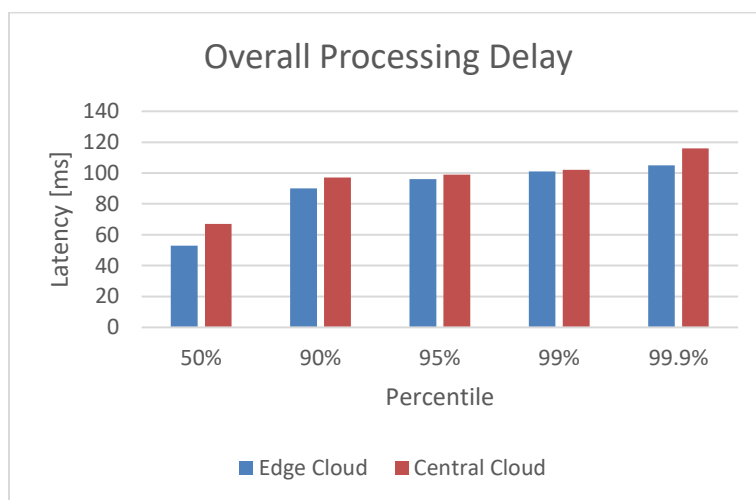


Figure 4.17: Overall processing delay of the edge and central cloud scenarios.

This comparison only considers the two extreme cases, of having the applications deployed close to and with a dedicated link to the radio access network, versus nowadays typical cloud setup of having a few data centers distributed over Europe. More detailed follow-up studies should consider different deployment densities of the edge cloud (e.g. continent-level, country-level, city-level, multi-base-station edge, and single-base-station edge), as well as different backhaul types (exclusive fiber, shared fiber, microwave, etc.).

Latency Conclusion

Figure 4.18 illustrates the higher percentiles of the various latency contributions in the system. While the latency contributions discussed previously are most significant in general, it was observed that even latency contributions that are mostly small have peaks that become significant when approaching high reliability. While no detailed study was performed to characterize these latencies, such occasional latency peaks on the transmission links usually originate from TCP layer effects such as retransmissions (order of several 100 ms), TCP slow starts (not significant in the implemented setup, due to hanging TCP connections) and again Nagle's algorithm (order of several 10 ms). In the processing delays, such delay peaks – even though not as significant – are naturally come with multi-threaded operating systems.

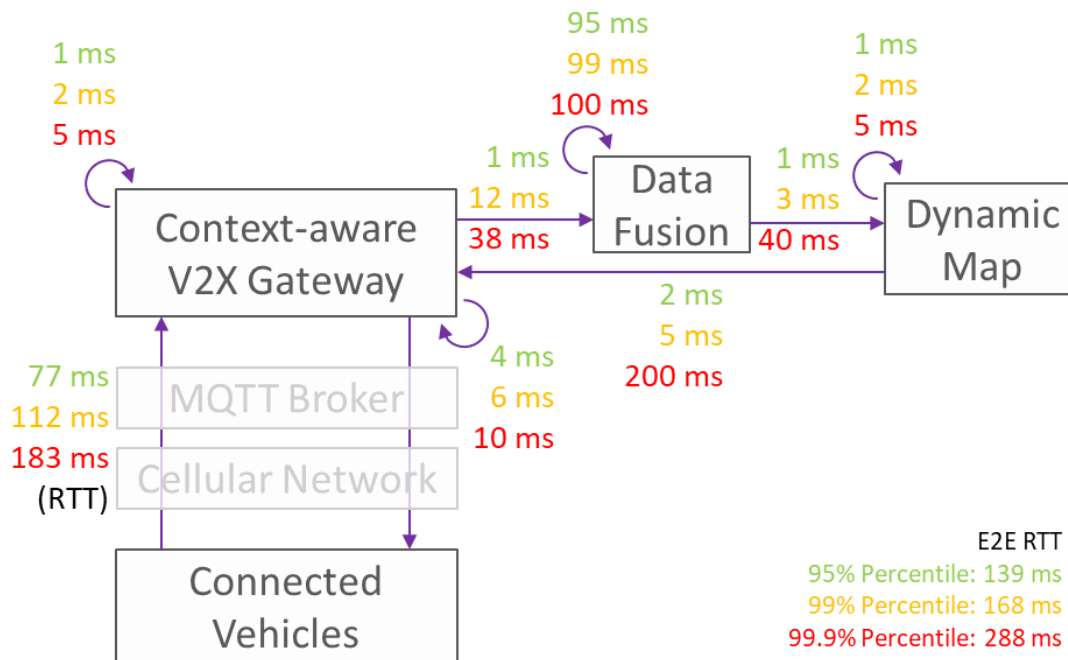


Figure 4.18: Latency percentiles of all relevant transport and processing steps in the messaging framework as used in the lane merge coordination demonstration.

The Data Fusion aggregation delay can be greatly improved by reducing the sampling interval from 10 Hz to e.g. 30 Hz or lower, or even implementing an event-based mechanism, potentially at the cost of efficiency and complexity.

It was shown that the latency impact of Nagle’s algorithm is very significant, which is why the worse packet header to payload ratio should be accepted for such low latency use cases. The RTT between Connected Vehicle and V2X Gateway was not measured without Nagle’s algorithm for this use case, but a separate experiment was done to measure the ping RTT between these two components. LTE and 5G-NR both do not have synchronous latency characteristics, the downlink has latency benefit compared to the uplink (i.e. downlink latency is lower, unless more loaded). As an upper approximation, we can assume half of the ping RTT as the downlink one-way delay from V2X Gateway to Connected Vehicles. To get an idea about the one way latency from an application such as the Traffic Orchestrator to Connected Vehicles, we add the single pass MQTT Broker delay without Nagle’s algorithm shown in Figure 4.14, the V2X Gateway processing delay and a typical TCP one way delay from Figure 4.18, by convolving the corresponding distribution functions. As a result, we get the one-way delay as illustrated in Figure 4.19. Since LTE radio access was used in this case, the 99.9-percentile is significantly larger than the KPI requirement of 30 ms. Detailed 5G-NR latency measurements are not yet publicly available, so the expected performance cannot be assessed here. However, as the remaining latency in this case is dominated by the delay in the radio access network, it is plausible that using 5G-NR would enable delays below 30 ms with 99.9% reliability, as long as a low latency jitter can be maintained. Further field tests would be needed to confirm this.

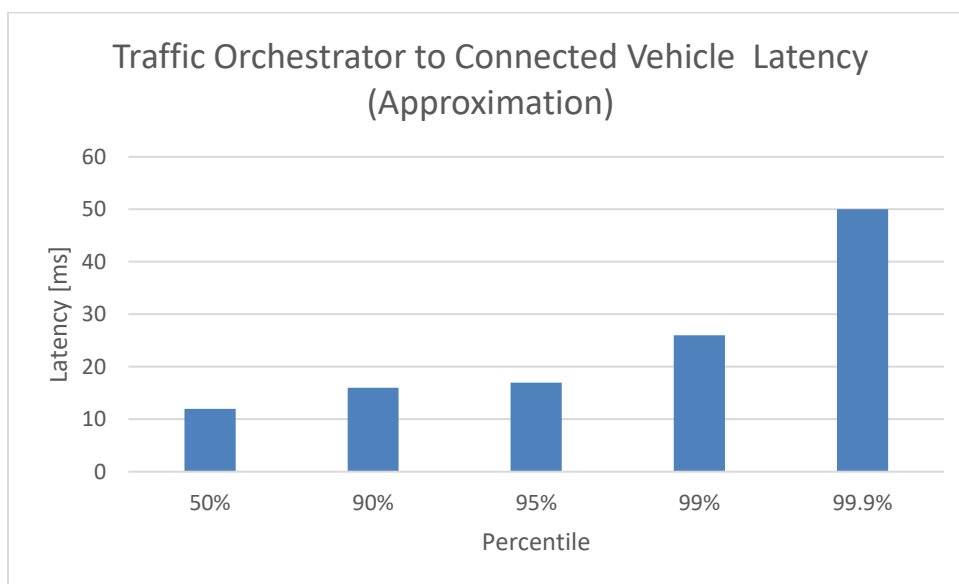


Figure 4.19: Approximation of one-way latency from Traffic Orchestrator to Connected Vehicle, when using half the radio access network RTT as upper approximation for the downlink one-way latency.

Reliability

In general, the communication reliability has been evaluated by comparing the number of sent vs. received messages. As the Data Fusion function may combine input messages, thus changing the number of output vs input messages, this evaluation was done in two steps: from Connected Vehicles to the Data Fusion, and from the Data Fusion to Connected Vehicles. All in all, more than 147,000 messages were considered for the evaluation, where 100% reliability was achieved, i.e. no messages were lost.

The reliability requirement of 99.9% corresponds to maximum 1 out of 1,000 messages lost, so for the number of messages considered in the experiment, this corresponds to a maximum loss of 147 messages, which is why the reliability is assessed with sufficient confidence. However, a reliability of e.g. 99.999% cannot be derived with confidence with this number of messages sent (would correspond to approximately 1 message lost).

There are two reasons for not losing any packet:

1. During this experiment, it was taken care that no component was restarted, i.e. 100% availability of the system, and messages received right at the start or sent right at the end of the experiment were not included in the evaluation.
2. Aside from the communication between Connected Vehicles and V2X Gateway, all communication was done via exclusive links, or even on the same machine. For the communication between Connected Vehicles and V2X Gateway via the cellular network, MQTT based on TCP was used, on top of LTE radio access with 4 HARQ retransmissions and RLC acknowledged mode. Consequently, there are multiple layered retransmission



mechanisms in place for increasing reliability, which is why a reliability of more than 99.9% is expected.

4.1.4 Learnings

Some general learnings from the demonstration are summarized in the following.

In general, managing several prototypes, and considering all the dependencies among such high number of prototypes made the integration and testing of the complete setup really challenging, because the performance of one component depends on the quality of the input information coming from other components, which again depends on the performance of said components. It is recommended to keep dependencies on prototypes to minimum, when planning such a demonstration, to avoid bottle necks and increased time pressure on other components. Furthermore, it is recommended to keep the technical interfaces between components of different partners as simple as possible, and e.g. avoid integration of closed code between partners, as this greatly complicates troubleshooting and bug fixing.

On the positive side, the choice of using MQTT greatly facilitated the integration of components, as it is easy to structure content, have a subscription mechanism, and comes with minimal client complexity. For low latency performance, it is really important to use a QoS of 0, which deactivates any MQTT acknowledgements and retransmissions, for both subscriptions and publishing and disable Nagle's algorithm. For some interfaces, TCP sockets were used, which is definitely not recommended due to huge complexity on both ends (message segmentation, manual detection of broker sockets and reconnection, etc.). Furthermore, using docker and docker-compose to easily and flexibly host all applications on VMs on the hardware next to the test track was very efficient.

Finally, even though RTK was used, lane-accurate positioning was difficult to achieve, especially with the low latency requirements. This was done in the project by mapping the GNSS position to lanes, based on a static map, and the lateral jitter was too large to avoid lane position estimation errors. Due to the low latency requirements, filtering options (e.g. removing outliers by comparing succeeding and preceding samples) were limited. It is recommended to evaluate the positioning requirements in detail early in the project and consider using advanced in-vehicle positioning systems (including dead reckoning and camera-based lane detection) at the cost of flexibility if this is feasible.

For the Intelligent Camera System, it became clear that the accuracy of the camera calibrations is of key importance since large distances (up to 200 m) are to be covered. The calibration process took more time than expected. Furthermore, simple models for localization lead to large localization errors at large vehicle-camera distances. The reason is the relatively low camera height compared to the large distances. The reliable estimation of the 3D shape of the vehicle is expected to greatly improve the localization accuracy. Thus, a smaller localization error can be achieved.

For the Data Fusion, it proved useful to have a weighting function for considering different parameters in the association step, which can be influenced by accuracy parameters for the



different parameters. Furthermore, using the vehicles' size and color proved to be less helpful than expected, due to difficulties in precise estimation in the camera system, coupled with similar values among the used cars.

The Traffic Orchestrator faced a few hardships during testing phase, from its strong dependency on other components to shortage of lane merge data for different scenarios. Since the stack is a closed loop application, testing how the Traffic Orchestrator adapts its maneuver according to the driving behavior happened very late in the project, however, the Traffic Orchestrator would have benefited from a simulator such as SUMO [LBB+18]. This would enable new imagined scenarios to be created, as well as a more thorough testing phase for a well-rounded, complete solution concerning the neural network aspect. Although reinforcement learning is somewhat adaptable to other scenarios, the Traffic Orchestrator should have been aided with previous footage of lane merge on the same test track, to ensure maximum human-like trajectories, reducing the need for scenario-specific pre- and post-processing of input and output data of the Neural Network, and finally provide a source of comparison for calculated and human trajectories.

4.2 See-Through Sensor Sharing

For the see-through sensor sharing the evaluation is separated into an application and a communication part, which are detailed in the following Sections.

4.2.1 Application KPIs

In [5GC18-D51], the following KPIs have been selected to evaluate the demonstration of the see-through application:

- Good image quality of the see-through video overlay in the rear car, which can be influenced by the image resolution and compression. These two values can be adopted to the link quality and the available bandwidth via the direct V2V sidelink.
- Good fit of the position of the see-through video overlay over the front car which should be hidden. This aspect is depending on a good relative positioning and a low packet error rate during the transmission of the car poses.
- Low latency of the see-through video overlay. This latency can be recognized when the video content in the video overlay coming from the front car is not perfectly aligned with the rest of the video. This is especially visible in corners where the position of the front car is moving fast relatively to the rear car. This latency can result from transmission delays and processing times in the algorithm.

To evaluate the first KPI the maximum bandwidth/throughput for the see-through application was measured with a good quality of pixel-by-pixel transmission and a very low compression. The throughput of the application is depending on the distance between the two cars because the see-through video overlay will be smaller if the front vehicle is more far away and the front car will only transmit the necessary part of the video stream.

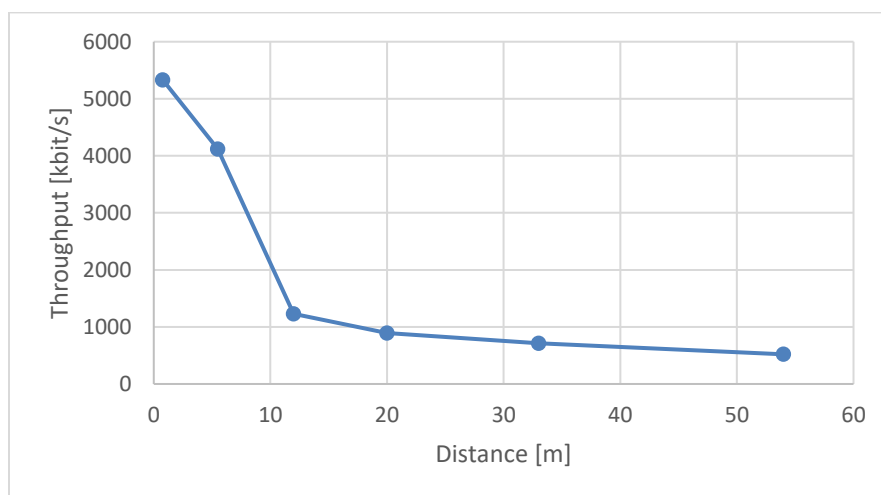


Figure 4.20: Throughput of the see-through application over distance of the cars.

The throughput required by the see-through application is shown in Figure 4.20. Due to the pre-processing performed before sending the stream, the maximum throughput of the application reaches 6 Mbit/s and can be handled by the 5G direct communication link. Therefore, the image quality is not limited by the performances of communication module.

The second KPI is evaluated in the next section for the communication KPIs, as the low packet error rate is depending on the modem and the software will not induce additional packet drop.

For the third KPI, the latency of the end-to-end system as depicted in Figure 3.15 is measured. This latency from one see-through application computer to the other and back to the first is shown in Figure 4.21. We used RTT because the cars do not have a GNSS system attached and therefore the clocks are not synchronized. Video stream latencies are not included as the delay of the camera itself can't be measured without special tools. The latency is low and does not decrease the latency of the see-through video overlay noticeably.

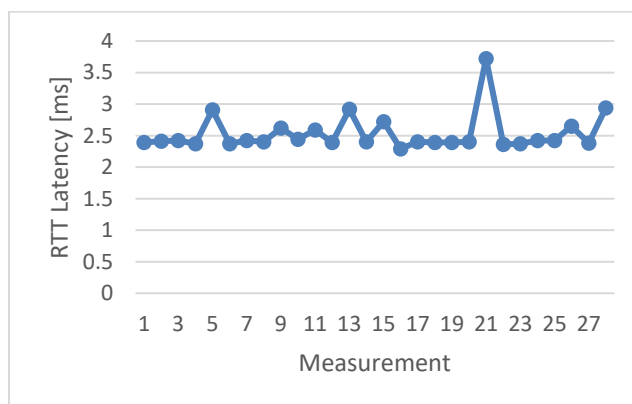


Figure 4.21: E2E latency between the application endpoints.

4.2.2 Communication KPIs

Before the field tests and final demonstration, the reliability enhancement, especially the enhancement with multi-antenna Tx and Rx diversities has been evaluated by simulation studies. Figure 4.22 presents the simulation results of diversity gains which seem to be significant comparing with the single antenna case, even when the channel correlation is still high. As can be seen, the raw BLER performance when the channel correlation factor is 0.8 or 0.6 is quite close to the fully uncorrelated channel, which is over 5 dB better than the single antenna case.

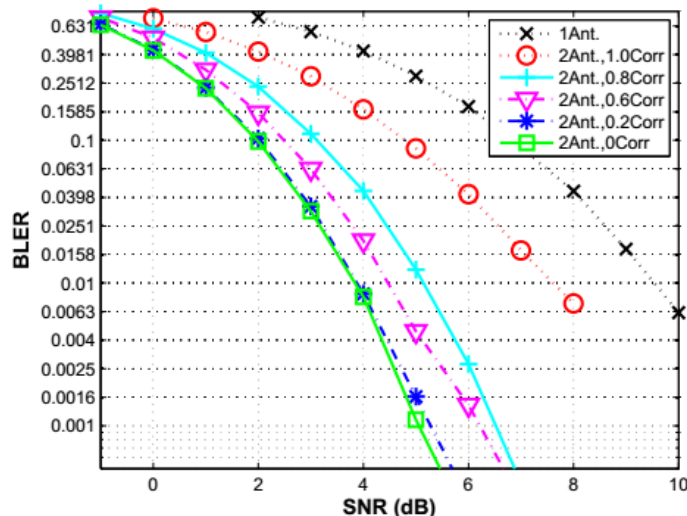


Figure 4.22: Performance gain with Rx MRC beamforming.

Before the field tests, the data throughput is tested in a lab by connecting a laptop to internet over the 5G sidelink testbed. One-way throughput of 12 Mbps is achieved which allows for stable streaming of 4K/30 fps video from YouTube. Besides, the round trip ping delay is also measured, which is stably below 3 ms. After interconnected with Bosch's see-through system during the preparation test and the final demonstration, the same low Ping delay is also achieved, as shown in Figure 4.23.

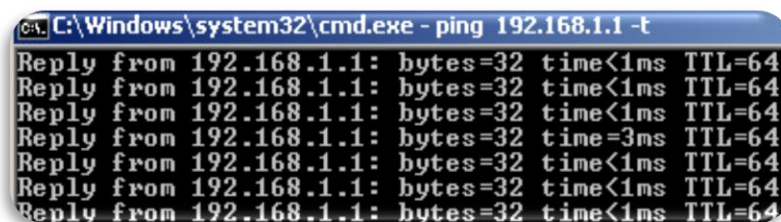


Figure 4.23: Ping delay test.

During the final demonstration which lasted for around 2 hours, we got the statistics listed in Table 4.2. We found at one particular area of the test track that strong out-of-band interference from another demonstration appears, which was observed through abnormal fast degradation of SINR values. This led to an overall BLER of $1.928e-4$. However, by excluding the interfered



transmission, much lower BLER of $3.9e-6$ was achieved which translates to high reliability over the sidelink.

Table 4.2: Packet loss measured during the final demonstration.

Results	Values
Running Time	2 hours
Total Transmitted Packet	11,023,784
Lost Packets	2,123
Packets Lost Due to Strong Interference	2,080
Overall BLER	$1.928e-4$
BLER w/o interference	$3.9e-6$

4.3 Long Range Sensor Sharing

In this section, the KPIs defined in [5GC19-D21] are analyzed for the specific scenario of the demonstration executed in UTAC. Automotive requirements and network requirements are then considered separately.

4.3.1 Automotive KPIs

The scenario showcased in the demonstration represents an urban environment. The vehicles involved have a speed between 40 km/h and 50 km/h. The intersection crossing time at these speeds is around 1 s considering one lane and the sidewalk. The relevance area is given by the line of sight of the LiDAR, up to 150 m. This is the current practical limitation of the on-board sensor used for the demonstration. LiDAR sensors are evolving and there are new announcements in the market with improved performances as it has been evoked in Section 3.3.1.

In terms of the minimum car distance, for the speed described in the urban scenario, a hard-braking deceleration could be up to 6 m/s^2 in order to avoid a crash in a situation with reduced visibility. A soft braking could be less than 2 m/s^2 . In Figure 4.24, a sequence of collision scenarios is shown where the deceleration is clearly different in the case of a collision warning (values lower than 2 m/s^2) from the case where the braking is done only when the car is in the line of sight (up to 6 m/s^2). In this figure, a sequence of periodic braking is shown. This is the complete recording, including the movements to go back in starting position. Blue curve is normal driving (move around the test track to come back to initial position after each test). Green curve is soft braking, after receiving an alert. Red curve is harder braking, where the driver needs to react to the situation cause no alert was sent. Note that the test track as clear line of sight, so the driver was not really surprised in this showcase. In real danger condition, the breaking would have been even stronger.

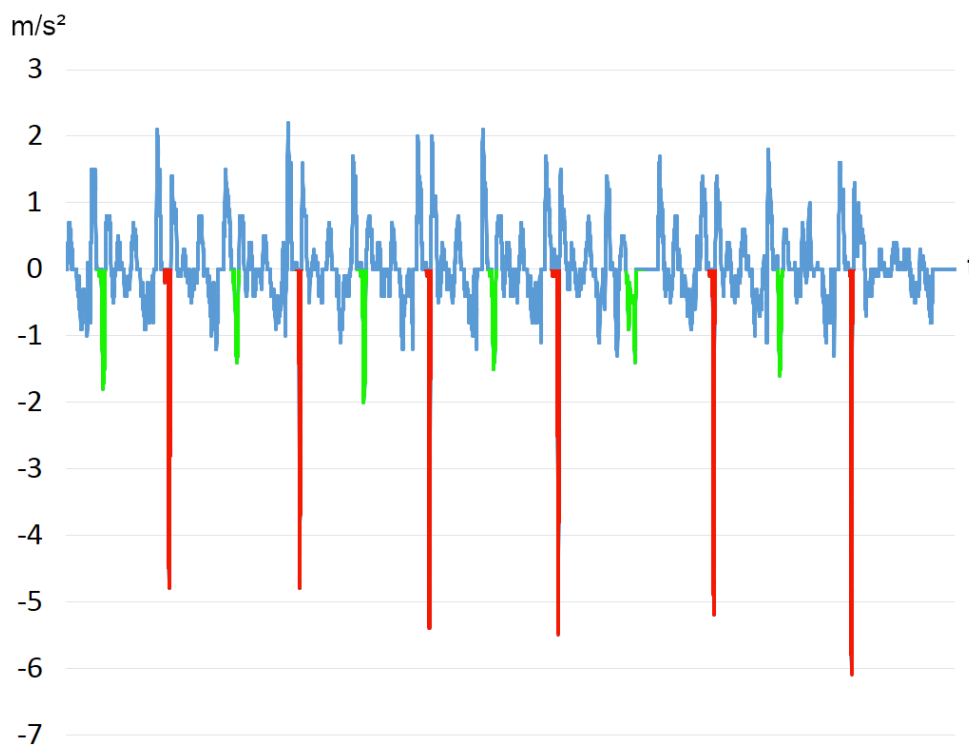


Figure 4.24: Deceleration with (in green) and without (in red) sensor sharing, over time.

The localization requirement is low for the layout of the demonstration, there is only one lane on each direction, so around 10 m accuracy should be enough in perpendicular direction of the vehicle orientation. A succession of GPS positions from the detected vehicle can be seen in Figure 4.25 as they were obtained during a test run on the UTAC TEQMO track. It was found that the key parameter to estimate is the heading, in order to clearly identify if the vehicle is approaching the intersection and represents a potential risk. In scenarios with more than one lane per direction, the localization accuracy requirement should be around 4 m in lateral, to identify in which lane the vehicle is.

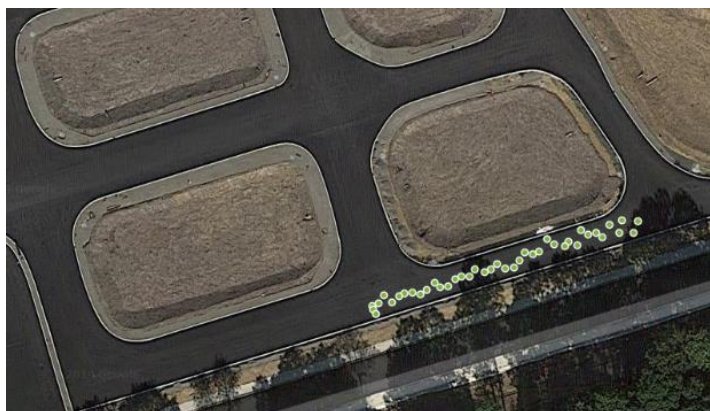


Figure 4.25: Target positions (green points) calculated by VBOX during a test run.

In Figure 4.26, an evaluation of the localization accuracy provided by the LiDAR is compared with the localization given by a connected vehicle equipped with GNSS and RTK. Twelve consecutive detections are represented in the figure. As it can be seen in the figure, the blue dots of the 200 ms detection samples match almost precisely with the information provided by the GPS. The graph shows all the two main problems explained in Section 3.3.1 about the LiDAR behavior: fake detections; which are the isolated points, and the gaps in the detection; corresponding to discontinuity in the blue lineal trajectories. Both are consequences of fusing the sensor data with no other on-board information. In the infrastructure side, to cope with these difficulties, a particle filter-based tracking algorithm is used, as explained in Section 3.3.2. This solution provides a predicted trajectory independently of the identifier assigned by the LiDAR.

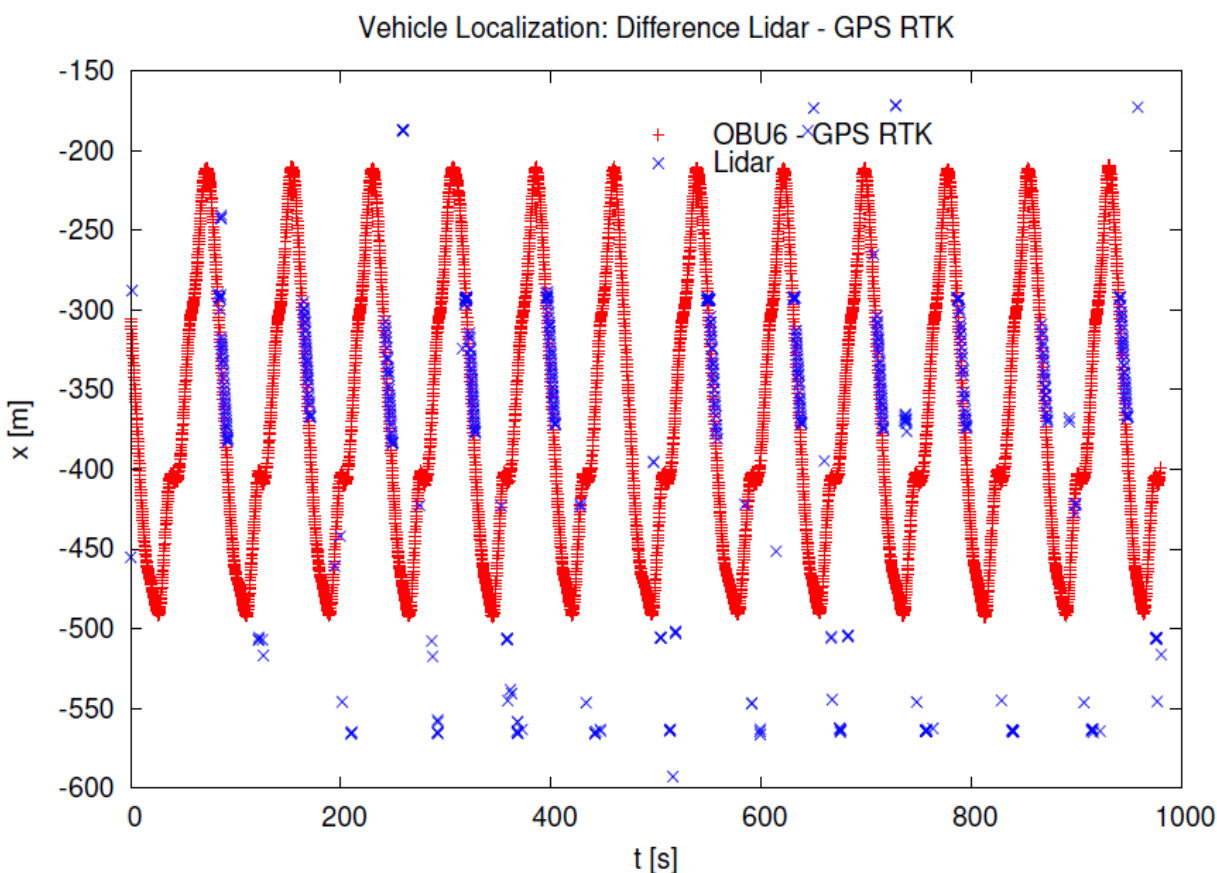


Figure 4.26: Vehicle localization, difference between LiDAR and GNSS RTK.

4.3.2 Communication KPIs

This use case benefits from the communications infrastructure provided on UTAC TEQMO and described in the introduction of Chapter 3. The latency requirement depends on the server collision algorithm estimation. With a sample available every 200 ms and with a server processing collision window of 200 ms, a value of less than 50 ms should be kept. The key factor is to keep as much time as possible between samples for the algorithm calculations. The delay requirement will be stricter for higher speeds and more vehicles. Linked to the low latency, a reliability of at

least 99% becomes crucial in order to get the localization samples within a fixed time window, helping in the trajectory estimation. In the tests done using the cloud deployment described in Section 3.6.1 (around 100 ms delay), the delay variation between largest and smallest measurement (over 50 ms) made it impossible to obtain a successful prediction with good accuracy.

The service data unit used in the demonstration respects the overall message format defined as a global brick for the information exchange. Every 200 ms the connected vehicle sends a message with the information as defined in Section 3.3.1, containing the service data unit: 300 bytes including the position, the heading, the speed, the acceleration and the yaw rate. In the case of the detected vehicle, the same frequency is kept using the common message format but only position, heading and speed are estimated based on the LiDAR information.

This leads to a very low data rate, around 10 kbps in the uplink, from each vehicle, the connected and the detected one. Even lower in the case of the downlink, where only the alert is sent towards the affected vehicle.

The communication range is limited in the case of the detected vehicle for the LiDAR line of sight. As explained in Section 3.3.1, this can go a little bit beyond 100 m. Once the information is available in the network, there is no range limit.

The availability of the system should be high for cost-benefit reasons, but it is not safety critical. If the system does not work the risk of the maneuver shall be mitigated by the driver or by the AD system by adapting the speed to the angle vision of the driver or the field of view of the on-board sensor. As a non-functional requirement, the vehicle must be aware if the system is available or not. In the demonstration, with a driver receiving HMI alerts, the car must show if the collision detection alert is working or not. For this purpose, a specific icon was included in the main HMI screen, shown in Figure 4.27.



Figure 4.27: Icons in main HMI showing the availability status of the system.

4.3.3 Qualitative KPIs

This service is not very demanding in terms of security requirements. Privacy is not affected, because the detection is only based on the trajectory, not even the identifiers assigned by the LiDAR are considered and only from this information there is no way to identify the characteristics of the detected car. Therefore, the required confidentiality for the system is low, the information shared with the server is not sensitive and the message sent towards the vehicle contains no personal information. On the other side, authentication and integrity must be guaranteed, in order to be sure that the collision alert comes from the certified server and the content of the alert is not modified from the original message sent.

The cost and the power consumption of the infrastructure elements needed cannot be neglected. An edge cloud environment is needed, with the suitable climate conditions. In the case of the demonstration, with two cars involved and the collision prediction made in real time, the application is consuming significant machine resources, with three processes in parallel, two for tracking and one for prediction of collision. With more complex traffic scenarios, server performance may become even more important.

4.3.4 Evaluation and Conclusions

The sensor sharing showcased in this demonstration provides a similar effect of an increase in the field of view, in situations where the sensor perception or human driver line of sight is limited due to the scenario layout. It is similar to the see-through use case, but instead of seeing through vehicles based on V2V communication, in this case there is an extended vision through obstacles (as shown in Figure 4.28) thanks to the connectivity infrastructure and the cooperative perception.

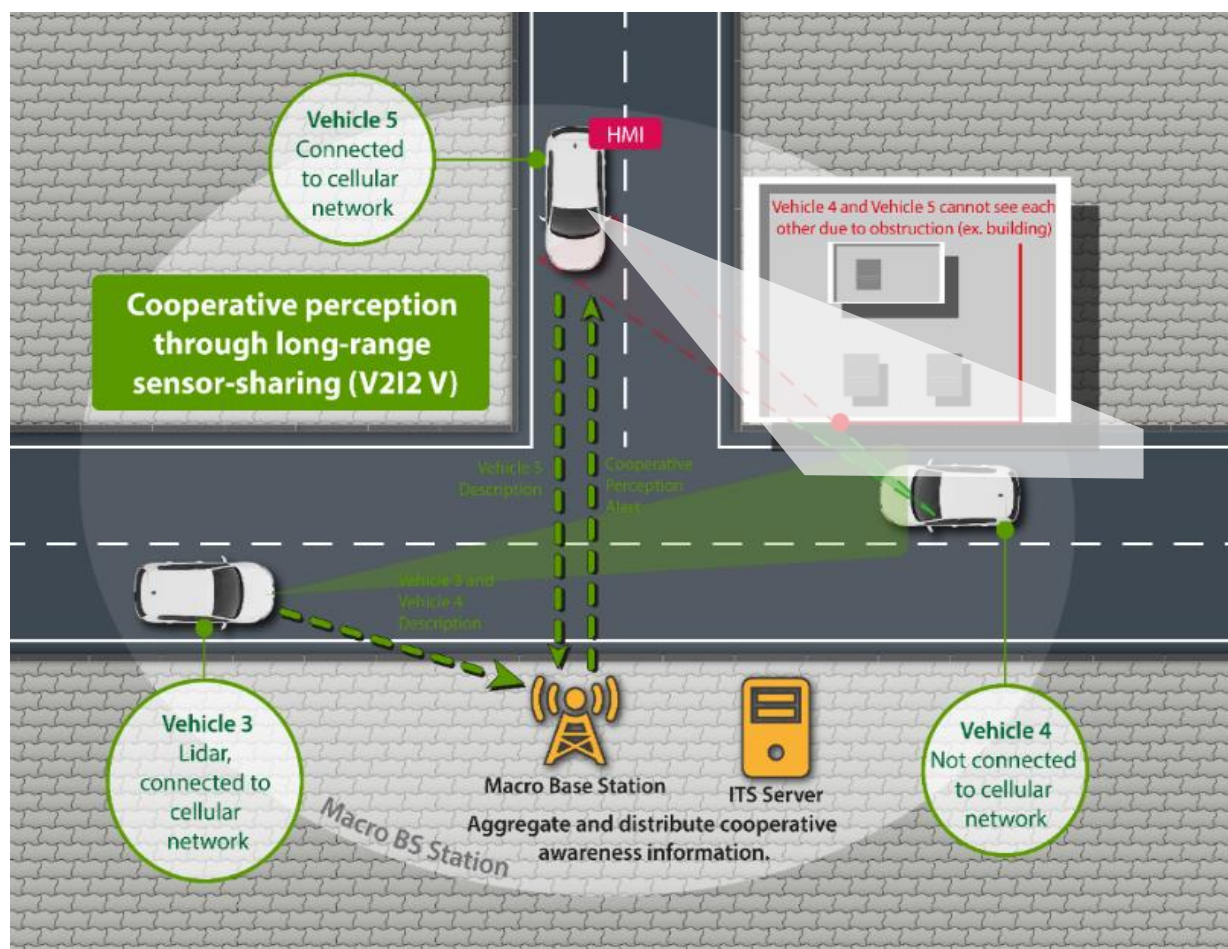


Figure 4.28: White triangle representing the field of view extension thanks to connectivity.

Results show that the notification alert arrives at least 15 m before the collision providing a safe and comfortable braking distance.

Figure 4.29 shows the recorded trajectories of vehicles (black and red line) as well as their anticipated positions in the next six seconds based on the constant velocity model (yellow and purple squares). We also see the probability density function of the future position after six seconds. We plot for both vehicles the lines corresponding to equal pdf values (yellow-green-blue scale, in descending strength). As the maximum of the function (yellow area) of the vehicles overlap, the determined collision probability will be very high.

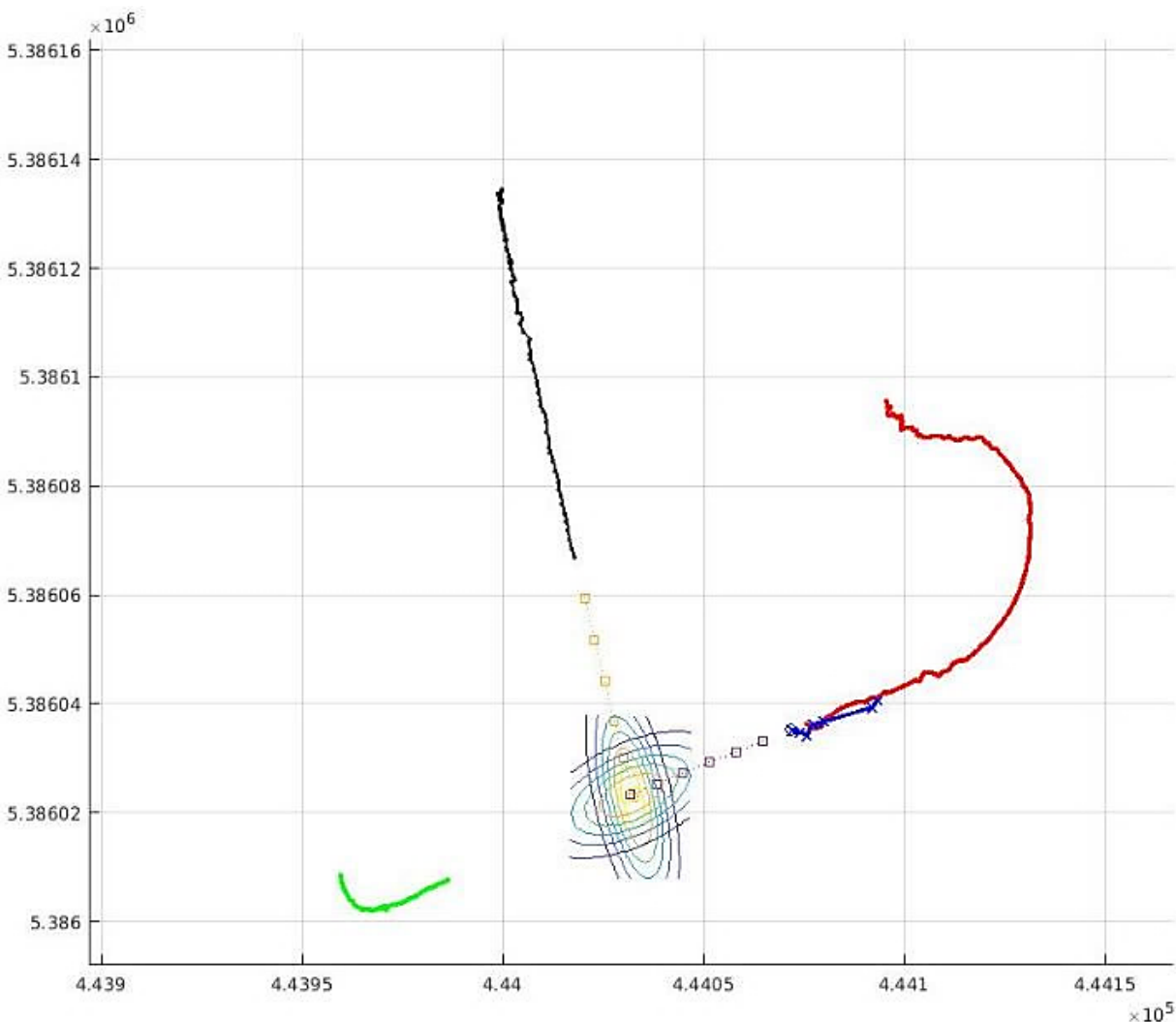


Figure 4.29: Measured trajectories and anticipated positions with probability density functions (yellow-green-blue scale, in descending strength).

Figure 4.30 shows the calculated collision probability versus collision forecast times (or time windows) between 1 s and 6 s. The time of the collision that would have happened without braking is indicated by the red dotted line at $t = 8$ s. In case of a collision forecast time of 2 s (yellow line) the system correctly anticipates the collision with a probability of 80%, and the warning message would appear at $t = 6$ s. We observe that, with longer collision forecast times, the probability

becomes smaller. The reason is the increasing uncertainty reflected by a flatter shape of the probability density function of the state variables position and speed.

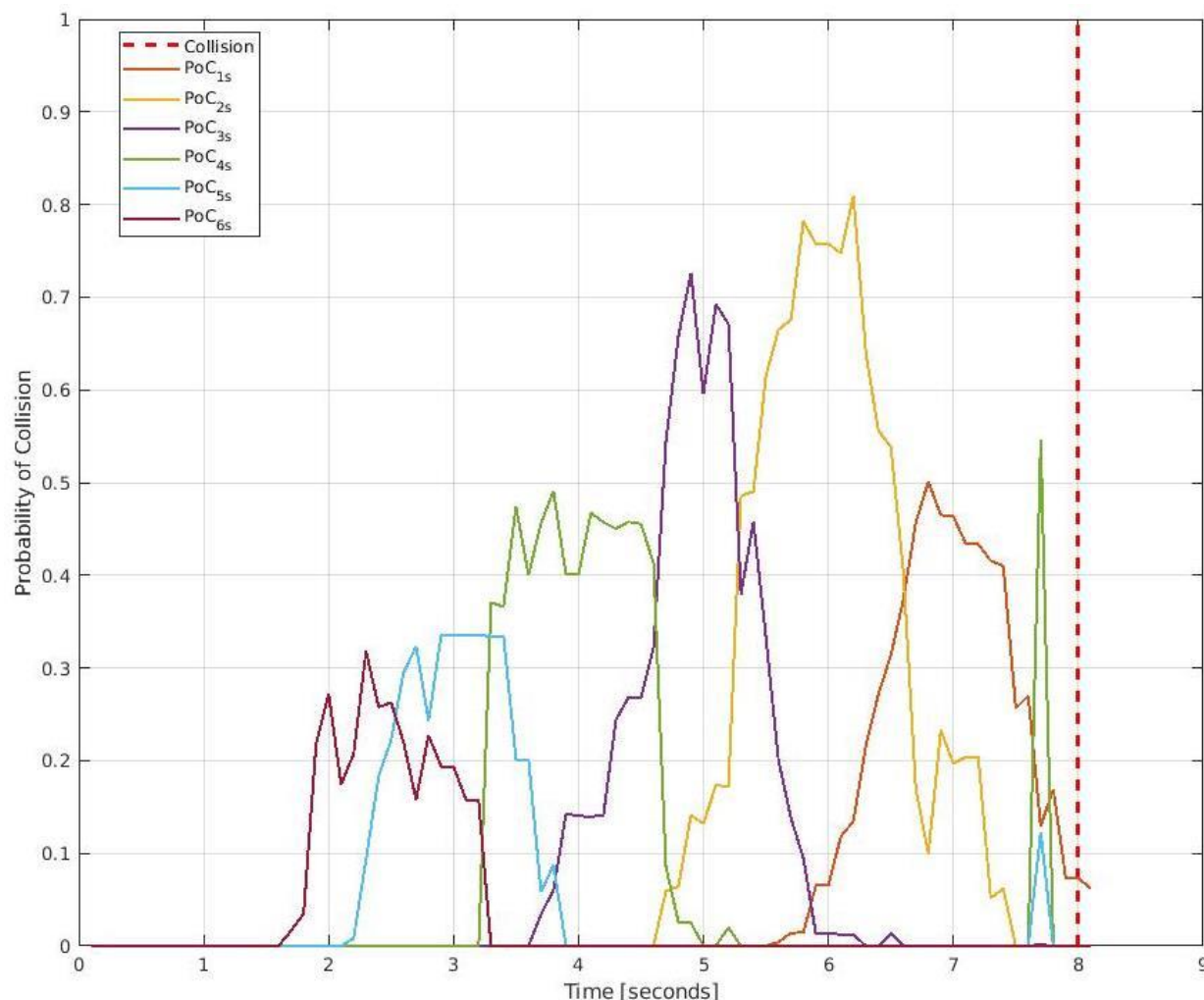


Figure 4.30: Collision probabilities versus collision forecast time.

The appropriate setting of the collision forecast time and the alarm threshold is not straightforward. Generally, we wish an early alarm to increase safety and comfort. According to Figure 4.30, this would require a relatively low threshold, e.g. 30% for collision forecast time 6 s. The drawback would be a high false alarm rate, i.e. we receive many alarms in non-critical situations, which may make the driver stops taking these warnings seriously. Vice versa, a high threshold eliminates false alarms at the cost of missed detection, i.e. even in case of a critical situation no alarm will be sent, or it will come too late. As it can be seen in Figure 4.30, a threshold of 80% will trigger an alarm only 2 s before the collision.

In Section 4.4, we present complementing simulation results for a deeper analysis of the dependency between collision forecast time and collision threshold and their impact on false alarm rate and the precision of the collision warning system.

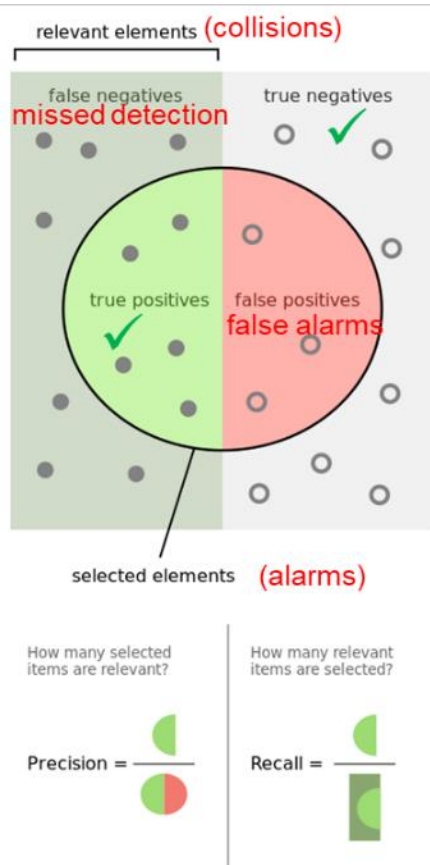
4.4 Vulnerable Road User Protection

In the VRU task several measurements on the track are conducted in May and in June before the demonstration event to provide algorithm input test data sets. In tuning algorithm parameters, the final demonstration scenarios are tested before and evaluation was done, see following sections.

4.4.1 Key Performance Indicators

For the demonstration evaluation the following KPIs are of interest:

- For the real-time localization of road users with the 5G radio network the achievable accuracy is of interest. It is defined as the mean error of position estimates from the true position.
- Concerning the collision warning system, we determine the false alarm rate, recall, precision and F1-score as defined in Figure 4.31.



False alarm rate is the ratio of false positives per total number of warnings.

Precision is the number of true positives per total number of warnings

Recall is the number of true positives per total number of relevant events (collisions)

F1-score combines precision and recall in one single metric, it is defined as $F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

Figure 4.31: Definition of the KPIs for the collision warning system.

4.4.2 Evaluation Results

Positioning Accuracy

We determined the positioning accuracy at eleven reference points on the test track. As illustrated in Figure 4.32, the points C0 - C7 are located on the trajectory of the car in 5 m distance, and the points V2, V4, and V6 are located on the trajectory of the VRU in 5 m distance. Except C0, all points are assumed to be in the coverage area of all six antenna panels. For the reference points we assume the positions measured with the Leica Zeno20 to be the true positions.

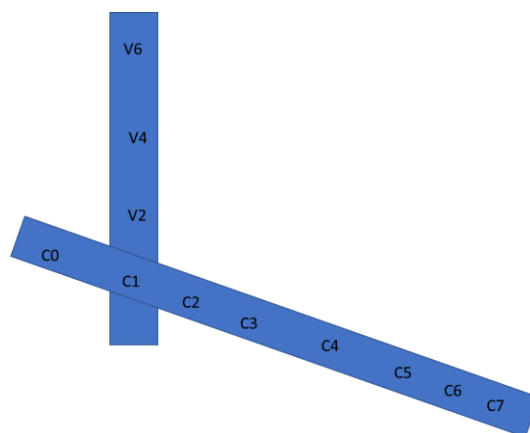


Figure 4.32: Illustration of reference point positions on the test track.

Figure 4.33 shows a comparison of the accuracy of 5G radio-based positioning (red) with GPS (blue) averaged over measurements at the eleven reference points. The GPS measurements were done with a Sony Xperia Z1 Compact and the app Locus Map. It shall represent a typical positioning accuracy we can expect from a handheld device of the VRU. The stepwise behavior of the blue curve indicates some unknown post processing of the raw sensor data. Anyway, we can observe a benefit of the 5G radio-based positioning for both the 50%-quantile (1.5 m vs. 1.8 m) and the 80%-quantile (2.25 m vs. 3.25 m). However, it is important to note that these results just reflect the situation at the eleven reference points. They are not universally valid.

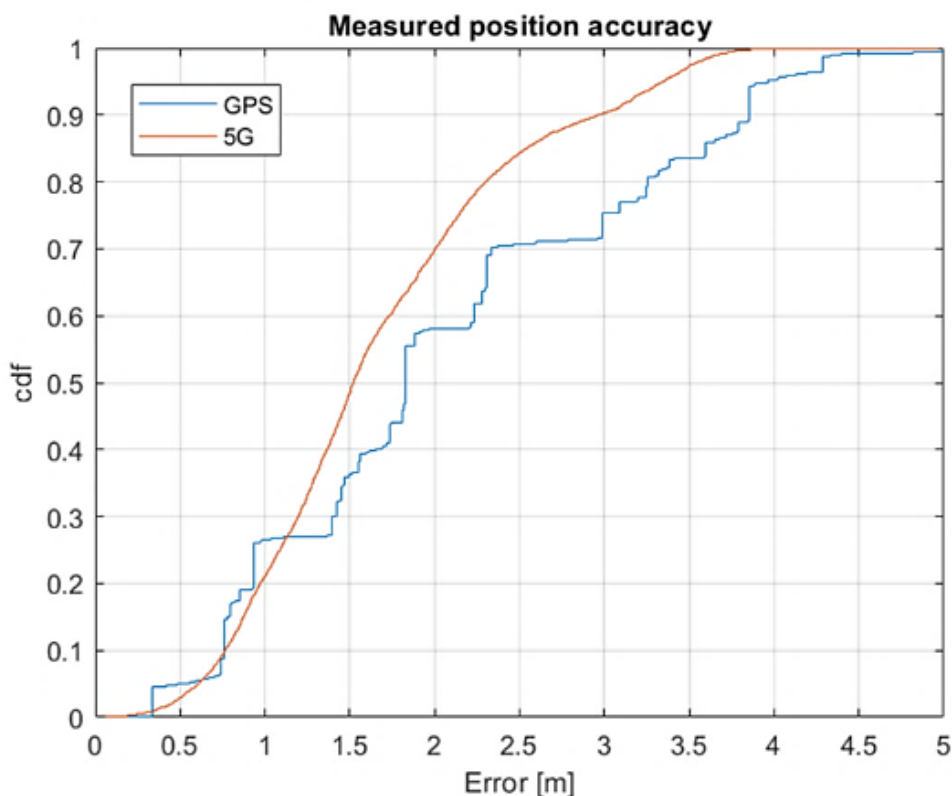


Figure 4.33: Measured accuracy of the 5G radio-based positioning in comparison to GPS averaged over eleven reference points compared to ground truth.

Reliability of Collision Warnings

As it was not possible to reproduce a sufficiently high number of critical and non-critical events in the given time to get statistically relevant results for false alarm rate, precision, recall and F1-score (see Section 4.4.1) we present in the following samples of our simulation-based evaluation presented in [5GC18-D32, 5GC19-D33].

Figure 4.34 shows F1-score, precision, recall and false alarm depending on the time window for alarm thresholds 50% and 90%. A warning message is sent when the collision probability exceeds the threshold. Some discussion about an appropriate setting of these parameters to exploit the service in the best possible way was presented in Section 4.2.1.

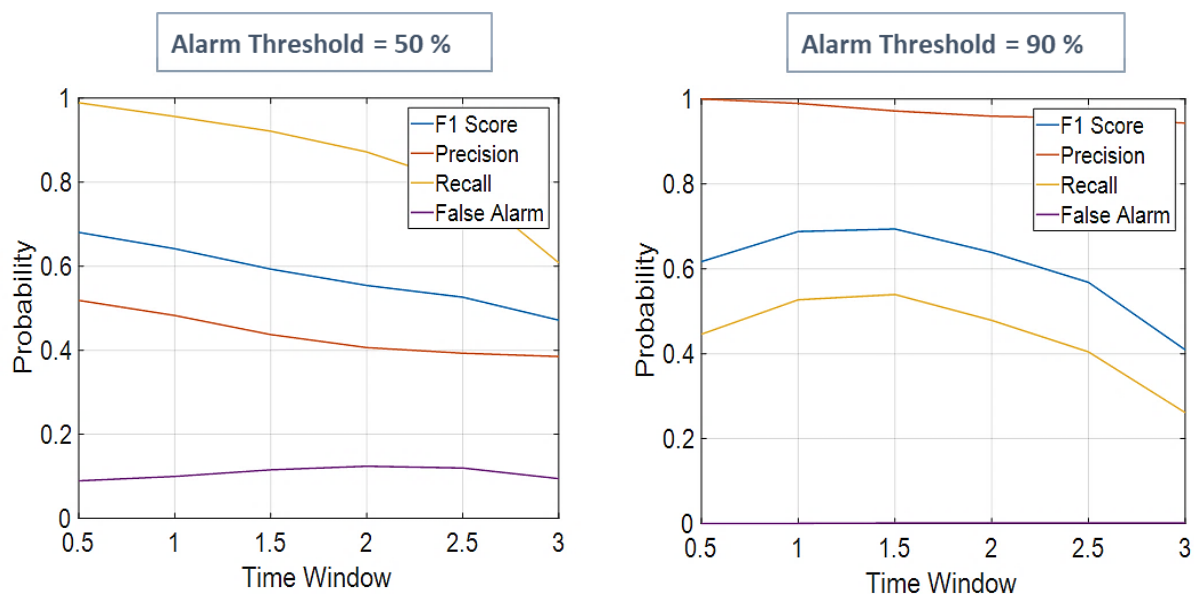


Figure 4.34: Results of simulation-based evaluation of the collision warning system for alarm thresholds 50% and 90%.

We observe that the high threshold (90%) can almost eliminate false alarms (purple curve). However, at the same time the number of missed detections increases, which is visible in the smaller recall (yellow curve). We also observe a performance degradation with increasing time window. In Section 4.2.1, we saw that the determined collision probability shrinks with longer time windows which is due to the increasing uncertainty in the anticipated position of the users. According to the F1-score, a good compromise may be a time window between 1 s and 2 s with alarm threshold 90%. It is also recommended to observe the collision probability at different time windows and make the decision whether an alarm is sent depending on a combination of them.



5 Conclusions and Learnings

The 5GCAR project successfully demonstrated four automotive demonstrations, each representing different use cases and corresponding environments. To that end, various 5G concepts were applied to the demonstration design of these use cases and implemented in a realistic environment on an automotive test track. More importantly, an end-to-end implementation for each use case was done, enabling a detailed analysis of the feasibility of said solutions, focusing on the identification of challenges to be addressed in a real deployment, which enabled a great amount of knowledge transfer between partners of different industries and of different background. The main findings are summarized in the following for each demonstration.

For the Lane Merge Coordination demonstration, several KPIs have been evaluated, with a focus on latency and reliability in the communication domain, and a focus on acceleration and inter-vehicle distance in the automotive domain. In general, most KPI requirements have been either fulfilled, or were not feasible to be evaluated in the scope of 5GCAR. For the latency requirement, a concrete evaluation of the one-way delays was not possible due to lack of time synchronization, but RTTs were evaluated in detail. While the network RTT is a significant component, also aggregation of data for the data fusion step is a major contributor. Furthermore, it was found very important to investigate configuration optimizations on higher layers outside the radio access network, and collocating involved software components, preferably in an edge cloud. Finally, it was shown that the latency greatly increases at higher percentiles, as occasional effects from multi-threaded operating systems and retransmissions become significant.

For the See-Through Sensor Sharing, the KPIs introduced in [5GC18-D51] have been evaluated with regards to communication and application aspects. The application KPIs consist of image quality, fit of the see-through overlay and latency of the video stream. These result in the following communication KPIs: high throughput, low packet error rate and low latency. Furthermore, different multi-antenna setups have been evaluated in simulation.

For the Long-Range Sensor Sharing demonstration, the evaluation focused on the enhanced driving comfort with respect to reduced deceleration in a collision case due to earlier warnings, and on the collision prediction timing. It was shown that the LiDAR produced useful detections up to distances over 100 m, and therefore, collisions could be predicted approximately 2 s ahead of collision, with a confidence of over 80%. The dynamics between collision forecast time and prediction confidence was presented in more detail.

For the Vulnerable Road User demonstration, the evaluation focused on the positioning accuracy of the 5G-NR prototype, and the collision prediction performance including the reliability of alarm messages. The positioning accuracy was found to be superior to GPS at higher quantiles in the concrete given scenario, but it should be noted that this is not necessarily universally valid. For the reliability of alarm messages, the dynamics between collision forecast time and collision warning reliability was presented, and it was shown that a time window between one or two seconds was identified as a good compromise, allowing a collision warning reliability of approximately 90%.



Aside from the use case-specific findings, several tools and ways of working proved very useful, while having some insights earlier in the project would have been beneficial.

Having a common messaging framework enabled sharing insights and testing support between the different demonstrations. Aside from partially sharing robustness testing, tools developed for individual demonstrations could be leveraged by other demonstrations easily. Finally, the complexity on the client side was greatly reduced, while having a large degree of flexibility when sharing test clients/cars between demonstrations.

One of such shared tools is the 24/7 cloud deployment of a messaging framework and used for testing and integrating software components largely independent from other partners. Through the shared message format, compatibility checks were facilitated across use cases. Retrospectively, introducing sanity checks of the various message parameters (e.g. speed) in a central component at an early point would have facilitated the tracking of unexpected behavior in the system, saving troubleshooting time. Furthermore, using a sharable schema for the messages (JSON schema, protocol buffer, etc.) would have simplified discussions on message format details, reducing room for interpretation and implementation mistakes.

Also, having a central aggregation and evaluation log system was very helpful for noticing, understanding, and locating performance issues in the messaging system. As this is not directly coupled to making a demonstration work, this piece was postponed longer than it should have been, which is why some performance flaws were detected quite late in the project. Retrospectively, having a mature evaluation framework at the beginning of the field tests would have saved a lot of work in the testing and evaluation procedures. Also, it has proven beneficial to include detailed reference information in the logs, so that the context of each measurement is unambiguously clear, including time and location. Finally, precise time synchronization (on millisecond level) would have significantly reduced the evaluation effort, and it should be clearly stated that NTP-based time synchronization is not sufficiently accurate when using a cellular radio access network such as LTE or 5G-NR.

In the field tests, it was clear that a central coordination is needed for the tests with multiple vehicles. Next to increased efficiency, this also facilitates the detailed documentation of tests, including detailed descriptions (test scenario, timeframe, test configuration, applied changes, observations, etc.). Preparing and using a documentation template is recommended. For communicating between the control room, the cars, and other locations, a standard telco bridge has proven most efficient, due to integration with hands-free systems, headsets, conferencing systems, etc. Furthermore, it has proven very useful to have a good monitoring device capable of displaying which cars are receiving what in the control room and make it easier to understand for all people involved in the test what is happening.

Finally, when working with several prototypes in the field tests, having test days alternating with development and integration days is very time- and cost-efficient. Naturally, issues in various components are identified, and some time is needed to identify collected logs, to understand issues behind, and to develop improvements.



6 References

- [3GPP19-23401] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", June 2019.
- [3GPP19-36521] 3GPP TS 36.521, "User Equipment (UE) conformance specification; Radio transmission and reception", June 2019.
- [3GPP19-38101] 3GPP TS 38.101, "NR User Equipment (UE) radio transmission and reception", June 2019.
- [5GC18-D32] 5GCAR D3.2, "Report on Channel Modelling and Positioning for 5G V2X", November 2018.
- [5GC18-D51] 5GCAR, Deliverable D5.1 Version 1.0 "Demonstration Guidelines", May 2018.
- [5GC19-D21] 5GCAR, Deliverable D2.1 Version 2.0 "5GCAR Scenarios, Use Cases, Requirements and KPIs", February 2019.
- [5GC19-D33] 5GCAR D3.3, "Final 5G V2X Radio Design", May 2019.
- [AMG+02] Arulampalam, M. S., Maskell, S., Gordon, N., Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Transaction on Signal Processing, vol. 50, February 2002.
- [AND08] A. Howard: "Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles", In International Conference on Intelligent Robots and Systems. IEEE, 2008.
- [FEAT94] J Shi and C Tomasi, "Good features to track", In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600, 1994.
- [ICCVE19] K. Cordes, N. Nolte, N. Meine, and H. Broszio: "Accuracy Evaluation of Camera-based Vehicle Localization", In International Conference on Connected Vehicles and Expo (ICCVE). IEEE, 2019.
- [LBB+18] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation using SUMO", IEEE Intelligent Transportation Systems Conference (ITSC), November 2018.
- [MOT17] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by detection without using image information. In 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6, 2017.
- [NGSIM18] Next Generation Simulation (NGSIM) program, "Vehicle Trajectories and Supporting Data", September 2018.
<https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajectory/8ect-6jqj>
- [SAM15] Samsung Newsroom: "The Safety Truck Could Revolutionize Road Safety".
<https://news.samsung.com/global/the-safety-truck-could-revolutionize-road-safety>



- [SKS+10] K. Shin, J. Kim, K. Sohn, C. Park, S. Choi, "Online Gaming Traffic Generator for Reproducing Gamer Behavior". In: H.S. Yang, R. Malaka, J. Hoshino, J.H. Han (eds) Entertainment Computing (ICEC 2010), vol 6243, pp. 160-170, 2010.
- [UTAC-TEQMO] UTAC Technology Center for automated and connected mobility.
<https://www.utacceram.com/teqmo>
- [VERM02] H. Vermeille: "Direct transformation from geocentric coordinates to geodetic coordinates", Journal of Geodesy 76, pp. 451-454, 2002.
- [VISAPP19] K. Cordes and H. Broszio: "Constrained Multi Camera Calibration for Lane Merge Observation", In International Conference on Computer Vision Theory and Applications (VISAPP), SciTePress, pp. 529-536, 2019.
- [VOLVO-BROKER] OpenSource software for reading in-vehicle data from Volvo vehicles.
<https://github.com/volvo-cars/signalbroker-server>
- [WR09] The White Rabbit Project. <https://ohwr.org/projects/white-rabbit.2009>
- [YOLOv3] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.



A Technical Component Details

In this Annex, more technical details are provided on some aspects that were only touched superficially in the main document.

A.1 Vehicle Position Extrapolation in Data Fusion

One step of the Data Fusion component (introduced in Section 3.1.2) in the Lane Merge Coordination use case was the Data Synchronization of different sources, i.e. extrapolating data to an equal point in time. The use of a GPS point extrapolation improves the freshness of the positioning data by extrapolating vehicle positions relative to their base coordinates and speeds. To understand: this quantum represents a delta on earth of 57 cm on Sweden, but of 1.11195 m at the Equator. It depends on the delta time of the "tick" too: at 100 km/h a car travels at 27.7 m/s. So, if the "tick" is at 1 s, the correction can be an improvement. The tick must also stay higher than the position update frequency from the car, to be efficient and not compute many times an extrapolation on the same original point. Moreover, the GPS coordinates are precise with 7 significant digits. At 80 km/h, with a 15 ms tick, the distance travelled is 33.3 cm: it justifies this GPS correction.

We implemented the pretty simple and relatively fast Euclidian geometric method. The position $t_0 + \Delta t$ was a calculation of the distance traveled during a lapse of time ' Δt '. We used speed, heading (angle) and acceleration to get this value.

- h = heading in degree trigonometric rotation and 0° or 360° for North
- $P(\text{lat}(0), \text{lng}(0))$ = Represent initial position on GPS WGS 84 (decimal degrees)
- α = Acceleration in m/s^2
- Δt = time passed
- $P(\text{lat}(\Delta t), \text{lng}(\Delta t))$ = final position on GPS WGS 84 (decimal degrees)
- d = distance travelled in m
- R = earth radius in m

Distance travelled during Δt : $d = v_0 \cdot \Delta t + (\alpha \cdot \Delta t^2) / 2$

Distance projected on GPS coordinates:

$$\text{lat}(\Delta t) = (180(\arcsin(\sin((\text{lat}(0) \cdot \pi) / 180) \cos(d/R) + \cos((\text{lat}(0) \cdot \pi) / 180) \sin(d/R) \cos((h \cdot \pi) / 180)))) / \pi$$
$$\text{lng}(\Delta t) = (180(((\text{lng}(0) + \pi) / 180 + \text{atan2}(\sin((h \cdot \pi) / 180) \sin(d/R) \cos((\text{lat}(0) \cdot \pi) / 180), \cos(d/R) - \sin((\text{lat}(0) \cdot \pi) / 180) \sin((\text{lat}(\Delta t) \cdot \pi) / 180)) + \pi) \% 2\pi - \pi)) / \pi$$

NB: the latitude must be calculated before longitude.

A.2 ELK Stack Interfaces

This section introduces the interfaces for aggregating all logs in the elastic stack, which was introduced in Section 3.6.2. Figure A.1 depicts the communication between the software components and LogStash, when reporting logs.

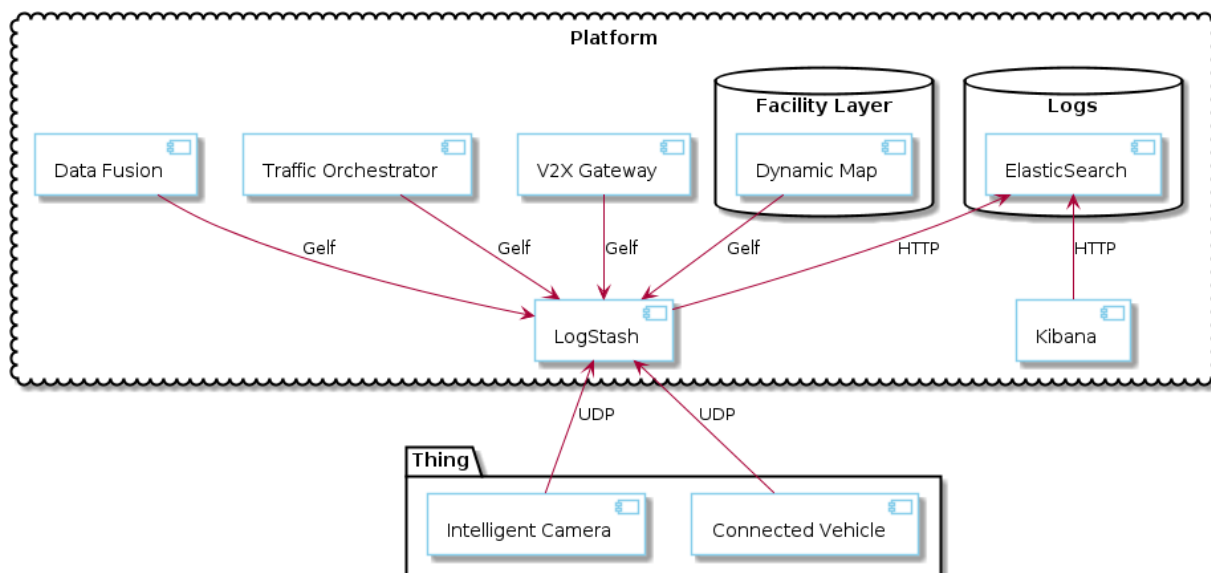


Figure A.1: Each component sends its message to the ELK system.

Figure A.2 illustrates how Kibana and Logstash use the HTTP Elasticsearch API.

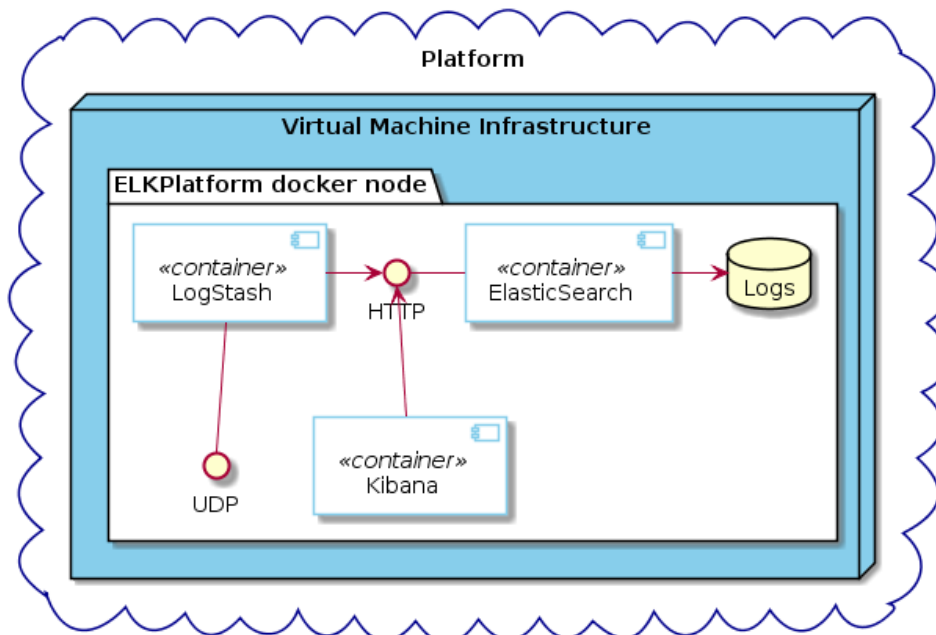


Figure A.2: All the logs are written into a same database.

In the project, the Elastic Search database was stored on the host through a docker mount point. Figure A.3 depicts the communication between the software components running as docker containers, and the ELK platform, using Gelf.

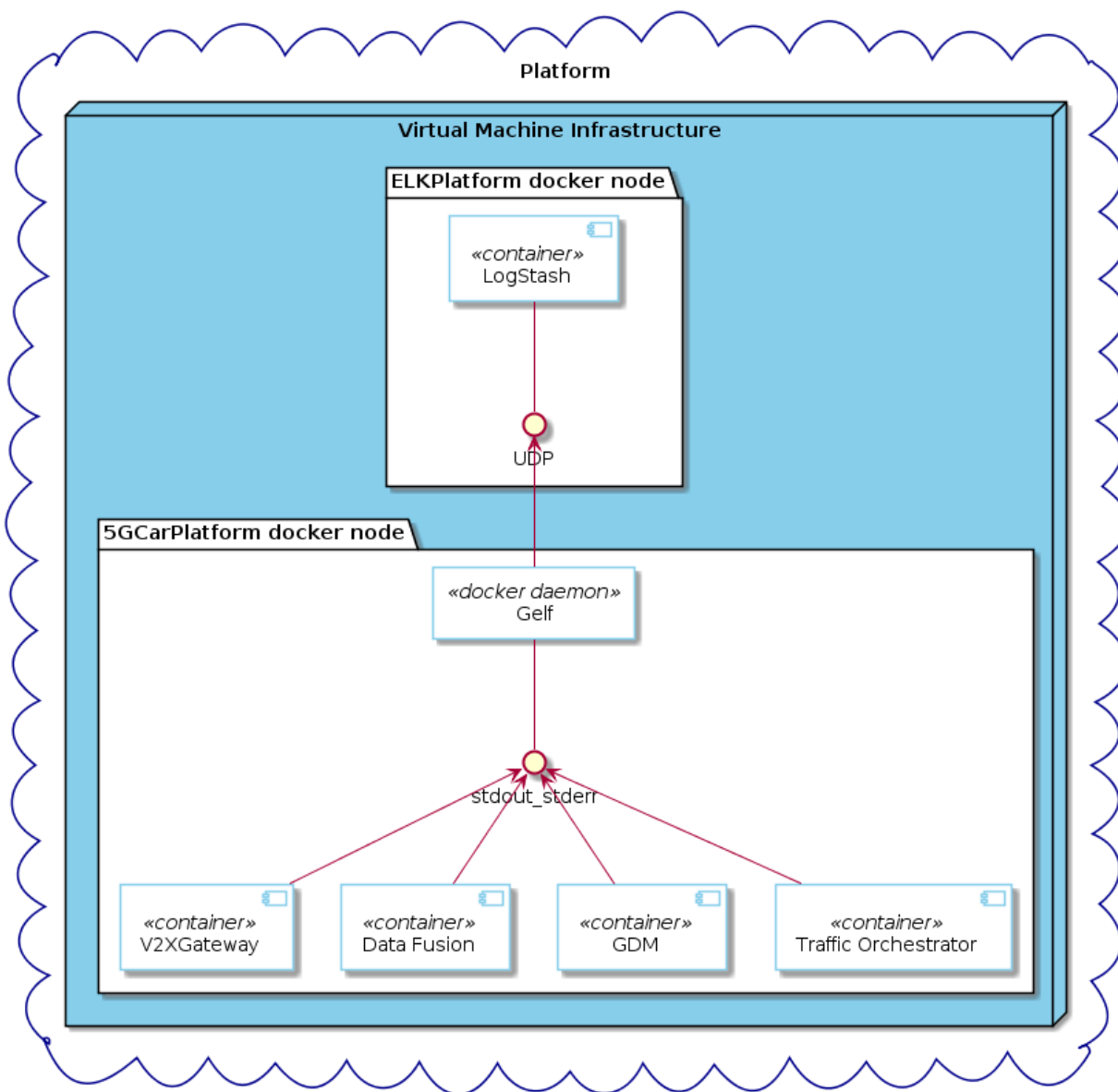


Figure A.3: Each component uses stdout and stderr to write logs.

All the containers (from a Cloud platform or from any server at the test site) used the Gelf docker daemon to share their logs. It permitted to preprocess logs as soon as possible. Each component wrote on stdout for a simple log, and optionally on stderr for an error. Figure A.4 depicts the communication between other components and the ELK platform using UDP.

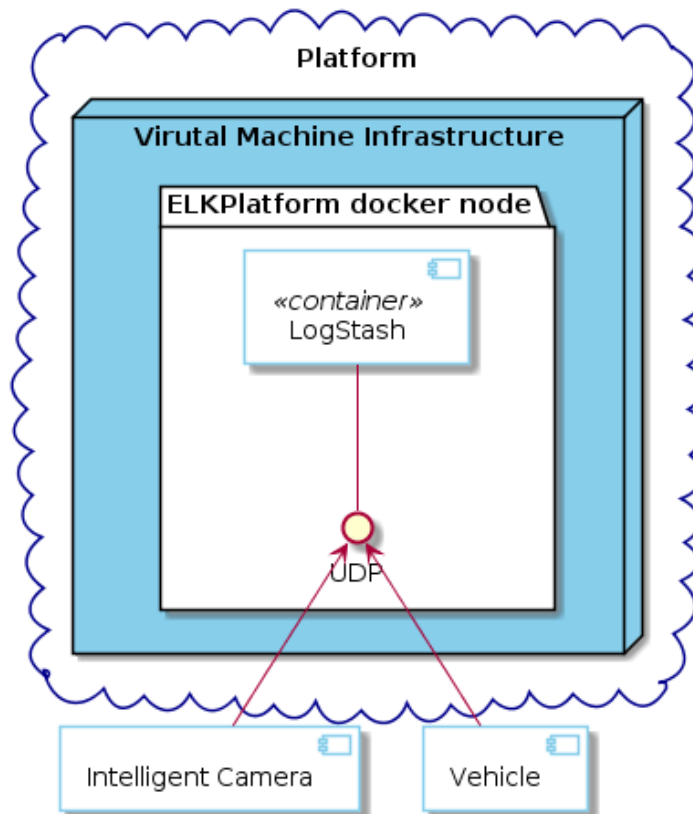


Figure A.4: Some components used UDP to send logs to LogStash.

These components used a UDP connection to upload their logs manually. We studied other protocols to communicate with Logstash (see <https://www.elastic.co/guide/en/logstash/current/input-plugins.html>), but it would complicate the deployment of the platform.

A.3 Data Injector Mechanisms

The Data Injector, introduced in Section 3.6.2, can be used to send input or accept output to/from various components, by supporting not only basic message parsing and creation, put also e.g. subscription mechanisms.

Figure A.5 and Figure A.6 illustrate how the injector can be used to provide input to the Data Fusion, in place of the V2X Gateway, in order to test the Data Fusion and subsequently the Dynamic Map.

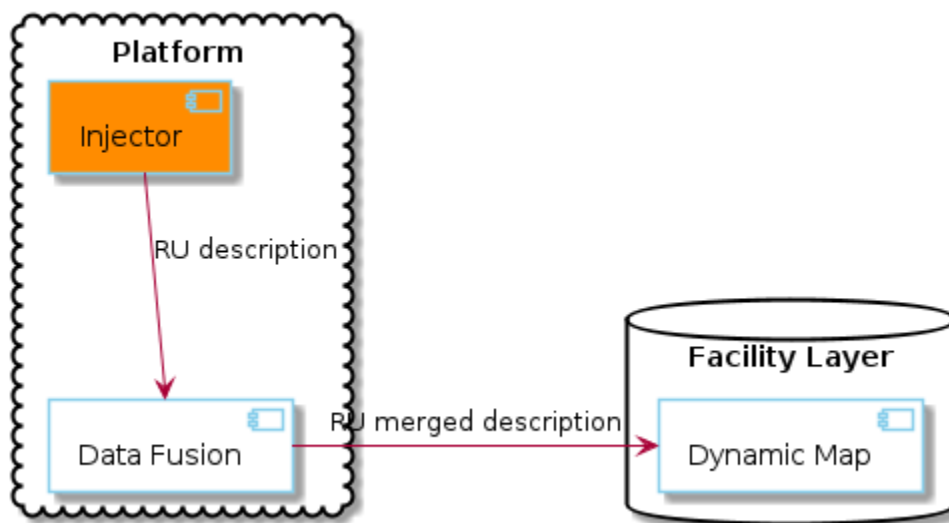


Figure A.5: The injector calls the Data Fusion server from the same platform.

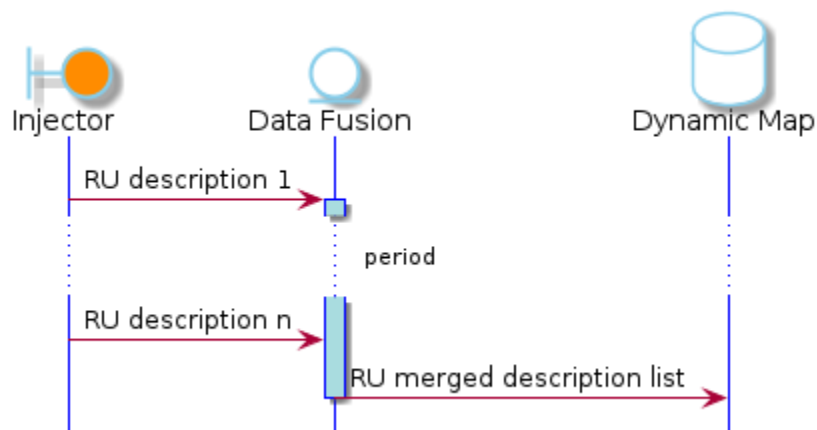


Figure A.6: The injector calls the Data Fusion server many times.

Figure A.7 and Figure A.8 illustrate how to use the injector to simulate the complete interaction with the Traffic Orchestrator as a stand-in for the V2X Gateway.

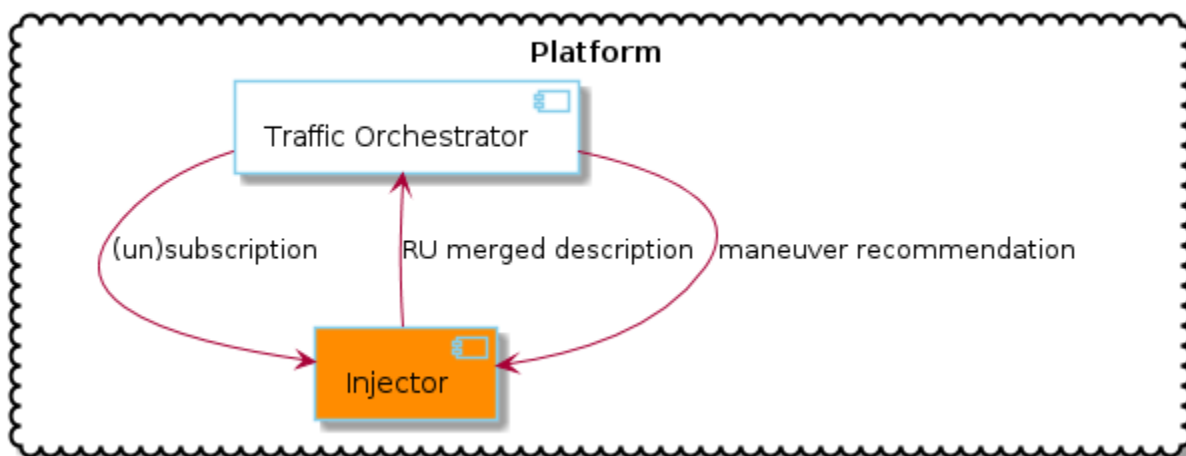


Figure A.7: The injector exchanges with the Traffic Orchestrator many types of message.

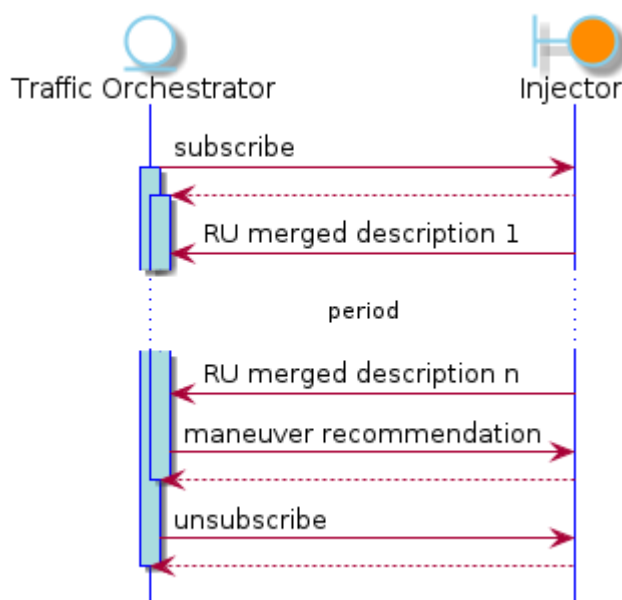


Figure A.8: The injector allows validating the result of a maneuver recommendation creation.

Figure A.9, Figure A.10, and Figure A.11 illustrate how to test the complete communication involving the Data Fusion, the Traffic Orchestrator, the V2X Gateway and the Dynamic Map, by acting as a set of vehicles sharing their status information. We used this global simulation on weekly online integration sessions and on the UTAC test track in order to verify full functionality.

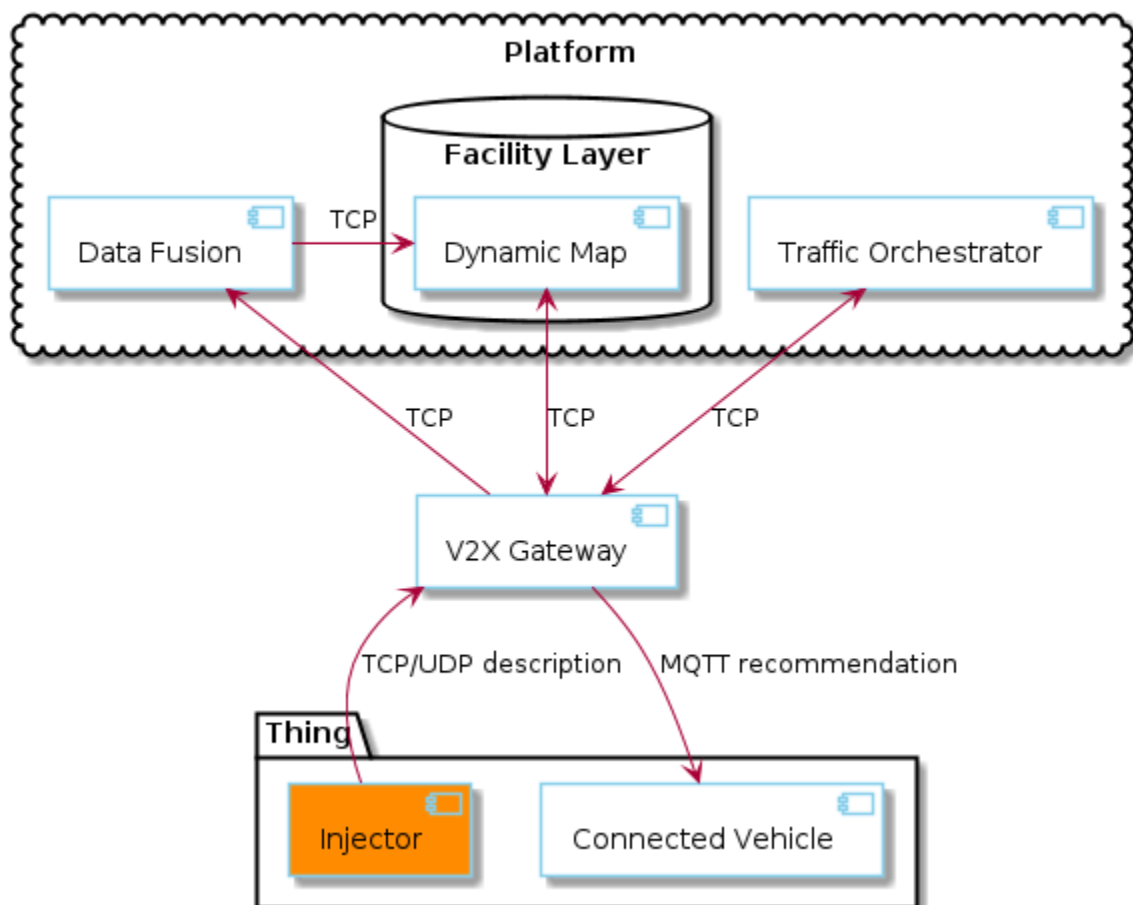


Figure A.9: The injector allows simulating things on a real final architecture just without the Intelligent Camera System.

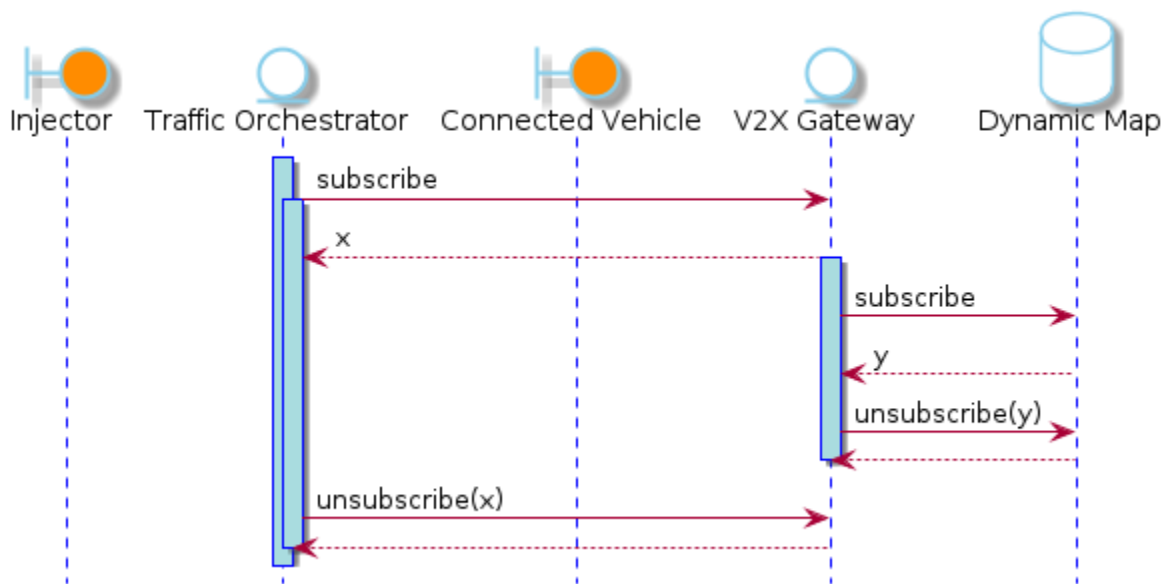


Figure A.10: The injector needs of complete initialization of the platform before sending data.

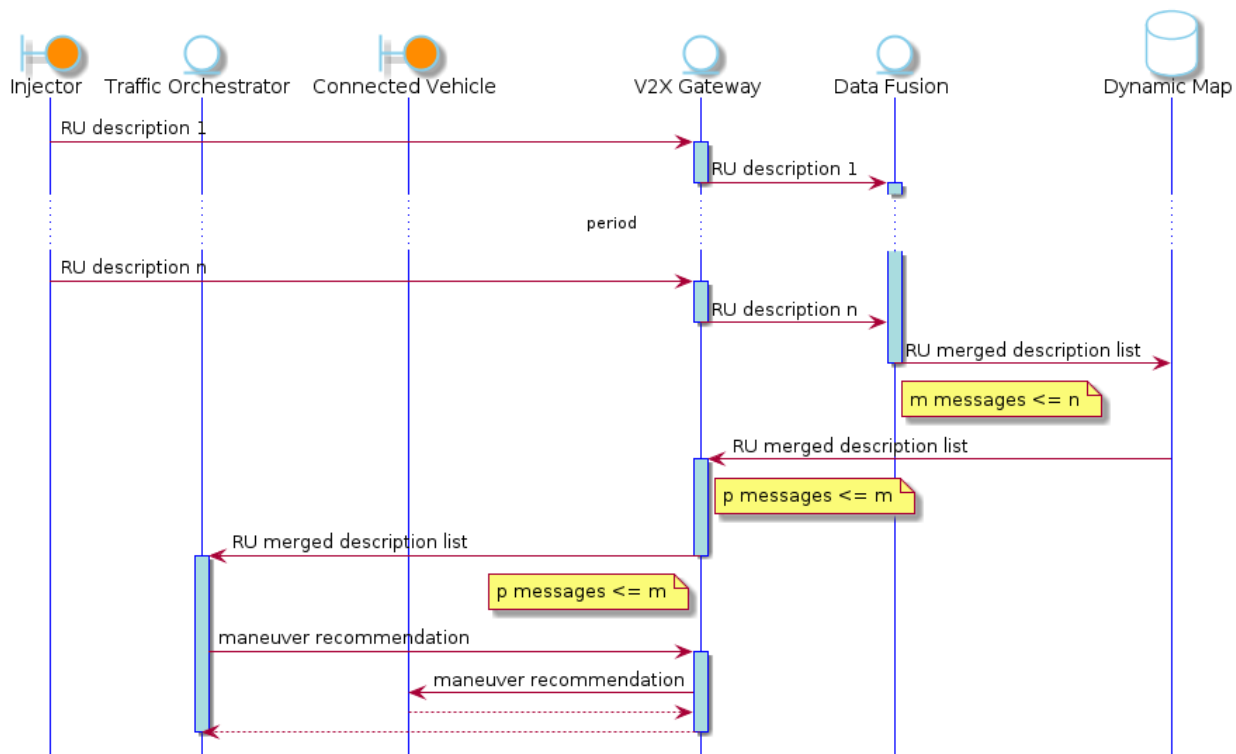


Figure A.11: The injector allows validating the result of a complete maneuver recommendation exchange.

B PSA Car Integration

For this project, we developed some specific messaging services in the car to interact with the other modules of the project. A CAN Gateway (called VBOX) was developed to extract the messages in the CAN bus and to send messages to the main HMI via a dedicated Ethernet connection, as illustrated in Figure B.1.

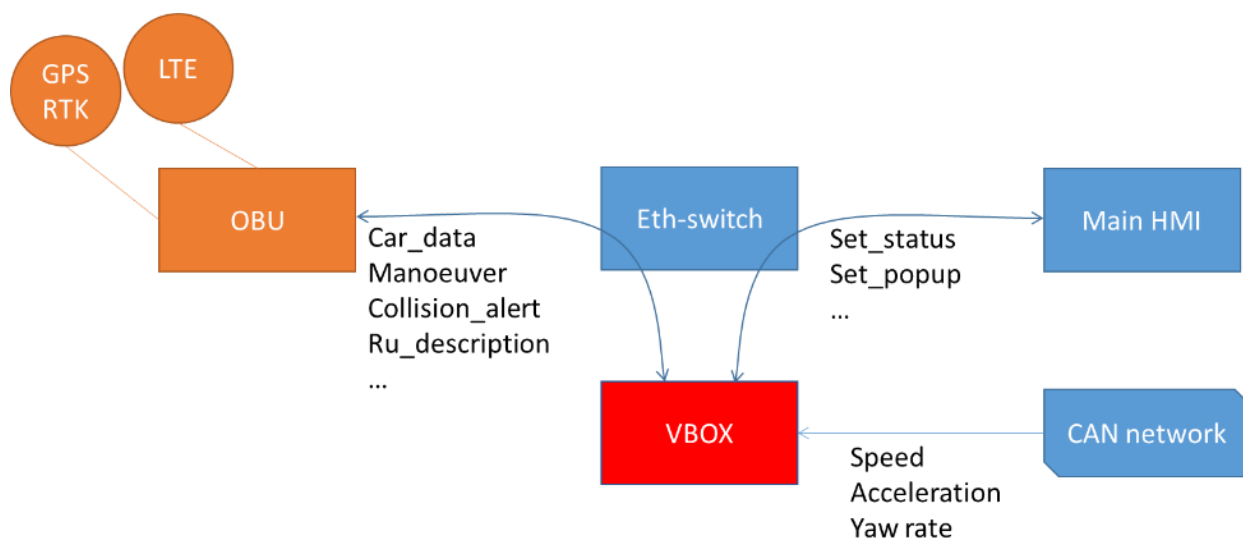


Figure B.1: In-vehicle architecture of PSA cars.

B.1 VBOX Definition

The VBOX in the previous figure is our CAN gateway for this project, and much more. Here is the hardware definition. It is used to read data on the CAN and send car information to the OBU, to act as an interface with the LiDAR system and build the detected vehicle message, and to interface the serial embedded HMI to display driving instructions or alert messages.

- Mechanical Constraints
 - Length: 153 mm, Width: 90 mm, Height: 50 mm
 - Weight: around 300 g
 - Connectors box face implementation: Both sides of the unit (802.11p and GPS on one side and rest on the opposite)
 - Material: Extruded Aluminum
 - Fixation: One face must be plane to be stuck
- External Access
 - Main Connector: 12 pin, Tyco: 1379662-1
 - APC, CAN, GND, Serial
 - Connector GNSS: Fakra 2 code L Carmine
 - Connector USB type B: USCAR
 - WiFi 802.11n: Internal module, external antenna with Fakra connector
 - Secondary USB. USB 2.0 OTG. Non-automotive connector (micro AB)



- Ethernet. Non-automotive connector (RJ45)
- Flash memory size: 4 GBytes
- Electrical constraints:
 - Operational range
 - 0 to 5.5V: No operation
 - 5.5V to 8V: May not operate. Except Stop&Start profile. Context must be stored, and RF power minimized
 - 7V to 24V: Nominal Mode
 - 6V to 7V, 24V to 36V V: Degraded Mode
 - Power-on consumption (peak current):
 - Expected: 15 A (less than 500 ms)
 - Maximum Active and Diagnosis state consumption: 3A
- Environmental constraints:
 - Ambiance temperature Range: -40° to 85° C (without enclosure)
 - Hardware Tests profiles:
 - SAE J1113; B217100 - B217110 - B217130

B.2 CAN Bus Interface

The data listed in Table B.1 is extracted from the CAN bus.

Table B.1 Sensor data extracted from the CAN bus.

Data	Frequency
Speed	25 Hz
Acceleration	25 Hz
Yaw rate	100 Hz

A message is built every 100 ms with the up to date CAN data and sent to the OBU. Size and color are static values, sent only once at the first connection.

B.3 HMI Interface

A modified HMI was developed in this project, with our serial supplier. Three serial products were modified to allow the display of specific messages for our demonstrations. The messages are shown in the main screen of the car, 8" screen (for 3008 and 5008) or HD 12" screen (DS7).

The main part was to introduce an Ethernet connectivity with the VBOX, and exchange JSON messages over standard TCP sockets.

Some methods were implemented on each part, which are listed in Table B.2.



Table B.2: Methods to interact with the VBOX.

Method Name	Description
setPopupNotification()	Set a notification on the HMI
getCurrentNotification()	Get the current notification displayed on the HMI
resetNotification()	Reset a specific notification
internalSystemNotification()	To activate or disable the Internal System Notification
itsServiceActivation()	Notify ITS Services activation/deactivation is not present

Each popup notification contains a picture, two lines of text and a sound alert. All of these can be configured with the method call.

Each message is geo-located, so we can display the image on the navigation map with the button “Show on map”, as illustrated in Figure B.2.

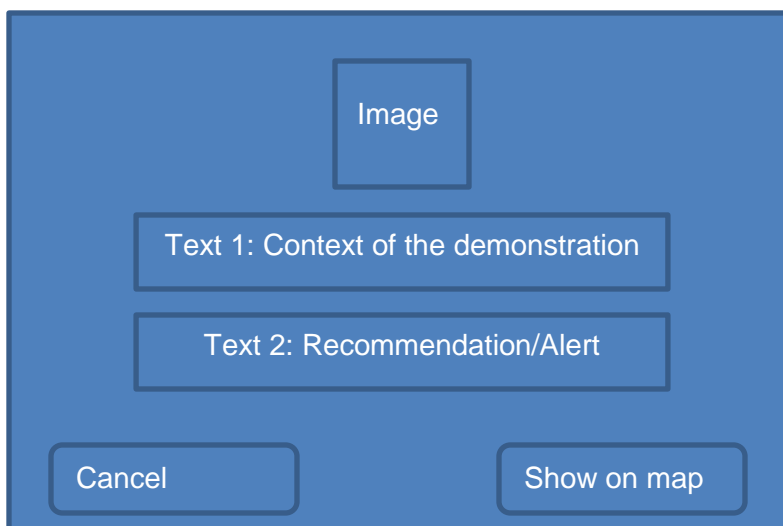






Figure B.2: Layout of popup messages in the HMI.

The image is an “easy to understand” representation of the message for the driver. Some examples are given in Table B.3. Text1 describes the current context of the demonstration (“Lane merge”, “Sensor Sharing”, or “VRU Protection”). Text2 explains the action, e.g. “Go to 70 km/h”, “Merge to the left”, “WARNING: incoming vehicle”.

Table B.3: Example of notification images.

Image	Meaning
	Recommended speed: 70 km/h
	Merge to the left
	Warning: incoming vehicle from the left
	Warning: incoming pedestrian from the right



C PSA Car Definitions

C.1 DS 7 Crossback



Vehicle Identification Number: VR1JJEHZRHY095555

VEHICLE FAMILY	DS 7 CROSSBACK
SILHOUETTE	S.U.V.
TRIM LEVEL	GT SPIRIT
ENGINE	DIESEL 2.0 HDi 180 FAP
EMISSIONS STANDARD	EURO 6.2 ENGINE EMISSION CONTROL
PARTICLE FILTER	WITH PARTICLE EMISSION FILTER
STOP AND START	WITH STOP AND START ALTERNATOR STARTER
TRANSMISSION	STT TYPE 8-SPEED AUTOMATIC GEARBOX
PAINT TYPE	METALLIC LACQUER
BODY COLOUR	ARTENSE GREY PAINT
ROOF GLASS	WITH SLIDING TILTING ROOF WINDOW ONE-TOUCH CONTROL ANTI-PINCH
ACCESS AND STARTING	WITH HANDS-FREE ACCESS/STARTING
LOUDSPEAKER	4 HP FRONT (2 TW + FR) + 4 HP REAR
SATELLITE NAVIGATION SYSTEM	WITH NAVIGATION SYSTEM COLOUR SCREEN
CRUISE CONTROL	SPEED LIMITER ACC STOP AND GO REGULATOR
DISPLAY	DIGITAL TOUCHSCREEN+12" DFT BLOCK



C.2 Peugeot 5008



Vehicle Identification Number: VF3MCBHZWJL002820

VEHICLE FAMILY	5008
SILHOUETTE	MULTIPURPOSE CROSSOVER
TRIM LEVEL	ALLURE
ENGINE	1.6 HDi 120 FAP 88KW TURBO DIESEL
EMISSIONS STANDARD	EURO 6.1 ENGINE EMISSION CONTROL
PARTICLE FILTER	WITH PARTICLE EMISSION FILTER
STOP AND START	WITH STOP AND START ALTERNATOR STARTER
TRANSMISSION	AUTO GEARBOX 6-SPEED STT
PAINT TYPE	METALLIC LACQUER
BODY COLOUR	EMERALD CRYSTAL PAINT
ROOF GLASS	WITHOUT ROOF WINDOW
ACCESS AND STARTING	FRONT ACCESS + BOOT + HANDS FREE STARTING
LOUDSPEAKER	4 FR SPEAKERS (2 TW + 2 FR) + 2 RR SPEAKERS
SATELLITE NAVIGATION SYSTEM	WITH NAVIGATION SYSTEM COLOUR SCREEN
CRUISE CONTROL DISPLAY	ADAPTIVE CRUISE CONTROL AND SPEED LIMITER DIGITAL TOUCH SCREEN

C.3 Peugeot 3008



Vehicle Identification Number: VF3MJHZRKS164468

VEHICLE FAMILY	3008
SILHOUETTE	5-DOOR CROSSOVER
TRIM LEVEL	GT
ENGINE	2.0 HDi 180 FAP 132KW
EMISSIONS STANDARD	EURO 6.2 ENGINE EMISSION CONTROL
PARTICLE FILTER	WITH PARTICLE EMISSION FILTER
STOP AND START	WITH STOP AND START ALTERNATOR STARTER
TRANSMISSION	STT TYPE 8-SPEED AUTOMATIC GEARBOX
PAINT TYPE	METALLIC PAINT + COLOURED VARNISH
BODY COLOUR	ULTIMATE RED PAINT
ROOF GLASS	WITHOUT ROOF WINDOW
ACCESS AND STARTING	FRONT ACCESS + BOOT + HANDS FREE STARTING
LOUDSPEAKER	5 FRONT SPEAKERS (2 TWEETERS + 2 FRONT + 1 CENTRAL) + 4 REAR SPEAKERS
SATELLITE NAVIGATION SYSTEM	WITH NAVIGATION SYSTEM COLOUR SCREEN
CRUISE CONTROL	ADAPTIVE CRUISE CONTROL AND SPEED LIMITER
DISPLAY	DIGITAL TOUCH SCREEN



D Volvo Car Integration

A Volvo XC60 T5 AWD model year 2019 was used for the VRU use case and a Volvo V60 D4 model year 2019 was used for the Lane Merge use case. The vehicles have an electrical system with up to more than 100 electronic control units (ECU) depending on power train and accessory level. Communication between the ECUs is realized by several different automotive in-vehicle communication links like Flexray, CAN, LIN, Ethernet and MOST. Sensor signals from the vehicle sensors like radar, camera, ultra-sonics, wheel speed sensors, inertial sensors, GNSS and other sensors monitoring the vehicles surrounding, movement and position are sent over the in-vehicle communication links to be used by vehicle functions in the different ECUs.

In this project a CAN gateway was installed as an interface to the in-vehicle networks to extract three different vehicle sensor signals, speed, acceleration and yaw-rate, see Figure D.1 and Figure D.2. An on-board unit (OBU) receives the information and generates a message also including position data from a separate GNSS RTK module connected to the OBU. The signals are sent to the V2X gateway via the cellular network. For positioning using 5G the vehicle was equipped with a 5GTX unit as described in Section 3.4.

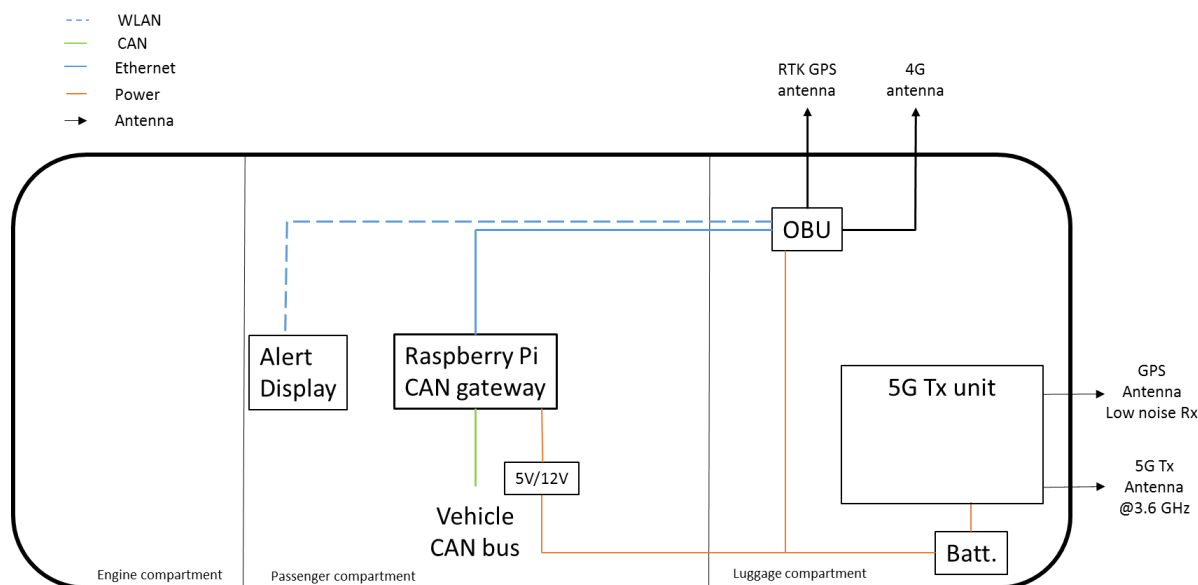


Figure D.1: Car integration in the Volvo XC60 T5 AWD used for the VRU use case.

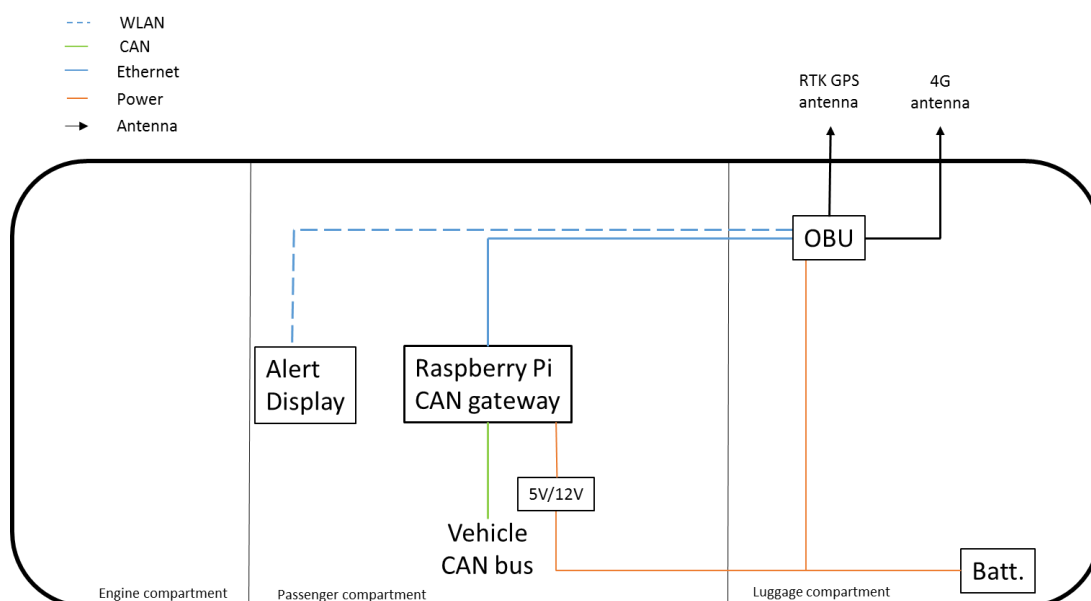


Figure D.2: Car integration in the Volvo V60 D4 used for the lane merge use case.

For detailed specification of the cars see <https://www.media.volvocars.com/global/en-gb>.

D.1 CAN Signal Definition

The data listed in Table D.1 is extracted from the CAN bus.

Table D.1: Dynamic data extracted from CAN bus.

Data	Frequency
Speed	100 Hz
Acceleration	100 Hz
Yaw rate	100 Hz

Size and color of the vehicle are static values, sent only once at the first connection.

D.2 Volvo CAN Gateway

The CAN gateway used in the Volvo Cars in this project is a Volvo developed solution based on a standard Raspberry Pi B model 3+ equipped with a PiCAN2 Duo CAN Bus Board for Raspberry Pi 2/3 with SMPS from SK Pang Electronics.

The software, also developed by Volvo Cars, are based on erlang/elixir and the code is available on GitHub at [VOLVO-BROKER].

Two applications, signalbroker and car5g (i.e., signalbroker client), are deployed in the RaspberryPi, see Figure D.3. The signalbroker acts as a server with the access to the CAN buses. After startup, the signalbroker would read a local JSON interface definition file to learn where to

fetch the data when it receives a signal subscription from a client. The car5g act as a client, it would connect to signalbroker and subscribe the signals it would like to receive. It would also automatically connect to the orange OBU after start-up and forward the signals received to the OBU in JSON format.

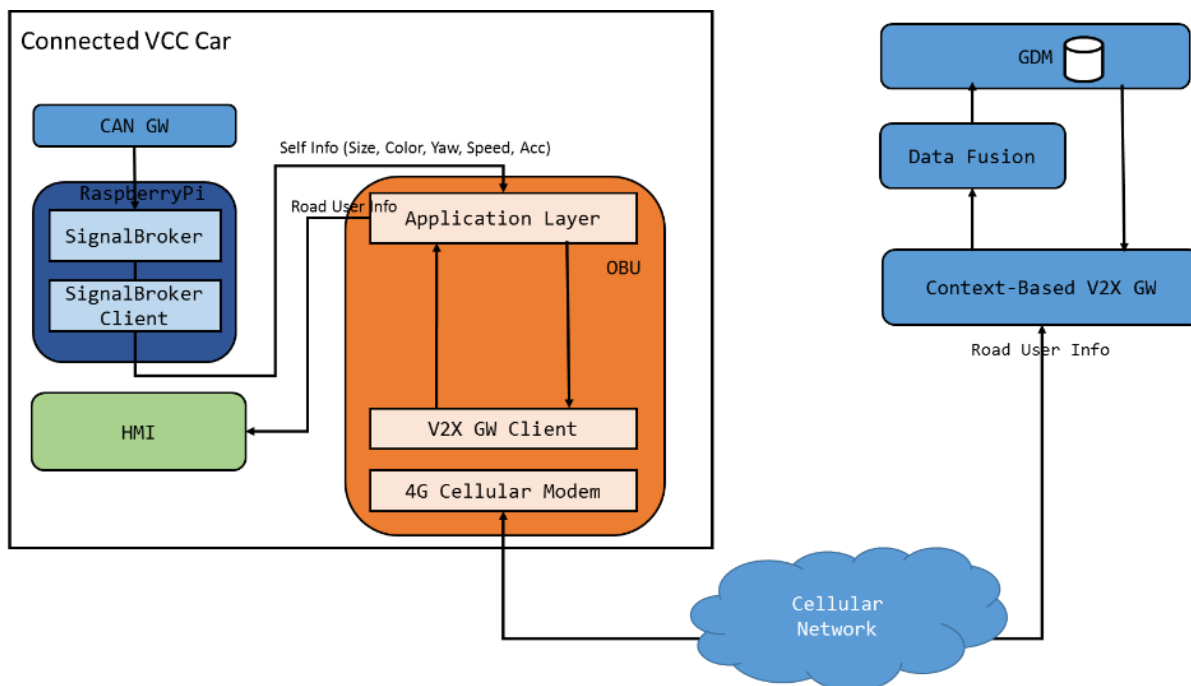


Figure D.3: Architecture view of the Volvo solution.