5GMOBIX

5G for cooperative & connected automated MOBIlity on X-border corridors

Deliverable D3.4

Report on corridor infrastructure development and integration

Dissemination level	Public (PU)
Work package	WP3: Development, integration and roll-outWP3: Development, integration and roll-out
Deliverable number	D3.4
Version	V2.0
Submission date	31/01/2021
Re-submission date	13/04/2022
Due date	31/01/2021

www.5g-mobix.com

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No [825496]

Authors

Authors and contributions by institution		
Organisation	Name	Contributed to sections
AALTO	Xuebing Liu Oussama el Marai Aziza Zhanabatyrova Edward Mutafungwa	6.2.1, 10.2
A-to-Be	Tiago Dias	6.1.1, 7.1.4, 7.1.6
AKKA	Fatma Raissi	10.3
CCG	João Ferreira Juel Puga	5.1.5, 6.1.2, 7.1.3
CTAG	Daniel Jáuregui Cortizo Margarita Saez Francisco Parada Andrea Becerra	4.2.1, 4.2.2, 5.1, 5.1.1, 5.1.3, 5.1.6, 6.1.1, 6.1.2, 7.1, 7.1.1, 7.1.2, 7.1.4, 7.1.6, 8.1, 10.7.2
DUT	Lujun Wang, Yanjun Shi	10.5
FORD OTOSAN	Tahir Sari	4.3.1, 5.2, 6.2.1, 6.2.2
GT-ARC	Doruk Sahinel	3.3
IMEC	S. Dima Hadiwardoyo Daniel van den Akker Vasilis Maglogiannis Dries Naudts	4.3.2, 7.2.2
Instituto de Telecomunicações	João Almeida	5.1.2, 5.1.6, 6.1.2, 7.1.4
Korea Automotive Technology Institute	You-Jun Choi	10.6
NOKIA ES	Ignacio Benito Frontelo Jaime Jesús Ruiz Alonso Raúl López López	5.1.7, 6.1, 7.1.5, 10.8
SENSIBLE4	Timo Mustonen	10.2.2
Siemens	Igor Passchier	10.4, 10.7.3
Siemens (PT)	Francisco De Freitas	5.1.4
TUB	Sebastian Peters	T3.4 lead and main document editor for D3.4, contributions to 1, 2, 3, 4.1, 9, 10.1, 10.7.1
	Federico Murciano	10.1, 10.7.1
TUBITAK BILGEM	Arda Taha Candan	5.2, 6.2.2

VEDECOM	Ahmed Soua Laurent Fevrier	10.3
WINGS ICT SOLUTIONS	Kostas Trichias Polina Dimiropoulou Ioannis Dimitriadis Christina Daskalopoulou Vasilis Foteinos	5.2, 6.2.1, 6.2.2, 7.2.1. 8.2

Control sheet

Version history			
Version	**Date**	**Modified by**	**Summary of changes**
V0.1	22/03/2019	M. Khan (TUB)	Table of contents
V0.8		S. Peters (TUB)	Restructured version with new ToC
V0.9	06/11/2020	S. Peters (TUB), all contributors	Initial version with merged TS/CBC contributions
V0.99	16/11/2020	S. Peters (TUB), all contributors	First integrated version
V0.999	17/11/2020	S. Peters (TUB), all contributors	Version submitted for peer review
V1.0_r	25/01/2021	S. Peters (TUB), all contributors	Peer reviewed version ready for quality check review
V1.0_q	27/01/2021	M.-L. Watrinet (LIST)	Quality review
V1.0	30/01/2021	Sebastian Peters (TUB), contributors	Final revisions after quality review. Final version submitted to EC portal.
V1.1_p	15/03/2022	Sebastian Peters (TUB), all contributors	Revised version submitted for peer review
V1.1_q	22/03/2022	M.-L. Watrinet (LIST)	Quality review
V1.1	13/04/2022	Sebastian Peters (TUB), all contributors	Final revisions after quality review for final check.
V2.0	13/04/2022	Sebastian Peters (TUB), all contributors	Final version submitted to EC portal.

Peer review		
	Reviewer name	**Date**
Reviewer 1	Joaquim Ferreira (IT)	27/11/2020 22/03/2022 v1.1_p
Reviewer 2	Johan Scholliers (VTT)	22/11/2020 v1.0_r 19/03/2022 v1.1_p

Legal disclaimer

The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any specific purpose. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. The 5G-MOBIX Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. Copyright © 5G-MOBIX Consortium, 2018.

Table of contents

List of figures

List of tables

ABBREVIATIONS

Abbreviation	Definition
5GC	5G Core Network
AMQP	Advanced Message Queuing Protocol
ASN.1	Abstract Syntax Notation One
AVM	Around View Monitoring
CAM	Cooperative Awareness Messages
CBC	Cross Border Corridor
CCAM	Cooperative, Connected and Automated Mobility
CPM	Collective Perception Messages
CoCA	Cooperative Collision Avoidance
C-V2X	Cellular V2X
DENM	Decentralized Environmental Notification Message
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
E2E	End-to-end
ECU	Engine Control Unit
EDM	Edge Dynamic Map
eRSU	Enhanced Road-Side Unit
GNSS	Global Navigation Satellite System
GPCU	General Purpose Compute Unit
GUI	Graphical User Interface
ITS	Intelligent Transportation System
LBO	Local Break-Out
LiDAR	Light Detection and Ranging
MC	Maneuver Coordination

MCS	Manoeuvre Coordination Service
MCM	Manoeuvre Coordination Message
MEC	Mobile Edge Computing
MQTT	Message Queuing Telemetry Transport
NAT	Network Address Translation
PDU	Power Distribution Unit
PTZ	Pan–tilt–zoom
QoS	Quality of Service
RHO	Remote Human Operator
ROI	Region of Interest
ROC	Remote Operation Center
ROS	Robot Operating System
RPM	Rotations Per Minute
RSU	Road-Side Unit
RTSP	Real Time Streaming Protocol
SOC	Single Onboard Computer
TAC	Tracking Area Code
ToD	Tele-operated Driving
TOF	Time-Of-Flight
TS	Trial Site
UMRR	Universal Medium Range Radar
UPER	Unaligned Packed Encoding Rules
UUID	Unique User ID
V2X	Vehicle-to-everything
VM	Virtual Machine
VPN	Virtual Private Network

V-PLMN	Visiting Public Land Mobile Network
VR HMD	Virtual Reality Head-Mounted Display
VRU	Vulnerable Road Users
WP	Work Package

EXECUTIVE SUMMARY

This document is deliverable D3.4 "Report on corridor infrastructure development and integration". It is dedicated to report on the CCAM infrastructure development, integration and deployment, which has been carried out within the scope of the Task 3.4 of the 5G-MOBIX project.

D3.4 reports on the development, integration and deployment of roadside, traffic and corridor management infrastructure, and the evolution of the platforms and applications required for the use cases. In 5G-MOBIX, these use cases are performed in two CBCs (Cross-border Corridors) (Spain-Portugal and Greece-Turkey) and six local TSs (Trial Sites) amongst which four are located in Europe (France, Finland, Netherlands and Germany).

The CCAM infrastructure deployment activities at all CBCs and TSs, in alignment with the CCAM infrastructure specifications described in (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), aimed at delivering complete road facilities in December 2020. The activities to deploy or enhance the existing infrastructure in the CBCs and TSs according to this goal have been concluded, which also corresponds to the timeline presented in Section 2. Depending on the individual scope of the CBCs and TSs, the deployed components of CCAM infrastructure include Roadside Units (RSUs) and sensors, MEC or edge compute nodes, and ITS center or cloud infrastructure, a high-level summary of which is provided in Table 1. In addition to the hardware deployment, the specific CCAM applications that have been under development for the 5G-MOBIX use cases are completed and can be used in the trials. Within the scope of the T3.4 activities a simple functional testing of the deployed hardware and developed software has been performed. Further testing and integration of applications with the deployed infrastructure has been performed in the scope of integration and verification activities in T3.6.

This deliverable provides an overview on the planning and global alignment of the activities that were conducted by the individual sites in section 2, summarizes the CCAM infrastructure components and cross-border-corridor cooperation by each trial site in section 3. The remaining sections provide a detailed description of the roadside infrastructure components, the deployment and enhancement activities that have been conducted, the evolved applications and platforms, as well as the integration and testing activities of the CBCs (section 4 to 8). The same rich contents for the trial sites from DE, FI, FR, NL, CN and KR is provided in the Annexes to keep the focus of the main document on the CBCs and the overview of components.

1. INTRODUCTION

1.1. Corridor infrastructure development and integration in 5G-MOBIX

Being complex and distributed communication systems, the implementation of Cooperative, Connected and Automated Mobility (CCAM) infrastructure projects usually involves both the development and the installation of software, as well as the deployment of hardware to provide CCAM services. In the 5G-MOBIX project, this includes the development and provisioning of new software packages, procurement of networked hardware equipment, servers and/or specialized sensor hardware for the fulfilment of the specific use case goals of the two CBCs (Cross-border Corridors) (Spain-Portugal and Greece-Turkey) and the six local TSs (Trial Sites), amongst which four are located in Europe (France, Finland, Netherlands and Germany) and two in Asia (China, Korea). This deliverable serves the purpose of describing the development and integration activities of the roadside, traffic and automated driving management infrastructures on these CBCs and TSs conducted within the scope of T3.4.

While the infrastructure enhancement with 5G capabilities is mostly a T3.3 activity, the activities described in this deliverable focus fully on the CCAM infrastructure enhancement to attain the required corridor functionality, of which 5G modem hardware integration into Road-Side Units (RSUs) is only a smaller part. On the other hand, it should be noted that the purpose of the CCAM infrastructure described in this deliverable is to showcase and trial the 5G features for CCAM use cases.

As a part of WP3, the activities in task T3.4 follow the 5-phase rollout plan that has been defined in Deliverable 3.1 (5G-MOBIX, Deliverable D3.1: Corridor and Trial Sites Rollout Plan, 2020). This Deliverable is prepared during the final phase of the project and therefore reports on the completed CCAM deployment and enhancement activities, including test results pertaining to the correct functioning of the deployed infrastructure. Starting from the simple functional testing methodology, further verification and integration testing was performed within the scope of T3.6. This deliverable further provides an overview of CBCs' and TSs' CCAM architecture employed in 5G-MOBIX, the particular infrastructure including their features and implementation, as shown in Table 1.

1.2. Purpose of the deliverable

This document is deliverable D3.4 "Report on corridor infrastructure development and integration" dedicated to report on the CCAM infrastructure development, integration and deployment. It has been carried out within the scope of the Task 3.4 of the 5G-MOBIX project. In alignment with the CCAM infrastructure specifications in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), the CCAM infrastructure deployment activities at all trial sites have been concluded and the created assets are utilized in the trials.

1.3. Intended audience

Being open to the public, the Deliverable 3.4 provides a comprehensive overview to interested readers on the CCAM infrastructure that has been rolled out, on applications and platforms that have been enhanced, or new sensing components that have been added to attain the use cases of the different trial sites and cross-border corridors of the 5G MOBIX project. As such, this Deliverable also provides an overview on the outcome of the roadside infrastructure development activities conducted within the project, which then serve as the basis for the trialling activity that will be conducted within the scope of Work Package (WP) 4, as well as the evaluation activities that will be defined by the deployed technologies in WP5.

1.4. Structure of the document

It should be noted here that in the following structure of the document the main sections from Section 4 to Section 8 are describing the CBCs only, with the description of the individual trial sites being available in the annexes for improved readability and in order to keep the focus on the CBCs.

This deliverable is structured into the following sections:

- Section 2: Presents the implementation and deployment plan for the CCAM infrastructure and the execution steps of the 5-phase rollout plan.
- Section 3: Provides an overview on CCAM infrastructure deployments, the overall CCAM components employed in the TSs and CBCs and integration contributions of each local trial site to cross-border corridors.
- Section 4: Provides an overview of the CCAM infrastructure components of ES/PT and GR/TR.
- Section 5: Describes the infrastructure enhancement and deployment activities conducted by ES/PT and GR/TR.
- Section 6: Details ES/PT and GR/TR CBC's applications developed for use cases and enhancement of individual platforms.
- Section 7: Describes the integration and testing activities conducted within the scope of the previous sections' activities.
- Section 8: Describes any future work that is planned in ES/PT or in GR/TR.
- Section 9: Concludes the main part of the document.
- Annexes: Provides the details of the six trial sites, following the same structure as provided above for the CBCs, as well as preliminary test results of function-testing the infrastructure. In addition, a section for technical installation guides too detailed for the main part of the document is provided.

2. IMPLEMENTATION AND DEPLOYMENT PLAN FOR CCAM INFRASTRUCTURE

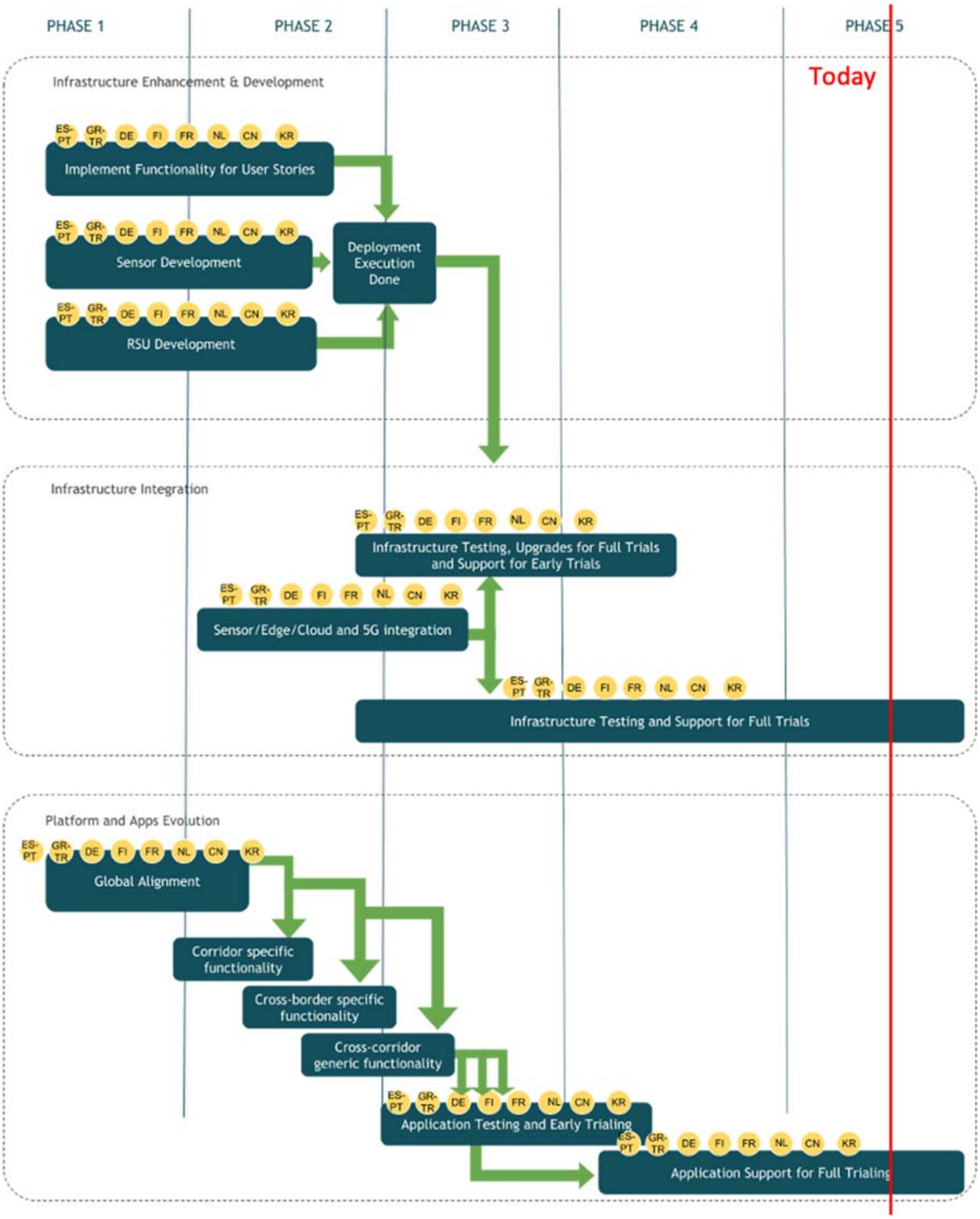
This section provides an overview of the timeline and the steps followed for the CCAM infrastructure deployment in 5G-MOBIX. The planning and alignment have been applied to all cross-border corridors and trial sites to structure the activities within task T3.4. The corridor infrastructure development and integration activities follow a 5-Phase rollout strategy that has been devised within the scope of T3.1.

2.1. Global Alignment

The work conducted within T3.4 to complete the road deployments is grouped in three main activities:

- Infrastructure enhancement and development
- Infrastructure integration
- Platform and Apps evolution

These activities are carried out in 5 phases which partially overlap, as depicted in Figure 1. The first phase of activities focuses on the execution plans and global alignment efforts, and the beginning of the development and deployment activities. The focal point of the second phase is the sensor, edge, and cloud infrastructure deployment and their integration with 5G networks. This is followed by infrastructure testing and verification activities in phase 3. The activities of the fourth phase aim at supporting early trialling with the CCAM infrastructure, while facilitating further upgrades to the deployments. Having completed these activities, phase 5 then focuses on the maintenance of the CCAM infrastructure to support the full trialling. Figure 1 provides an overview of these five phases implementing the rollout plan for the CCAM infrastructure.


PHASE 1
PHASE 2
PHASE 3
PHASE 4
PHASE 5
Today
Infrastructure Enhancement & Development
ES-PT
GR-TR
DE
FI
FR
NL
CN
KR
Implement Functionality for User Stories
Sensor Development
RSU Development
Deployment Execution Done
Infrastructure Integration
Infrastructure Testing, Upgrades for Full Trials and Support for Early Trials
Sensor/Edge/Cloud and 5G integration
Infrastructure Testing and Support for Full Trials
Platform and Apps Evolution
Global Alignment
Corridor specific functionality
Cross-border specific functionality
Cross-corridor generic functionality
Application Testing and Early Trialing
Application Support for Full Trialing

ES-PT = Spain-Portugal, **GR-TR** = Greece - Turkey, **DE** = Germany
FI = Finland, **FR** = France, **NL** =Netherlands, **CN** =China, **KR** =Korea

Figure 1: Overall Picture of the 5-Phase Rollout Plan for CCAM Infrastructure (March 2022)

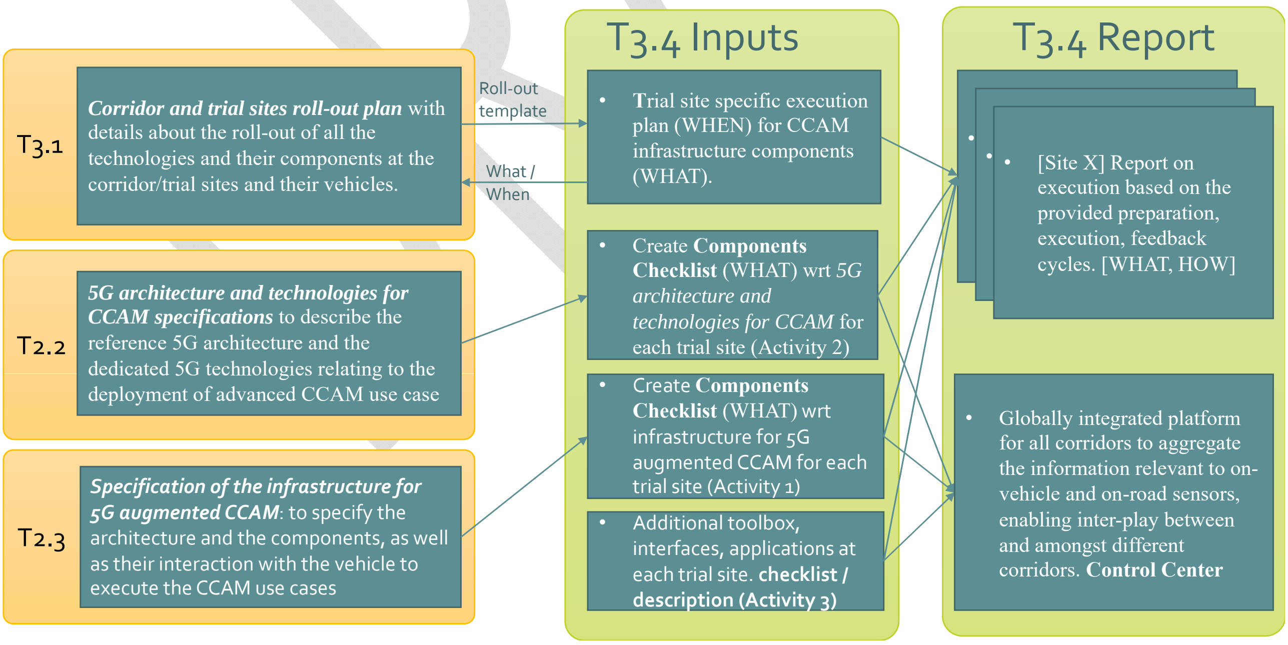
2.2. Development and Deployment Schedule

Following the 5-phase plan of the 5G-MOBIX project, the development and deployment activities of T3.4 have been concluded in December 2020, marking the end of phase 2. To monitor the work progress, all the task activities have been defined in the ClickUp tool under an *Infrastructure & Apps* category for each TS and CBC. Each activity provided by the different sites details the planned start & end dates and task assignees are also added to each activity. In addition, the owner of the task may set a slider to indicate the progress of the individual task. Given this approach the collaboration was managed for all phases of the project.

2.3. T3.4 position in 5G-MOBIX

The rollout of the CCAM infrastructure for 5G-MOBIX is closely related to other tasks in the project. These interrelations are shown in Figure 2 below. The activities in T3.4 are preceded by the work performed in WP2, most notably by T2.2 specifying the 5G architecture and technologies for CCAM, and by T2.3 specifying the infrastructure for 5G augmented CCAM. The specifications of T2.2 and T2.3 were reported on in the publicly accessible deliverables D2.2 (5G-MOBIX, Deliverable D2.2: 5G architecture and technologies for CCAM specifications, 2019) and D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019) respectively.

Last but not least, T3.1 defined the 5 Phase plan and provided rollout planning, taking into consideration the status of T3.4.



Corridor and trial sites roll-out plan with details about the roll-out of all the technologies and their components at the corridor/trial sites and their vehicles.

5G architecture and technologies for CCAM specifications to describe the reference 5G architecture and the dedicated 5G technologies relating to the deployment of advanced CCAM use case

Specification of the infrastructure for 5G augmented CCAM: to specify the architecture and the components, as well as their interaction with the vehicle to execute the CCAM use cases

- Trial site specific execution plan (WHEN) for CCAM infrastructure components (WHAT).
- Create **Components Checklist** (WHAT) wrt *5G architecture and technologies for CCAM* for each trial site (Activity 2)
- Create **Components Checklist** (WHAT) wrt infrastructure for 5G augmented CCAM for each trial site (Activity 1)
- Additional toolbox, interfaces, applications at each trial site. **checklist / description (Activity 3)**

- [Site X] Report on execution based on the provided preparation, execution, feedback cycles. [WHAT, HOW]
- Globally integrated platform for all corridors to aggregate the information relevant to on-vehicle and on-road sensors, enabling inter-play between and amongst different corridors. **Control Center**

Figure 2: Overview on inputs to T3.4 activities

3. OVERVIEW ON INFRASTRUCTURE DEPLOYMENTS, CCAM COMPONENTS AND COOPERATION

3.1. Overview on CCAM Infrastructure Deployments

This section provides an overview on the achieved CCAM infrastructure deployments throughout all CBCs and TSs. In contrast to pure software development projects, the approach for realizing CCAM infrastructure deployments differs significantly in terms of best practices, as implementation projects such as 5G-MOBIX face completely different challenges. One challenge in this connection is the deployment of roadside infrastructure elements, which usually requires an authorization from local government bodies, detailed planning of required civil works (e.g., for power supply) and the preparation of the installation date. Furthermore, the ability of proper maintenance after the deployment needs to be factored in, e.g., to restore access to a faulty network device. A typical CCAM implementation therefore requires resources from a variety of specialized areas, including programming and interfacing, but also networking, hardware deployment, and technical support. Another deployment challenge to the 5G-MOBIX consortium has been the delayed availability of 5G chipsets, to be included e.g., into roadside deployments in order to provide 5G connectivity.

As can be seen from Table 1, the main components of CCAM infrastructure deployments can be distinguished by Roadside Units (RSUs), MEC or edge compute nodes, (distributed) applications developed for the use cases, and ITS center or cloud infrastructure. At the time of writing this Deliverable in March 2022 the activities to deploy or enhance the existing infrastructure in the CBCs and TSs have been concluded. The table shows the number of items that have been deployed per CBC and TS, and gives an insight on the high-level status of the activity. For example, a device is first assembled and tested in a lab environment, afterwards the deployment is performed on the devised location. After successful simple function tests the device is operational. From here onwards the device is maintained to ensure that its functionality is provided as intended to the user. Similar to the hardware deployment, the applications that have been under development for the 5G-MOBIX use cases are completed. Following the development of the application comes the functional testing of the software and subsequently the deployment on MEC or cloud server infrastructure. Following the important deployment milestone, the testing and integration of applications with the deployed infrastructure has been performed as the next step. This testing and integration activity is part of Phase 3, and enables the utilization within early trials in Phase 4. Necessary upgrades / modifications of the CCAM infrastructure components that were identified during the early trials phase then allowed for the full trialling in Phase 5, which is ongoing in the CBCs and TSs at the time of writing this deliverable.

Table 1: High-level overview on CCAM infrastructure deployments

CBC/TS	MEC/Edge Nodes	RSUs	Applications	ITS centers / Cloud	Status & remarks
ES-PT	2, operational	5, operational	4, dev. completed and verified	2, operational	All elements ready for trials.
GR-TR	1 edge node ready (GR) 1 node pending (TR)	3	3, dev. completed and verified	2 Cloud services operational	All elements ready for trials.
DE	9 far edge nodes, 2 near edges operational	9+2 with PC5, operational	2, dev. completed and verified	1, ITS center operational	All elements ready for trials.
FI	2 MECs operational	-	4, dev. completed and verified	1, operational	Applications integrated, tested and used in trials
FR	2, operational at Satory and TEQMO	-	6 Apps. completed and verified	1, cloud operational 1, ITS centre operational	MECs in FR TS are deployed KPI manager operational
NL	2, operational	-	3, dev. complete	3, operational	Applications integrated tested and used in trials.
CN	2, operational	3	2 Apps	1, operational	-MECs servers in CN TS are deployed -Applications integration in MEC is ongoing
KR	-	1, testing	1 App	1	-

3.1.1. 5G Configurations in CCAM Infrastructure Deployments

In order to equip infrastructure components like RSUs with 5G capabilities, 5G modems need to be integrated into the deployment. This section briefly describes the most important aspects to be considered when upgrading infrastructure with 5G modems and discusses the options available in terms of configuration of the devices to facilitate the RSU to fulfil its purpose. It goes without saying that the deployment location of an RSU equipped with a 5G modem ideally has suitable 5G coverage available, with sufficient signal quality for stable operation.

Having accomplished the successful site survey for an RSU, the 5G modem selected for the deployment needs to be tested to ensure that it works with the specific 5G NSA configurations of the mobile network operator. This has been an issue specifically in the first generation of 5G modems, where the operator-specific ENDC combinations of 5G NSA Mode 3 were only partially supported. More recent releases of 5G modems have resolved this issue mostly, allowing them to connect reliably to 5G base stations by utilizing one of the diverse LTE anchoring frequency combinations employed by the mobile network operators. Regarding the use of (experimental) 5G SA networks, a similar maturity level as in NSA networks had yet to be reached by the commercial end user devices at the time of writing this Deliverable in March 2022.

Having ensured that a 5G modem works for a specific deployment location and connects to the 5G base station, only few configuration options are available to the end user. In addition to fitting the device with the SIM card that allows the 5G modem to connect to the mobile network operator that provides the 5G coverage on site, the configuration of the device is mostly limited to setting the APN configuration. The APN configuration of the 5G modem captures all the settings that an operator applies to the connection, like IP configurations, NAT or Firewall configurations, settings that enable local break out to the nearest MEC, et cetera. Therefore, in case that the standard APN configuration does not suffice for the application running on the RSU, specialized APN configurations need to be negotiated with the mobile network operator and implemented by them for subsequent use in the RSUs.

The present deliverable D3.4 reports on CCAM infrastructure, the in-depth discussion of how mobile network operators implement the operator-specific, custom APN settings within their networks is therefore out of the scope of this document. Furthermore, this deliverable captures the CCAM development and deployment activities, more information on the trials, test cases, and 5G configurations used in these activities are reported within the scope of WP4.

3.2. Overview on CCAM Infrastructure Components

In order to facilitate the reader, this section provides an overview on the CCAM components by introducing the CCAM infrastructure that has been realised in the scope of 5G-MOBIX by the different TSs and CBCs. The detailed description of the individual CCAM use cases is provided in the Deliverable D2.1 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019), the roadside and cloud infrastructure to support the CCAM use cases on the other hand is described in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019).

In this deliverable we therefore focus on the realized CCAM components, which are presented in detail in the specific TS/CBC sections. In the following we briefly focus on the constituent CCAM communication and infrastructure elements that facilitate the realization of these use cases. This includes the messages (either based on a standardized ITS or a custom non-standard protocol) that connected elements transmit to (and receive from) a central entity.

This central entity is also referred to as the V2X application server (ETSI TS 122 185, V16.0.0, 2020-08) and provides services for one or multiple V2X applications in a specific area. A V2X application server responsible for a certain CCAM application receives the messages from the relevant set of elements (e.g., vehicles, roadside infrastructure, pedestrians) and arranges them in topics, to which elements that participate in this V2X application can subscribe to, to learn about events that are of relevance for its own purpose. This communication coordination performed by the V2X application server is known as message brokering, and if deployed in the mobile network operators Multi-access Edge Computing (MEC) infrastructure can be referred to as the MEC Broker. An important tool that supports the MEC Broker in serving messages of a particular topic only to geographically relevant areas or communication participants is the Geoserver. All the information that reaches the MEC Broker can be displayed in a central monitoring entity that constitutes the ITS center, basically a control center / dashboard that visualizes the current CCAM message events for the geographical area of interest.

Putting all these components together, i.e., the CCAM messages, the communication infrastructure based on 5G Uu or PC5 interfaces, the MEC Broker with Geoserver and the central ITS center, create the distributed application that realizes the specific purpose of the CCAM application at hand. In 5G-MOBIX a wide range of CCAM applications have been realised with a varying focus on any of the aforementioned components. The following tables provide an overview on the V2X architecture of the specific TSs and CBCs. Table 2 shows the CCAM architecture per CBC/TS in an aggregated form, with all the components that have been deployed per site. Table 3 shows the alignment of 5G-MOBIX use cases with the 5GAA V2X application layer reference architecture.

Table 2: Overview on CCAM architecture per CBC / TS

CBC/ TS	C-V2X Direct PC5 / Uu-based	CCAM Messages	MEC Broker Deployment	Geoserver	MEC Applications
ES/PT	Uu	CAM, DENM, CPM, + custom messages	Implementation based on Mosquitto	Embedded in MEC Broker	VRU, Geoserver + MQTT broker, Remote driving and Registry
GR/TR	PC5 & Uu	CAM, DENM + custom messages	MQTT	-	Assisted zero-touch border crossing
DE	PC5 & Uu	CAM, CPM, DENM, MAP, SPAT + custom messages	Kafka, Mosquitto MQTT	ETSI-based, part of device layer or as Geobroker (embedded in MEC Broker)	Signaling Server for Surroundview, EDM Service
FI	Uu	Custom messages	Coordinator based on gRPC	-	HD Maps
FR	Uu & PC5	CAM, DENM, CPM, MCM, MAP + custom messages	MQTT	Embedded in MEC Broker	Data fusion, Risk analysis, Trajectory guidance, Predictive QoS
NL	Uu	CAM, DENM, MAP, SPAT, RTCM, IVI, CPM, MCM, + custom messages	Implementation based on Rabbitmq	Embedded in MEC Broker	Remote driving, collision avoidance application
CN	PC5	BSM, MAP, RSI, RSM, SPAT. CAM	Mosquitto MQTT	- (addition planned)	Cloud-assisted driving
KR	Uu	Custom messages	-	-	-

It should be noted here that depending on the specific use case not all components mentioned above are required. Similarly, while some use case might utilize already standardized ITS messages, another use case introduces a custom ITS message utilizing a non-standardized message type. Details on the particular implementation of use cases are described by the CBCs and TSs in their respective sections of this deliverable. Further differences can be found in the support for the V2X communication over the PC5 and/or Uu interface, which is a focus in some sites but not in all.

The approaches chosen for the V2X application server can be differentiated along the lines of the utilized software (e.g., based on popular tools like MQTT), the deployment of several instances responsible for (possibly overlapping) geographical areas and the support for advanced geo-based services, based e.g., on a Geoserver or based on custom solutions, to implement the intended message relaying behaviour. Finally, another important aspect is that of re-publishing the information from the domain of another MEC broker, e.g., because of the relevance to the communication participants of the local MEC broker.

In particular, this last aspect creates a whole new set of challenges to the MEC Broker and the participants of the use case, when the environment should be open to all stakeholders and work across countries. Taking this perspective, the mobile network operators, the ITS and road authorities, as well as the automotive industry need to be able to work with the CCAM deployments of other countries in order to not have scattered domains of service within specific areas or that are limited to particular borders of EU countries.

One important aspect in the discussion above is the need for standardization that enables the CCAM platforms implemented in the distributed infrastructure to become open for the participation of all stakeholders, avoiding scattered CCAM services and service areas as a result.

In order to create a solution that solves the complexities of an environment that is open to all stakeholders, the 5GAA has created a fitting V2X application layer reference architecture (A-200094, 2020), which has been published in June 2020, after the 5G-MOBIX use cases were defined and implementation had started. For this reason, in the following we present the alignment of the use cases developed within the scope of 5G-MOBIX with the V2X application layer reference architecture, matching the already implemented solution approaches with 5GAA's functional interface view depicted in Figure 3.

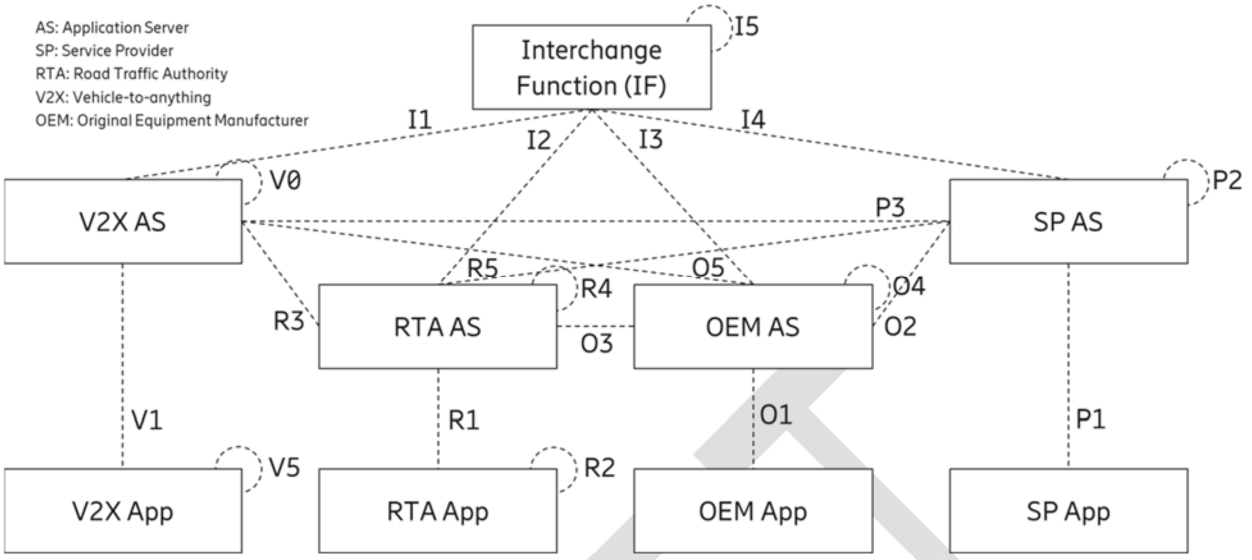


Figure 3: 5GAA V2X Application Layer Reference Architecture (A-200094, 2020)

In Table 3 we summarize the interfaces corresponding to the 5G-MOBIX use case implementations that realize communication between different entities of the V2X application layer reference architecture. The individual discussion of the use case implementations can be found in the CBC/TS-specific sections of this deliverable.

Table 3: Per Use Case Application Interfaces following 5GAA V2X Application Layer Reference Architecture

US No.	US Id	V0	V1	V5	R1	R2	R4	O1	O4	P1	P2	I1	I2	I3	I4	I5
1.1.a	LaneMerge		X		X											
1.1.b	Overtaking		X													
1.2	AssInfrastructure	X	X		X					X		X	X		X	
1.3	CoCA	X	X													
1.4	CloudAssisted		X			X										
1.5	CoopAutom		X		X											
2.1.a	SeeWhatISee		X	X						X	X					
2.1.b	5GPlat		X	X						X		X			X	
2.2	AsseRSU	X	X	X	X	X						X	X			
2.3	AssCloud		X			X										
3.1.a	HDMapsVehicle				X			X								
3.1.b	HDMapsPublicTransport				X			X								
3.2.a	AssBCrossing		X							X		X			X	
3.2.b	TruckRouting				X											
3.3	EDM	X	X	X												
3.4	EdgeProcessing	X	X					X								

3.5	CPM	X	X								X					
4.1	RCCrossing		X								X					
4.2	RedundantNE		X					X		X						
4.3	5GPositioning		X													
4.4	DataOwnership		X													
4.5	mmWave		X													
5.1	MediaPublicTransport									X						
5.2	Tethering									X						

In the following subsection we are providing a short overview on the MEC standardisation efforts with regard to V2X and elaborate on the characteristics of the MEC infrastructure being used in the different TSs/CBCs.

3.3. Overview on MEC infrastructure

The Multi-access Edge Computing (MEC) infrastructure, as standardized by the ETSI ISG MEC (ETSI GS MEC 003, 03-2022), and especially the inter-MEC communication of systems operated by different mobile network operators is deemed to be an important enabler of V2X interoperability in a multi-vendor, multi-network and multi-access environment. This amounts to the capability of the MEC architecture to support the inter-working and coordination of different MEC platforms e.g., operated by different MNOs. These aspects have been in the focus of a study on MEC federation (ETSI GR MEC 035, 06-2021), which explicitly takes the MEC federation for C-V2X into perspective. Furthermore, a specification of a V2XMEC service API has been released in (ETSI GS MEC 030, 04-2020), providing information to MEC applications in a standardized way. Given these standardisation efforts it is clear that future V2X application development will be able to fully exploit the capabilities of the MEC infrastructure while being agnostic to the complexities arising from the multi-vendor, multi-network and multi-access environment. The MEC applications developed within the scope of 5G-MOBIX predate these standardisation efforts and the interconnection and coordination of MEC systems / applications needed to be part of the application layer implementation and 5G-MOBIX project partners have pursued the alignment of their diverse CCAM use cases by aiming at the integration of their contribution into CBCs.

The MEC infrastructures of CBCs/TSs that have been used for the development and deployment of the MEC applications are summarized in Table 4.

Table 4: Overview on MEC infrastructures utilized in the different CBCs/TSs

CBC/TS	Number of near edge MECs & location	Type of MEC infrastructure employed (commercial/research deployment)	MEC nodes network interconnection (direct connection (e.g. IPX interconnect or dedicated fiber) / via Internet)	Status & remarks
ES/PT	2 (Vigo, ES and Riba d'Ave - Braga, PT)	Research deployment	Direct Interconnection (via dedicated fibre)	Both MECs are operational
GR/TR	2	Research deployment	Edge nodes will be connected in 2 ways, a) public internet interconnection and b) direct connection. Full trials will take place with both configurations.	GR Edge fully ready (All Applications installed and tested) TR Edge deployed. Pending SW and application installations.
DE	1 MEC at mobiledgeX (Berlin node), 1 MEC at TUB	Commercial (mobiledgeX) and research deployment (TUB)	Via Internet	MECs operative.
FI	2, at the same LAN of UPF	Research deployment	Direct connection (IP connection). Because the Core Network (UPFs) of the two PLMNs are deployed inside the same LAN, the MECs next to the UPFs are also inside the same LAN.	Running
FR	Two MECs, 1 at Satory, 1 MEC at TEQMO centre	Research deployment	Via Internet	Operational.

NL	Two MECs at Helmond, one at Siemens, one at KPN	Research deployments	Via Internet (VPN)	Both MECs are operational
KR	0	-	-	
CN	1, East zone of SDAS, Jinan city	Commercial	Fiber Connection by China Mobile	NTP server, ITS Web Server and Cloud App server

3.4. CCAM Trials & Cross-Corridor Cooperation Overview

In this section, we focus on the relationship between CBCs and TSs trial use-cases and introduce the CCAM infrastructure development and integration contributions of each local trial site to the cross-border corridors. The plans for the transfer from local trial sites to cross-border corridors and the integration of the applications, technologies and any other assets are detailed in D3.1 (5G-MOBIX, Deliverable D3.1: Corridor and Trial Sites Rollout Plan, 2020).

3.4.1. Local Trial Site - Cross-Border Corridor CCAM Infrastructure Integration

Apart from the four European trial sites (Netherlands, Germany, Finland, France), 5G-MOBIX has an active collaboration with the Chinese (CN) and Korean (KR) sites, as is obvious by the inclusion of the CN and KR user story definitions, architecture and trial setup plans. However, as the CN and KR sites are not receiving any EU funding, it is not feasible for them to provide tangible contributions to the European Cross-Border Corridors of 5G-MOBIX, due to both resource restrictions and financial constraints. For this reason, their contributions to the cross-border corridors are excluded from this section.

The focus of the project is to realize user stories that are categorized in five main use cases, which will be tested at the trial sites. The 5G-MOBIX user stories are:

- Advanced driving
- Platooning
- Extended sensors
- Remote driving
- Vehicle Quality of Service support

5G-MOBIX trial sites offer different user stories, environments and circumstances, using a variety of equipment, all of them aiming to showcase the effect of 5G in cooperative, connected and automated mobility. For more details about the distribution of use case categories and each of the user stories among the different sites, please refer to D2.1 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019). Table 5 provides a brief overview of the CCAM infrastructure contributions of the local trial sites to the two cross-border corridors' user stories listed above. CCAM infrastructure contributions cover all WP3/T3.4 related activities, such as, infrastructure deployment and enhancement activities and CCAM platform and application development for the infrastructure.

Table 5: 5G-MOBIX Cross-Border Corridor CCAM Infrastructure Contributions from Local Trial Sites

User Story	Local Trial Site	Cross-Border Corridor CCAM Infrastructure Contribution
Advanced Driving (ES-PT)	NL	**"MCS Application"** **Application of the Manoeuvre Coordination Service (MCS)** used in Cooperative Collision Avoidance (CoCA) user story will be transferred **to the ES-PT** corridor to evaluate the Manoeuvre Coordination Message (MCM) and the MCS communication framework and lane merging service with MCM at CBCs.

Vehicles Platooning (GR-TR)	FI	**"LEVIS Video Streaming Application"** **LEVIS video streaming application**, developed at FI trial site, is transferred **to the GR-TR** cross-border corridor together with 4K cameras, client devices, client and server software to be used in the "see-what-I-see" user story. Initial development and testing of the LEVIS video streaming application in FI TS is performed within the remote driving user story context in a Multi-PLMN environment. Modifications are being made for use in platooning user story in GR-TR CBC, as high-resolution video streaming is going to be shared between vehicles in a platoon to enhance safety and driving experience.
Extended Sensors (ES-PT)	DE	**"Vehicle with multi-SIM OBU solution and MEC Applications"** **Provide vehicles equipped for the extended sensors use case and MEC applications to ES-PT** corridor to deploy their own user story "EDM-enabled extended sensors with surround view generation" within the "HD maps" scenario conditions. Interoperability of RSU and Region of Interest (ROI)-based discovery service, EDM solutions, MEC architectures and networks in different countries (recommendations for inter-country deployments). Benchmarking also the roaming results.
Vehicle QoS Support (ES-PT)	FI	**"Edge Discovery Service"** The **Edge discovery service** that will be transferred by FI TS **to ES-PT** cross-border for the *extended sensors* user story will also be utilized in the *vehicle Quality of Service (QoS) support* user story in assuring the application connectivity across the border. The asset will enable uninterrupted HD video streaming (download), for entertainment purposes, in the QoS support user story at the ES-PT CBC.
Agnostic (ES-PT)	FR	**"Multi-flow fusion server"** FR trial site is transferring a multi-flow fusion server that receives data transmitted over different end-to-end paths. The service will be used for trials of using multi-SIM connectivity, providing error correction and packet reordering functionalities for the data flows transmitted by the vehicles connected to two PLMNs using multi-SIMs.

3.4.2. General Cross-Border Corridor Contribution and Integration Plan

In order to realize the cross-border corridor contributions relevant within the scope of T3.4, TSs and CBCs have worked together following an integration timeline, for the local trial sites to deliver and deploy all the required assets and resources, and to operate the planned user stories in cross-border corridors. From the FR TS the multi flow fusion server will be adapted and transferred for seamless connectivity to ES-PT CBC. Regarding the NL trial site, the MEC application that enables infrastructure centred management with MCMs is installed in the MEC at the ES-PT cross-border corridor, including the complete MCM/MCS communication framework for Edge-based evaluation. The DE trial site is going to transfer and run the EDM-enabled extended sensors approach. This includes the vehicle with multi-SIM OBU solution and MEC Applications and a portable PC5-capable RSU. The edge discovery service developed at FI TS is transferred to ES-PT cross-border corridor. Finally, the LEVIS integration at the GR-TR cross-border corridor is expected to complete the deployment of Jetson TX2 to the leader vehicle, the Real Time Streaming Protocol (RTSP) server to the server, and the player code to the follower vehicle. At the time of writing this deliverable the integration of the cross-border corridor contributions has been concluded and the individual trial sites are starting to trial the described functionality at the cross-border corridors. These activities will be reported within the WP4 scope.

4. OVERVIEW OF INFRASTRUCTURE COMPONENTS

The main infrastructure components that are relevant within the scope of this Deliverable and the definition of the overall architecture have been specified in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019) as:

- Roadside sensor/actuator platform integrated with 5G communication infrastructure
- Roadside unit (RSU) platform with required network interfaces for V2X and cloud communication
- Cloud infrastructure platform integrated with 5G and non-5G auxiliary communication infrastructure for V2X communication
- CCAM Application development and integration with roadside and cloud infrastructure

4.1. Cross-Border Corridors and Trial Sites CCAM Infrastructure Components Overview

This subsection provides a summary of the main infrastructure components at each trial site location, showing the similarity and possible complementarities between the trial sites and cross-border corridors. The distribution of the CCAM infrastructure components based on main component and service categories is listed in Table 6.

Table 6: CCAM Infrastructure Components at Cross-Border Corridors and Local Trial Sites

CCAM Infrastructure Components						
	Road sensors	RSU	RSU MEC (Far edge)	MNO MEC (Near edge)	Cloud	CCAM Services
ES/PT	Traffic Radar, Pedestrian detector, 5G smartphones, ITS Centers, Remote Control Center	Yes	No	Yes	Yes	Complex Manoeuvres, Automated Shuttle, Public Transport
GR/TR	Camera	Yes	No	Yes	Yes	See-through streaming, zero inspection
DE	Camera, traffic analysis, road condition	Yes	Yes	Yes	Yes	EDM, GDM, Edge MANO, edge service discovery

FI	No	No	No	Yes	Yes	Remote driving, video streaming, video crowdsourcing, HD mapping, MEC service discovery
FR	Cameras, LiDAR	No	Yes	Yes	Yes	Infrastructure assisted lane change manoeuvre, different MEC Deployment options
NL	Cameras	No	Yes	Yes	Yes	Roadside assisted merging, Remote driving, Cooperative Collision Avoidance

4.2. Spain-Portugal (ES-PT) Cross Border Corridor

4.2.1. Introduction

The ES-PT cross-border corridor manages 8 different user stories, as defined in Sections 5, 8 and 9 of D2.1 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019). This section describes the development stage and roll-out details of the ES-PT CCAM infrastructure depicted in Section 3 of D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), in order to carry out the defined user stories. The CCAM infrastructure relies on different roadside devices (e.g., radars, pedestrian detectors, RSUs or smartphones), cloud infrastructure elements and remote-control equipment. The ES-PT CBC has been developed from scratch, since none of the CCAM infrastructure elements presented was available before the 5G-MOBIX project.

4.2.2. Overview of CCAM infrastructure components at ES-PT CBC

Figure 4 depicts all the elements involved in CCAM infrastructure within the ES-PT CBC.

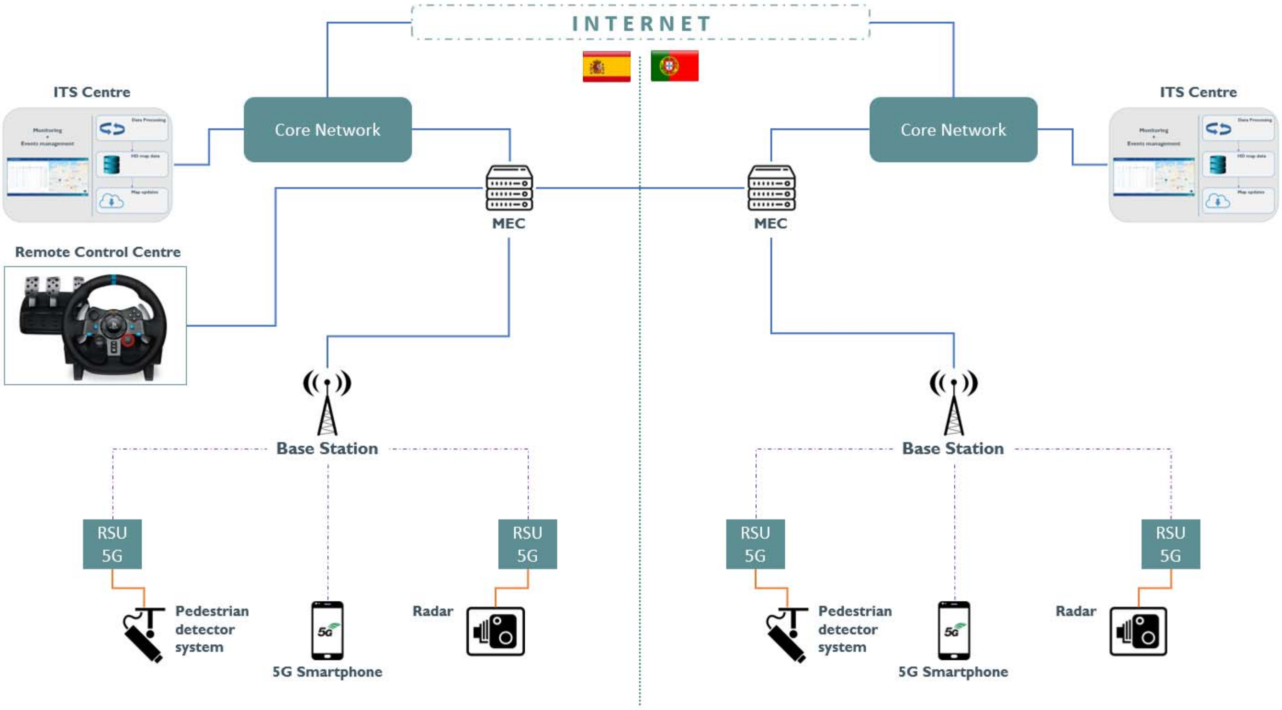


Figure 4: CCAM infrastructure in ES-PT CBC

- **Traffic Radars** are used to provide connected vehicles with information about other vehicles in the road. The radars used in the user stories are provided by Centro Tecnológico de Automoción de Galicia (CTAG) and Instituto de Telecomunicações (IT).

- **Pedestrian detectors** are used to provide connected vehicles with information about vulnerable users crossing roads. The pedestrian detectors used in the user stories are provided by CTAG and Siemens Mobility ITS.

- **5G RSUs** are used as a connectivity link to other sensors (like radars or pedestrian detectors) in order to provide them with 5G capabilities. 5G RSUs used in the user stories are provided by CTAG and IT.

- **5G smartphones** are used as a tool for Vulnerable Road Users (VRUs) to protect themselves by warning near vehicles about the VRU situation. 5G smartphones used in the user stories are provided by Centro de Computaçao Gráfica (CCG).

- **ITS Centers** is used as an ITS platform in the cloud with different objectives: to monitor connected vehicles, to generate road events and notify connected vehicles about them, and to generate updates of the vehicle's HD map. The cloud platform (HW and SW) is provided by several partners: A-to-Be, CTAG and Infrastruturas de Portugal.

- **Remote driving equipment** recreates a vehicle cockpit that is used as a tool for driving vehicles remotely in specific situations. This equipment (HW and SW) is provided by CTAG and Nokia Bell Labs.

- **MEC applications** are some applications such as a Message Queuing Telemetry Transport (MQTT) broker, which is the core of almost every communication in the common architecture of the user stories, the Remote Driving application for enabling the communication between the vehicle and the Remote Control Centre, or the Collision Detection application, which warns pedestrian's smartphones about possible collision risks with vehicles nearby, are developed and integrated into the MECs for complementing the CCAM infrastructure. These software modules are provided by Nokia Bell Labs, CTAG, IT and CCG.

4.3. Greece-Turkey (GR-TR) Cross Border Corridor

4.3.1. Introduction

GR-TR user stories are hosted by four different servers, each running a different application. Application details can be seen in D2.1 section 6 and 7 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019), CCAM infrastructure details can be seen in D2.3 section 3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019). Servers are listed below. More details can be found in section 5.2 in this deliverable.

1. Platooning server, hosted by Turkcell.
2. See What I See server, hosted by ICCS.
3. Assisted "zero-touch" border crossing server, hosted by WINGS.
4. Autonomous Truck Routing server hosted by TUBITAK.

4.3.2. Overview of CCAM infrastructure components at GR/TR CBC

The overall CCAM infrastructure components at GR/TR CBC can be seen on the Figure 5 below:

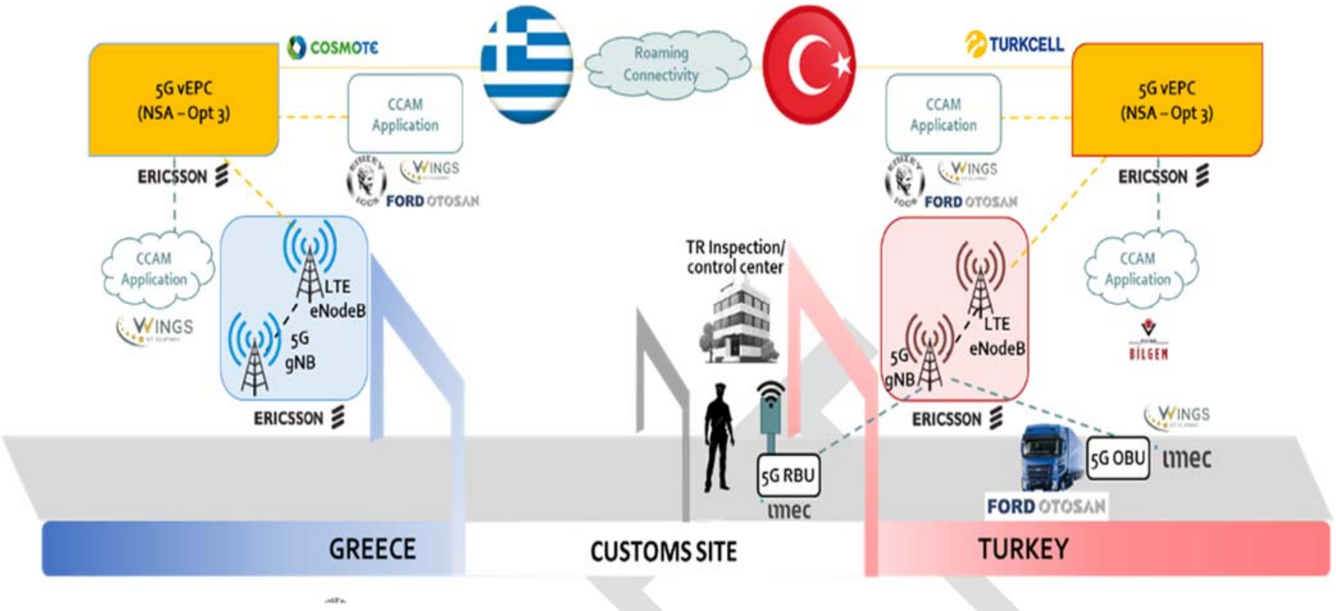


Figure 5: Overall CCAM Infrastructure for GR-TR user stories

Roadside Unit: IMEC will provide three roadside units with 5G, LTE-V2x connectivity. The Velodyne VLP-16 Light Detection and Rangings (LiDARs), provided by Ford Otosan, will be also placed in the RSUs. These LiDARs will be used to increase perception capabilities of the TUBITAK Servers.

As shown in Figure 6 below, the RSU consists of a "weatherproof box", with antennas on top, that is mounted on a free-standing mounting pole. The base of the mounting pole is attached to a standard euro-pallet so the entire setup can be moved as a whole with a forklift.

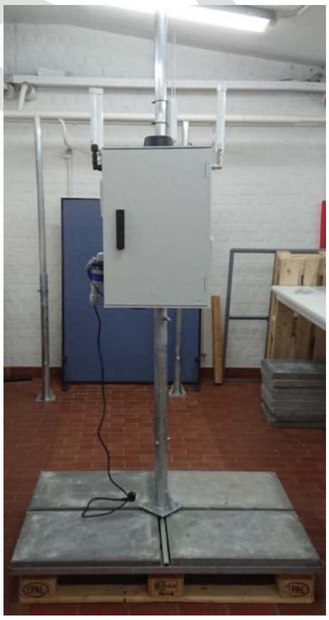


Figure 6: Front View of the RSU

The RSU provided by Interuniversity Microelectronics Centre (IMEC) is mobile. It can be installed outdoors during the testing period and positioned as desired in customs sites. The RSU can be managed and recovered remotely. It is able to provide power and network-uplink to a single LiDAR module. The LiDAR mounted in the RSU is the Velodyne VLP-16. The RSU allows minimal local processing of the data, e.g., converting data formats before sending it to the cloud. Hence, the heavy computations are done in the cloud. For this purpose, the RSU must be able to relay data received from the LiDAR and/or OBU to the cloud over a 5G modem.

The RSU has access to a fallback internet connection. This is needed to allow for remote support / remote configuration of the RSU both during development and when it is deployed on-site. The 5G-uplink included in the RSU cannot be used for this purpose as it is part of the setup being developed/tested and is thus prone to errors/reconfigurations which may cause this link to go down. The fallback-uplink is only used for testing/debugging/reconfiguration. It is not used to relay data to/from the cloud.

The (simplified) architecture of the RSU is shown in Figure 7:

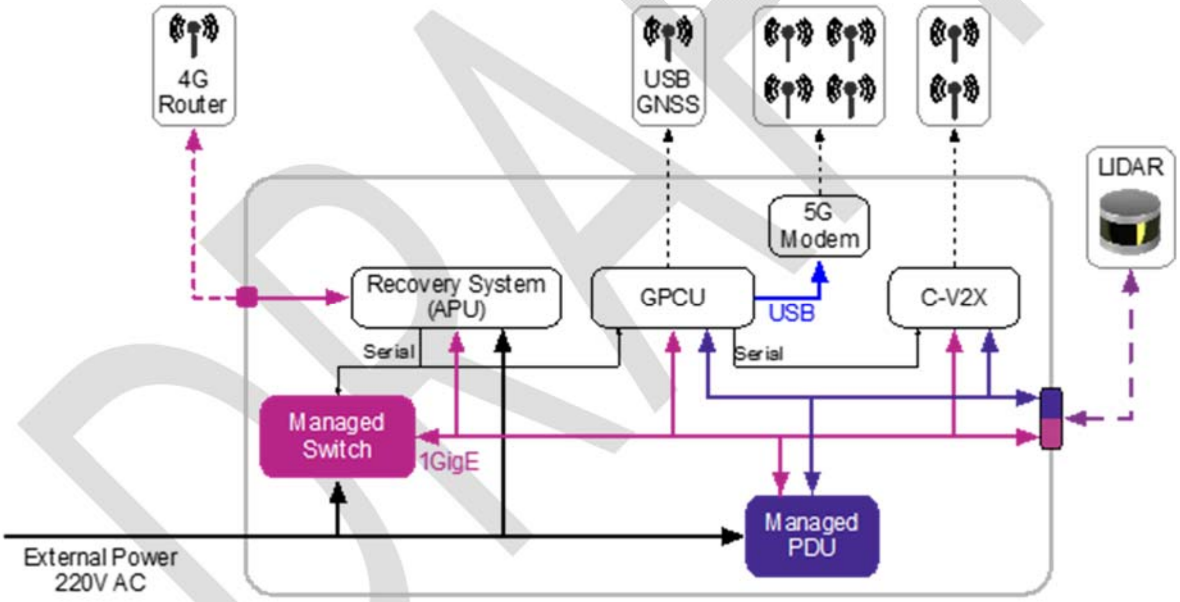


Figure 7: RSU Architecture

As Figure 7 shows, the RSU contains the following components:

- **GPCU (General Purpose Compute Unit)** is a PCEngines APU3D4 embedded PC that acts as the "main controller" of the RSU. It is directly connected to the 5G modem and this provides a 5G Uplink to the rest of the components. In addition, it can also be used for minimal local processing of sensor data received via other components.

- **5G modem** is a Quectel RM500Q 5G modem that is connected via USB3.0 to the GPCU. It is connected to two external SmartEQ LTE Omni Antennas and one SmartEQ AllDisc Combo Global Navigation Satellite System (GNSS) Antenna.

- **C-V2X** is a Cohda Wireless MK6c that is used for direct communication between the RSU and the OBUs (i.e.: to support car-to-infrastructure use cases and to allow the RSU to act as a gateway between the OBU and the cloud in those cases where the OBU has no direct 5G connectivity). The C-V2X module is connected to two external SmartEQ Wi-Fi Dual Band Omni Antennas and one internal Taoglas Magma.AA.107 GNSS antenna.

- **LiDAR:** the RSU is designed to allow a Velodyne VLP-16 LiDAR to be connected to it. The RSU allows data from this LiDAR to be routed to the cloud infrastructure via the GPCU & 5G modem.

- **USB GNSS Receiver** is a Navilock NL8012U USB GNSS receiver that is connected directly to the GPCU. It provides timing & location information to the other components of the RSU.

- **External 4G Router** is an external Mikrotik wAP LTE router that is used to provide a management (4G) uplink to the internet.

- **Recovery System (hereafter referred to as "APU**") is a PCEngines APU2E4 embedded PC that is used to facilitate remote management and remote recovery of the other components of the RSU (GPCU, C-V2X, …). It also acts as a Dynamic Host Configuration Protocol (DHCP) server for the 'internal subnet' of the RSU and has serial connections to both the managed switch and the GPCU to allow these devices to be recovered in case of a network issue.

- **Managed Switch:** A Planet IGS-4215-4P4T managed switch is used to connect all the components of the RSU. By default, all components of the RSU are connected to the same internal subnet, but the switch allows for much more fine-grained control over the network architecture of the RSU if needed.

- **Managed PDU** is a Robot Electronics ETH008 'Ethernet Relay'. It allows the GPCU, C-V2X and LiDAR of the RSU be power cycled remotely in case they get stuck for any reason.

The internal components of the RSU are highlighted in Figure 8, and are explained in Table 7.

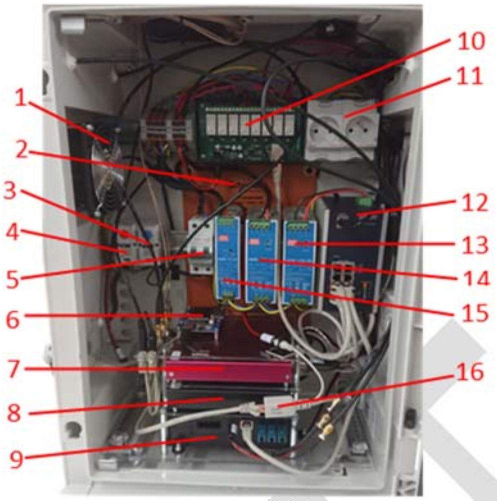


Figure 8: Components Inside RSU

These components are explained in Table 7:

Table 7: Main components of the RSU

1	Exhaust Fan. Turns on automatically when the temperature inside the RSU becomes too high
2	Electric heater. Turns on automatically when the temperature inside the RSU becomes too low
3	Fan temperature control
4	Heater temperature control
5	Circuit Breaker
6	5G modem
7	GPCU
8	Recovery System (APU)
9	C-V2X
10	PDU
11	220V sockets (e.g. for connecting test equipment during servicing)
12	Managed Switch
13	48V PSU: Supplies power to the Switch
14	12V PSU: Supplies power to the APU, PDU, GPCU, C-V2X and LiDAR
15	24V PSU: Supplies power to the external 4G Modem
16	Passive PoE Injector

See-What-I-See application parts and components

ICCS provides all the necessary equipment which is needed for the demonstration of the See-What-I-See application.

Leader Vehicle (LV): it represents the stream sender. On the LV, a Single Onboard Computer (SOC) is deployed, which consists of a Jetson AGX Xavier that has a camera attached to it. Both the Xavier and the LV are connected to outside world (5G network) through the same OBU – Xavier will be connected through an ethernet cable with leader's OBU.

The equipment consists of a Jetson AGX Xavier SOC having a GPU architecture with 256 NVIDIA CUDA cores, a Quad-Core ARMR CortexR -A57 MPCore, and 8GB 128-bit LPDDR4 Memory. The Jetson AGX runs Ubuntu 18.04. A 4K Camera (LOGITECH BRIO 4K PRO webcam) is connected with the Jetson AGX.

- USB camera (Logitech Brio 4K Pro).
- Internet connectivity: USB tethering case: The Jetson will be directly connected through a USB cable with a 5G UE for having the required 5G network connectivity. The 5G UE will be provided by COSMOTE.
- Django 2.1 framework for the client application. The client application allows the vehicle owner to start/stop the live streaming, as well as doing other stuff, which might not be useful for the 'see-what-I-see' use case, such as user authentication, recording locally and upload recorded streams.
- FFMPEG to encode input stream with H264 codec.
- Celery v3.1 for asynchronous time-consuming tasks.
- Redis v2.10 for message queuing.

Streaming Server: It receives the stream from the LV and makes it available to some other receivers.

Equipment consists of a virtual machine in our cloud server (a desktop Dell PC 3847) having 4 cores Intel CPU and 4GB RAM, and is running Ubuntu Desktop 18.04 OS.

- Django v2.1: the server application mainly allows subscribers accessing to the different streams (ex: live streams).
- Redis v3.3.8
- MySQL database.
- Nginx configured to deliver the streams with HLS (HTTP Live Streaming).
- FFMPEG to generate lower qualities from the original stream to offer adaptive streaming experience to the viewing clients.
- Streaming server will be directly connected with the GR edge server in Alexandroupoli premises.

The server's bandwidth to support 4K streaming depends on the number of stream sources (uplink) as well as the number of vehicles receiving the stream. To deliver a 4K stream, it is recommended to have an end-to-end (E2E) bandwidth between 15 and 25 Mbps.

Follower Vehicle (FV): it produces the stream (sent from the LV) consumer or receiver. The FV receives the stream sent by the LV through the application server.

Equipment consists of the receiver machine (Raspberry Pi 3 Model B+ with a 7inch Capacitive Touch Screen LCD (C), 1024×600, HDMI connected with it). Both the RPi and the FV are connected to outside world (5G network) through the same OBU – RPi will be connected through an ethernet cable with follower's OBU.

- It has the player that displays the received stream. It also operates in an Ubuntu Desktop 20.04 (Groovy).
- Raspberry Pi 3 Model B+ (Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz)
- Raspberry Pi 7" 800x480 DSI Touchscreen Display
- Internet connectivity: USB tethering case: The Jetson will be directly connected through a USB cable with a 5G UE for having the required 5G network connectivity.

5. INFRASTRUCTURE DEPLOYMENT AND ENHANCEMENT ACTIVITIES

This subsection provides an overview of the corridor specific infrastructure enhancement and development conducted at the different cross-border corridors to realize the CCAM infrastructure. The activities described in the following subsections focus on Sensor Development, RSU / Edge Deployment and Cloud Deployment.

5.1. ES-PT

The ES-PT CCAM infrastructure counts on a set of elements depicted in Figure 4. Some elements, like traffic radars and pedestrian detectors have been developed by different partners following a common interoperability strategy, in order to ensure every element accomplishes its expected functionality. All these elements are described along the next subsections.

5.1.1. Traffic Radar from CTAG

The basic sensing element of the infrastructure for vehicle's detection in the Spanish side of the border is a long-range radar (Smartmicro UMRR-29) installed by CTAG, which is able to detect passenger cars in the region of interest needed for the intended application.

The detection principle is Time-Of-Flight (TOF), in which 24-GHz electromagnetic wave pulses are emitted by several Tx antennas every 50ms. These pulses are partly scattered back to the sensor and received by a number of Rx antennas. After measuring the time employed in this two-way trip, the distance to the target is calculated. Additionally, comparative analysis of the received wave's phase shift between the different Rx antennas is performed to obtain the azimuth angle. With these two data targets, position is completely characterized. Finally, Doppler shift (a slight shift in received wave's frequency due to target's motion) is used to estimate its velocity.

Therefore, every 50ms a list of objects is sent by the radar with complete information about position and radial velocity for each one of them. This info is fundamentally used to detect the object of interest in each one of the scenarios presented, so that they can be communicated to incoming traffic with which the targets may cross their path. For this purpose, the information from radar detections is shared, via an ethernet cable, with a 5G RSU, which is responsible for parsing this data into ITS messages format and publish them on the network.

The region of interest covered by this radar is shown in Figure 9. It is important to note that "bluer" colours imply a lower sensitivity and thus bigger difficulty to detect objects. As a consequence, it is very important to calibrate and point the radar correctly as detection ranges on the margins of this region of interest are lower.

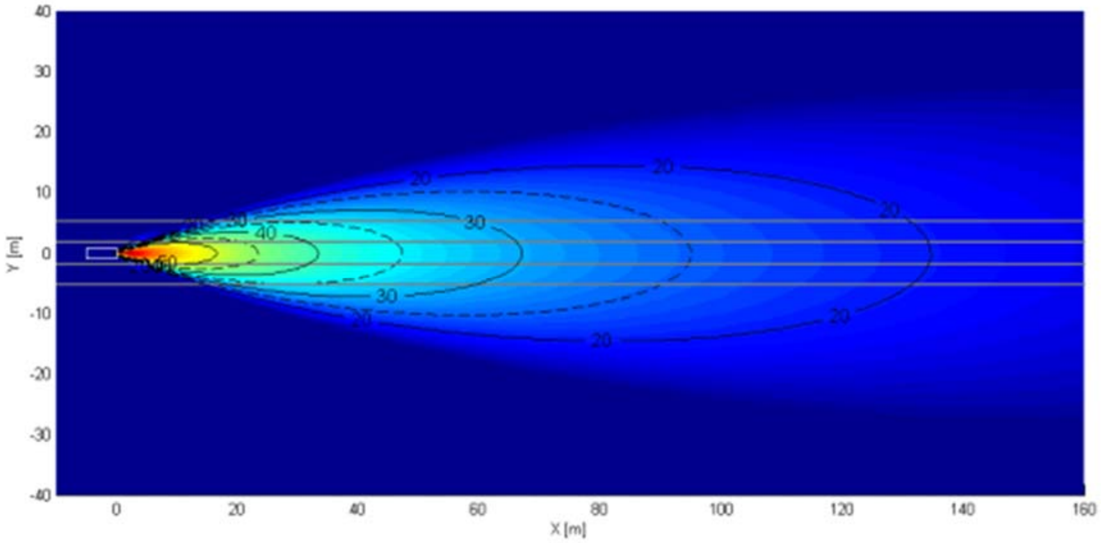


Figure 9: Region of interest covered by Radar (Smartmicro, 2014)

5.1.2. Traffic Radars from IT

In the Complex Maneuvers User Story of the SP-PT CBC, more specifically in the lane merging scenario, a 5G RSU is used to forward data from a traffic radar to the MEC Broker. On the Portuguese side, these radars are installed in the A28 motorway close to Porto's airport and on the A3 motorway next to Valença-Tui new bridge. Figure 10 depicts the radiation diagram for the traffic radar located on the A3 highway. The different coloured areas represent the detection ranges for the distinct classes of vehicles.

The radar model is the UMRR-0C Type 42 from Smartmicro company. It's a Universal Medium Range Radar (UMRR) with very wide horizontal angular coverage, working on the 24GHz frequency band, being able to operate in adverse conditions, almost unaffected by weather, and independent of sunlight, in a wide temperature interval. In comparison with traffic radars used in the SP side, the UMRR-0C is more suitable for highway applications, since it covers a longer range (250 m for passenger cars and 340 m for trucks) and can detect up to 7 classes of vehicles (Pedestrian, Bicycle, Motorbike, Passenger Car, Transporter, Truck/Bus, Long Truck). For calibration purposes, an IP camera is also attached to the 5G RSU, enabling also remote monitoring of the radar detection accuracy. This setup is shown in Figure 11, where all the components forming the road-side infrastructure are integrated (traffic radar + IP camera + 5G RSU).

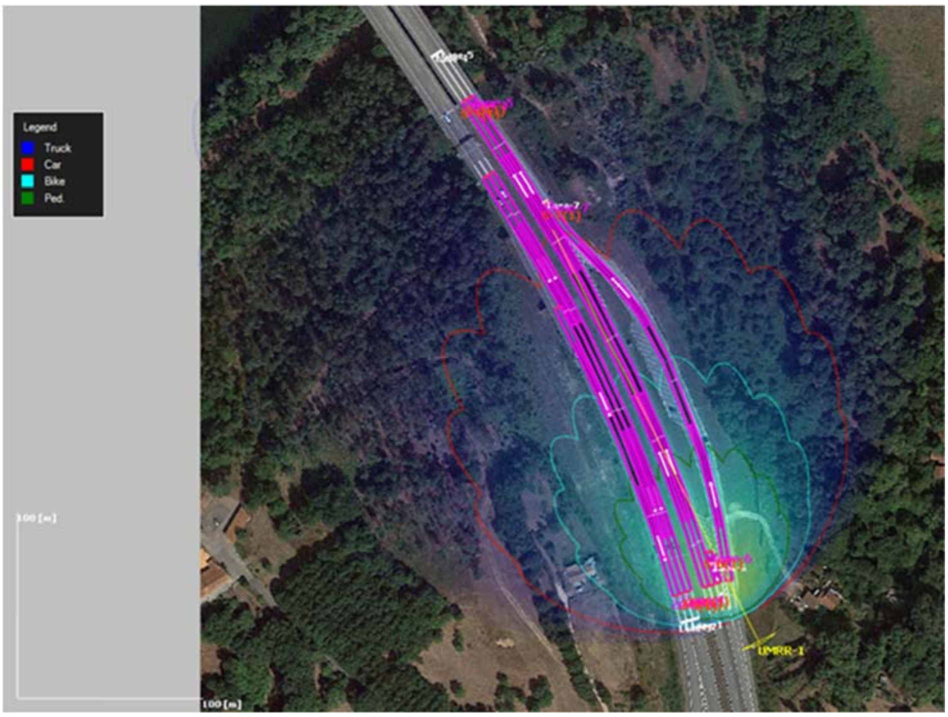


Figure 10: Radiation diagram of the traffic radar installed in A3 highway (PT side) for lane merging scenario

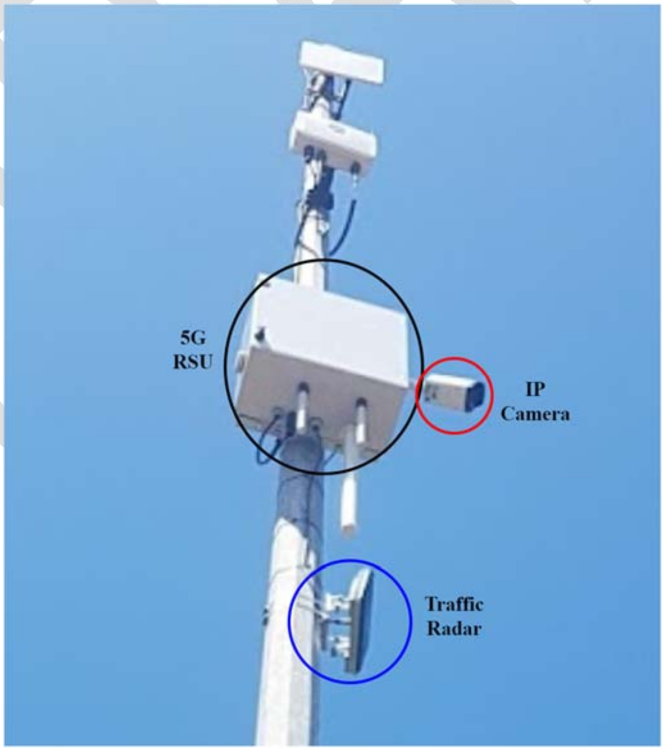


Figure 11: Roadside infrastructure equipment for the lane merging scenario (PT side)

The role of the 5G RSU is to translate the raw detection data from the traffic radar into standard ETSI Intelligent Transportation System (ITS) messages, named Collective Perception Messages (CPMs), and forward them to the MEC Broker, where all information can be accessed by the vehicles and the ITS Center. The complete data flow between the radar and the connected automated vehicles is the following (see Figure 12):

1. The 5G RSU receives the traffic raw data from the radar and uses that information to compose CPMs that are send via the 5G modem to the gNB.
2. The message travels from the gNodeB to the MQTT Broker located in the PT MEC, being published in a topic dedicated to CPM packets.
3. A Geoserver service located in the MEC broker publishes the incoming messages in the outqueue topic (following a geographical tiling scheme), forwarding the messages to the adjacent areas of the sender's location and to other interconnected brokers (in this case the Spanish one). This way, the CPMs also become available to the vehicles connected to the SP MEC broker.
4. The connected automated vehicles subscribe to the MQTT topic of interest, based on their geographical position, receiving the CPMs via 5G Uu interface and processing all information for better data fusion with vehicles' own sensors.

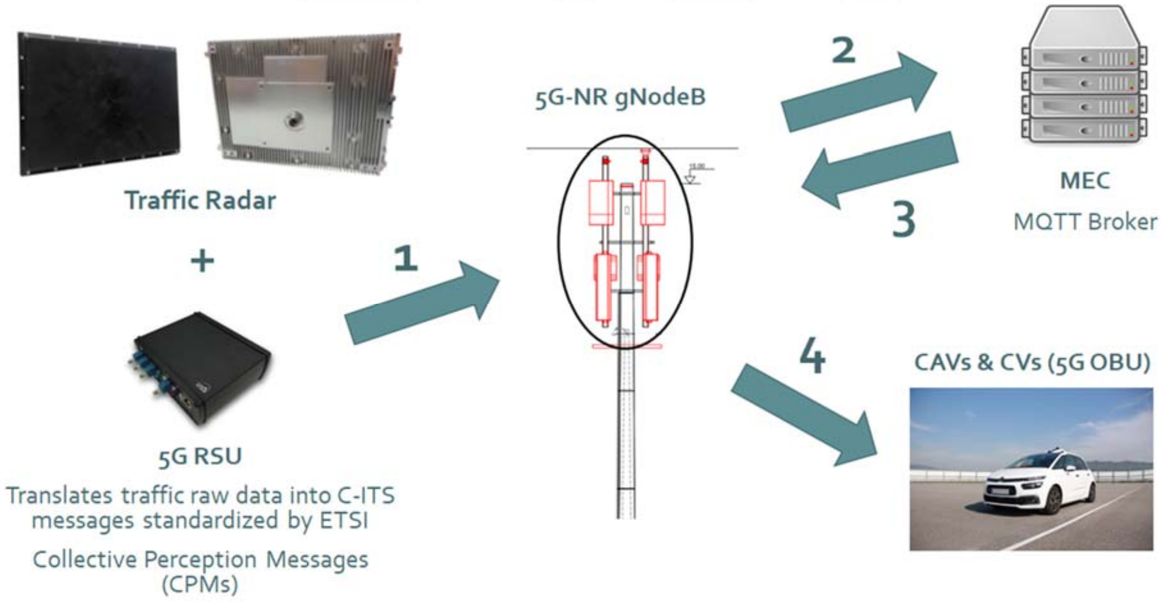

Traffic Radar
+
5G RSU
Translates traffic raw data into C-ITS messages standardized by ETSI
Collective Perception Messages (CPMs)
1
5G-NR gNodeB
2
3
MEC
MQTT Broker
4
CAVs & CVs (5G OBU)

Figure 12: CPM flow from the 5G RSU connected to the traffic radar until the CAVs & CVs (5G OBUs)

5.1.3. Pedestrian detector from CTAG

The pedestrian detection system installed in the Spanish side of the border has been developed by CTAG, and is composed by (see Figure 13):

- A camera which captures the scene. The model of the IP camera used is the UI-5220 from IDS manufacturer, which provides PoE power supply and IP67 protection, which facilitates the installation of the element in outdoor environments.
- An embedded computer equipped with Intel processor, which runs the image processing applications for extracting the moving objects from the image in real-time, and to classify them as pedestrians, by using an image processing pedestrian detector. These applications are developed in C++ programming language, making use of some components of the OpenCV library. Furthermore, the pedestrian detector application is based on machine learning techniques. The information of the detections made is shared with the 5G RSU through an Ethernet interface, using a proprietary JSON data format which contains the presence or absence of pedestrians, the latitude and longitude of that ones and the date and time of the detection with accuracy of hundredths of a second.
- A 5G RSU, described in section 5.1.6 of this document, which is responsible for creating ITS messages with the information received from the system, and for sending these messages to the network in order to make them available for connected vehicles.

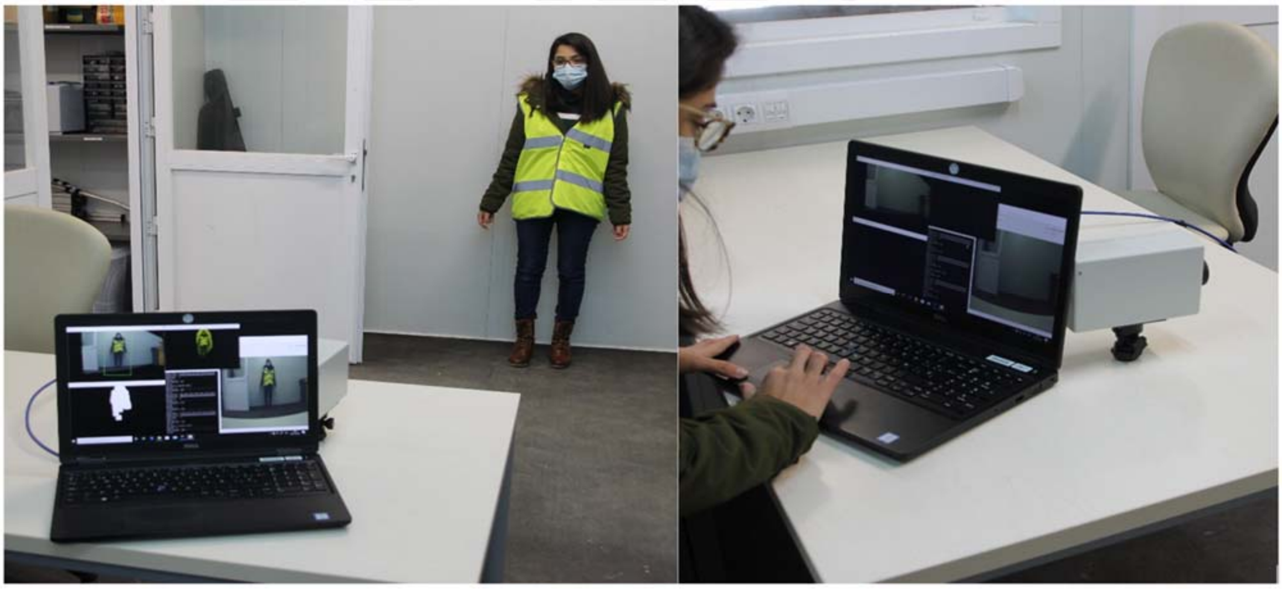


Figure 13: CTAG pedestrian detector from Siemens

The element for detecting the pedestrians, in the Portuguese side, will be the radar (Smartmicro UMRR-11 Type 44) installed by Siemens Mobility ITS, which is able to detect pedestrians in the region of interest.

The sensor measures range, radial speed, horizontal and vertical angle, reflectivity and other parameters of multiple stationary and moving reflectors (targets) simultaneously. It is capable of high definition (4D/HD), where HD resolution means that the sensor can separate objects regarding their speed and their distance to the sensor. The sensor is almost unaffected by weather, temperature and lighting conditions.

A 4D Doppler based radial motion detection principle includes:

a) Direct unambiguous Doppler measurement (speed)
b) Direct range measurement
c) Direct azimuth angle measurement (horizontal angle)
d) Direct elevation angle measurement (vertical angle)

Moving reflectors with an absolute radial speed component of typically >0.1m/s can be detected as well as stationary objects. With its multi-target capability, the sensor can detect many reflectors within the field of view at a time. The field of view typically covers up to six lanes. Additionally, filtering algorithms are implemented for tracking all detected reflectors over time. Those tracking algorithms are integrated in the sensor. Multiple objects can be tracked simultaneously.

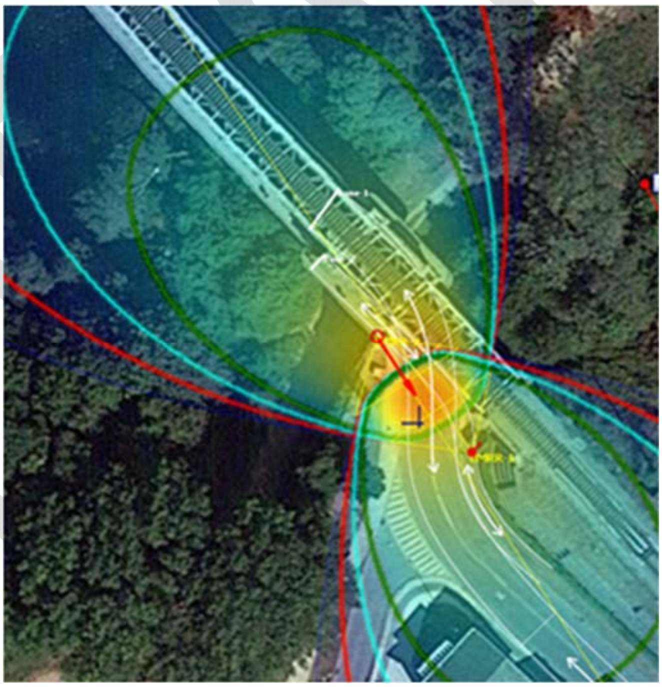


Figure 14: Radiation diagram of the traffic radar on the Portuguese side of the Old Bridge Valença

In the use case to be performed, in the Old bridge of Valença, the goal is to detect the pedestrians that will cross the path of the autonomous shuttle. The location for the installation of the radar was defined, taking into account the most critical places for pedestrians. Figure 14 depicts the radiation diagrams for the two possible locations of the traffic radar, that can be identified as red circles.

The locations of the pedestrians detected by the radar will be forwarded to the RSU developed by CTAG, which is responsible for creating the ITS messages that will inform the autonomous shuttle of the exact location of each pedestrian.

5.1.4. Pedestrian detector from Siemens

The implementation of VRU on the Portuguese side is a collaboration between CTAG and Siemens Portugal and the installation is a partnership between NOS and IP for the use of some of its infrastructures. In this collaboration Siemens provides the radar (detection and processing) and sends the information to the RSU of CTAG (creation of CITS messages and sending to the MEC).

To carry out this integration, a messaging and communication protocol was defined between the two teams and a simulator of this messaging was created to be able to simulate the radar, even if we did not have it. After the tests with the simulator, real logs were requested from Siemens and injected via Ethernet, checking the output to the MEC.

Finally, as the RSU had to do part of the radar processing, the load and saturation were tested.

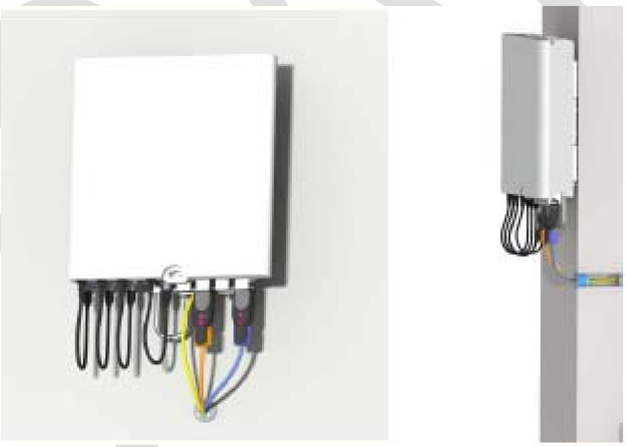


Figure 15: Siemens pedestrian radar

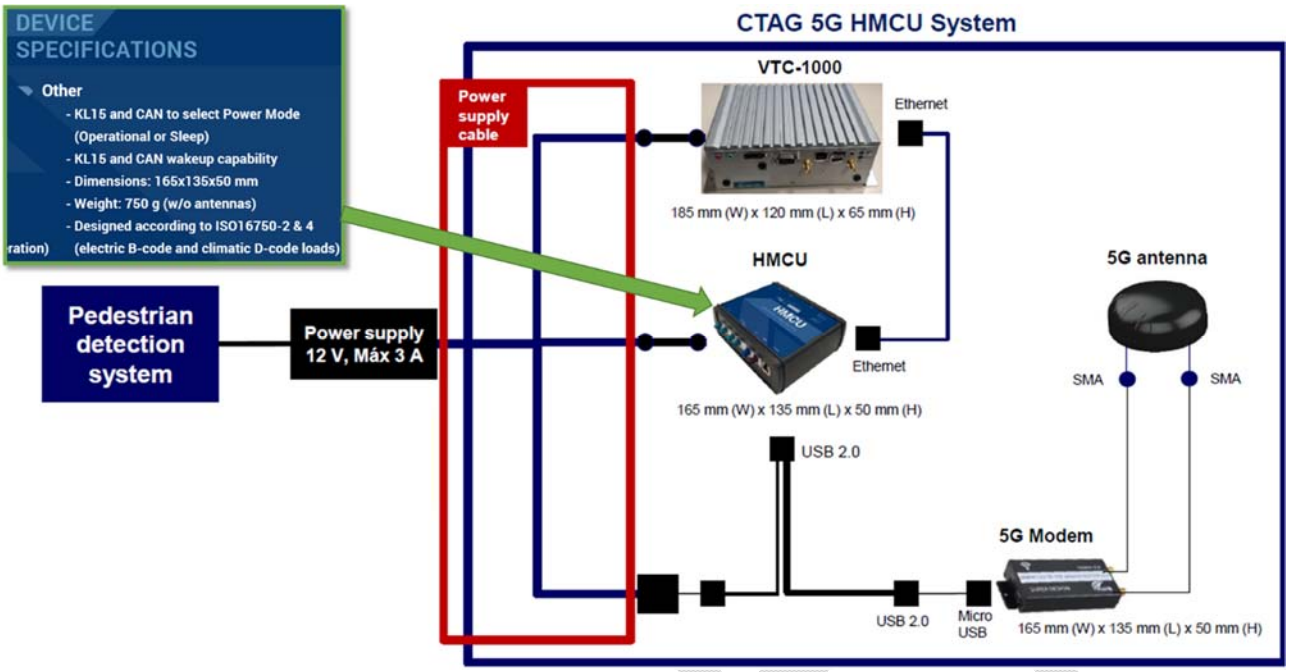
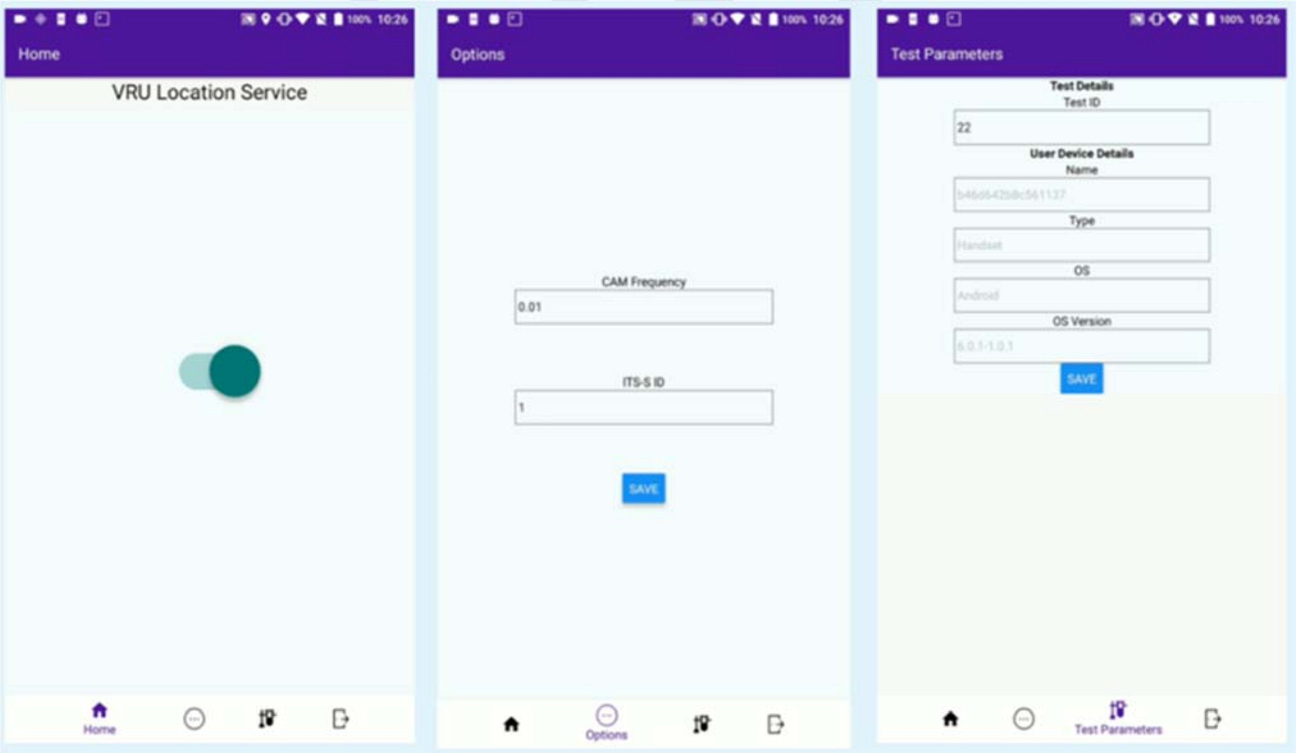

DEVICE
SPECIFICATIONS
Other
- KL15 and CAN to select Power Mode (Operational or Sleep)
- KL15 and CAN wakeup capability
- Dimensions: 165x135x50 mm
- Weight: 750 g (w/o antennas)
- Designed according to ISO16750-2 & 4 (electric B-code and climatic D-code loads)
ration)
CTAG 5G HMCU System
Power supply cable
VTC-1000
Ethernet
185 mm (W) x 120 mm (L) x 65 mm (H)
HMCU
Ethernet
165 mm (W) x 135 mm (L) x 50 mm (H)
5G antenna
SMA
SMA
Pedestrian detection system
Power supply 12 V, Máx 3 A
USB 2.0
5G Modem
USB 2.0
Micro USB
165 mm (W) x 135 mm (L) x 50 mm (H)

Figure 16: Overview on RSU by CTAG

5.1.5. VRU Smartphone Application from CCG



VRU Location Service

CAM Frequency: 0.01

ITS-S ID: 1

SAVE

Test Details

Test ID: 22

User Device Details

Name: b46d6425b8c561137

Type: Handset

OS: Android

OS Version: 6.0.1-1.0.1

SAVE

Figure 17: Overview of the VRU app interface for Enable/Disable location service, set CAM parameters and log information

VRU app

The Vulnerable Road User (VRU) app, presented by Figure 17, runs in a smartphone with Android OS. This app works as an alternative or as a complement to pedestrian radars, detailed in this document. The VRU app connects a Vulnerable road user to the Intelligent Transport System (ITS) service, through a CCAM service. The VRU Apps main function is to constantly send Cooperative Awareness Messages (CAM) to the MEC with information about the VRU location. This location is recorded by the smartphone and shared with the ITS infrastructure (see Figure 18).

The application uses the CAM message standard. The structure is the same used by vehicles. However, the field *stationtype* is set as *pedestrian* (value 1), an option available in the CAM standard. The CAM message used in the application only uses the required fields, which includes *stationId*, *stationtype*, *refposition*, *speed*.

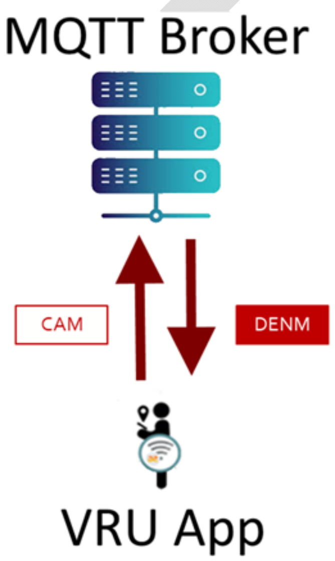


Figure 18: Schematic of the connectivity of VRU App

The modules of the App shown on Figure 19 collaborate in the following way. At a fixed frequency (configurable within the App, but usually between 10 and 1hz to comply with the ETSI ITS CA standard), the Timer module triggers the generation and sending of a CAM message. The CAM Generator module then uses the data provided by the device location services to create an object with the structure and data of a standard-compliant CAM message. This object is then sent to the Encoder module, where it is encoded into an Abstract Syntax Notation One (ASN.1) UPER CAM message. The resulting message in then sent, by the Sender module, to the MQTT broker on the MEC, specifically to the topic corresponding to the tile where the VRU is located. In the meantime, the Logger module records all data sent (with a timestamp and other metadata) in a database, from which it can be exported for future analysis.

Next steps include the reception of Decentralized Environmental Notification Messages (DENMs), with warnings about dangers to the VRUs, namely possible collisions with road vehicles. This will require the collaboration of several of the App's modules, according to the following workflow:

The message is received by the Receiver module as an ASN.1 UPER encoded DENM message, which is immediately sent to the Decoder module, where the received message is decoded and transformed in an object with usable data. This data is then processed, and the notifications module generates a notification warning the user of the danger (the specific way in which the VRU will be alerted is still being developed). Once again, the Logger module records the messages received with metadata for future analysis. It must be noted that the App will only receive DENM. The collision detection must be made on the MEC side. As such, the App would be prepared to receive DENMs of any kind, besides the one envisioned for collision warning.

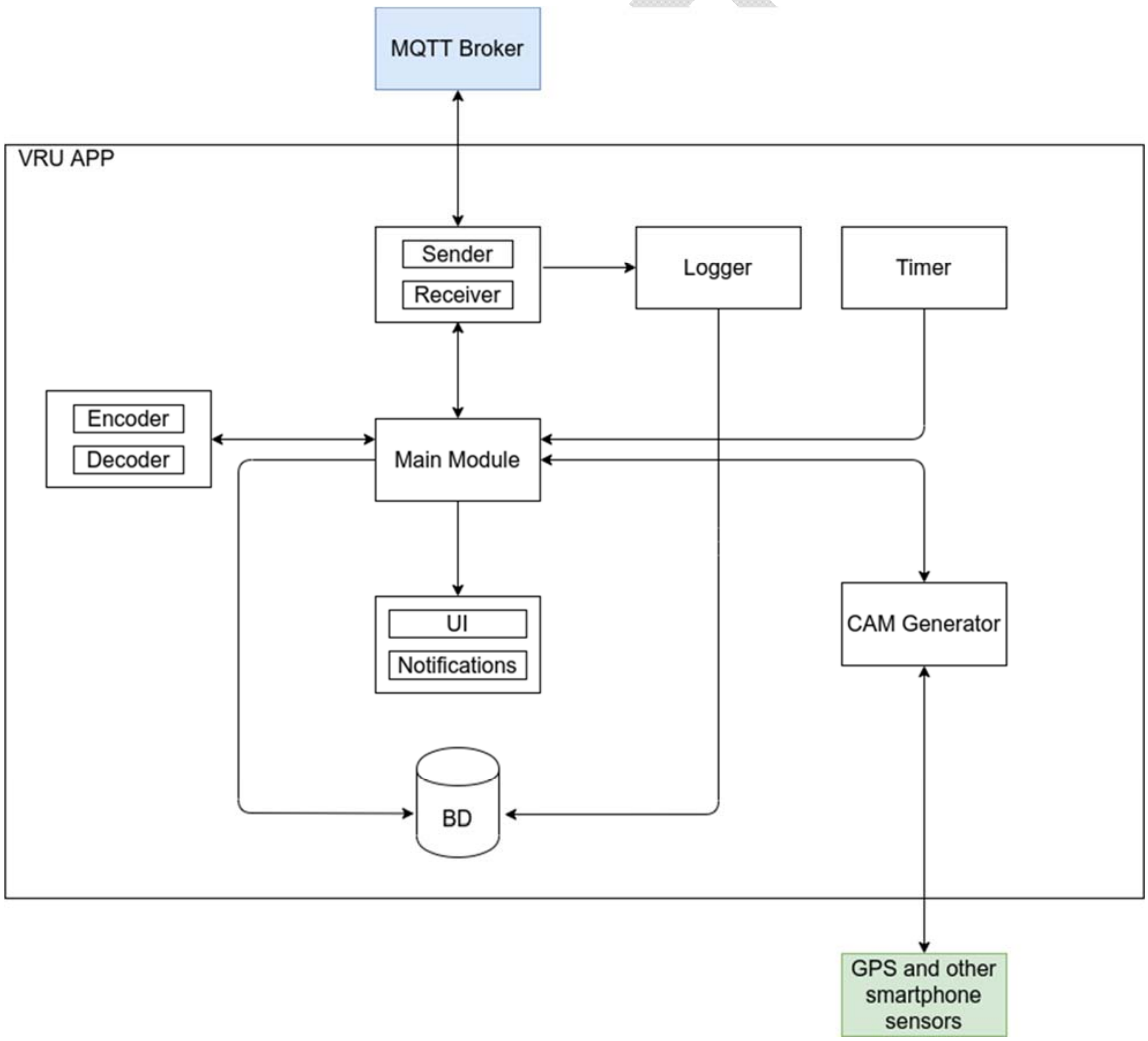


Figure 19: Overview on VRU app components

5.1.6. RSUs from CTAG

All the infrastructure sensors which are available on the Spanish side of the border, such as the pedestrian detector as well as the traffic radar, are provided with 5G connectivity through a 5G RSU designed, developed and integrated by CTAG.

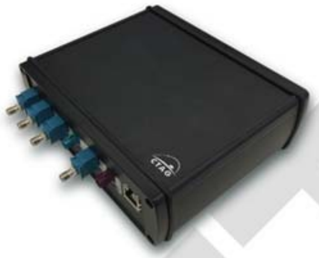


Figure 20: Picture of a CTAG RSU

This electronic equipment integrates different communication technologies, such as 3G/LTE, ITS-G5 and 5G. The 5G connectivity is provided by the Quectel RM500Q external modem, which is connected to the RSU through USB interface.

The RSU provided by CTAG (see Figure 20) relies on a custom Linux distribution that can be customized for different purposes. The software communication stack deployed is JAVA based and run on a OSGI Framework, allowing then a very flexible environment for the implementation and development of different connectivity solutions. Specifically related to the ITS Stack, ETSI Standards have been followed for needed components as described in (ETSI TR 103 562, 2019), (ETSI EN 302 637-3, 2014) and (ETSI EN 302 637-2, 2014).

The role of this equipment is to receive the data collected by the sensors. Each RSU is capable of decoding the data format of its attached sensor. This data is the input of a real-time application which parses the information of each message from the sensor, into the corresponding ITS message format. For traffic radars, the messages generated are CPM, whereas for pedestrian detectors, DENM messages are used. Afterwards, these messages are published in the proper topic of the MQTT broker within the ES MEC.

5.1.7. RSUs from IT

The RSUs on the PT side are based on an Advantech board (FWA-1112VC) and include a 5G module from Quectel (RM500Q), enabling the interconnection with the 5G network, being the only interface that allows remote access to the RSU. Besides that, the 5G RSU is also connected: to the UMRR-0C traffic radar for traffic raw data acquisition; to an IP Camera for road monitoring and radar calibration; via USB to a GNSS Receiver for accurate localization; and to a timeserver (Stratum 1) running PTP protocol for precise synchronization as shown in Figure 21.

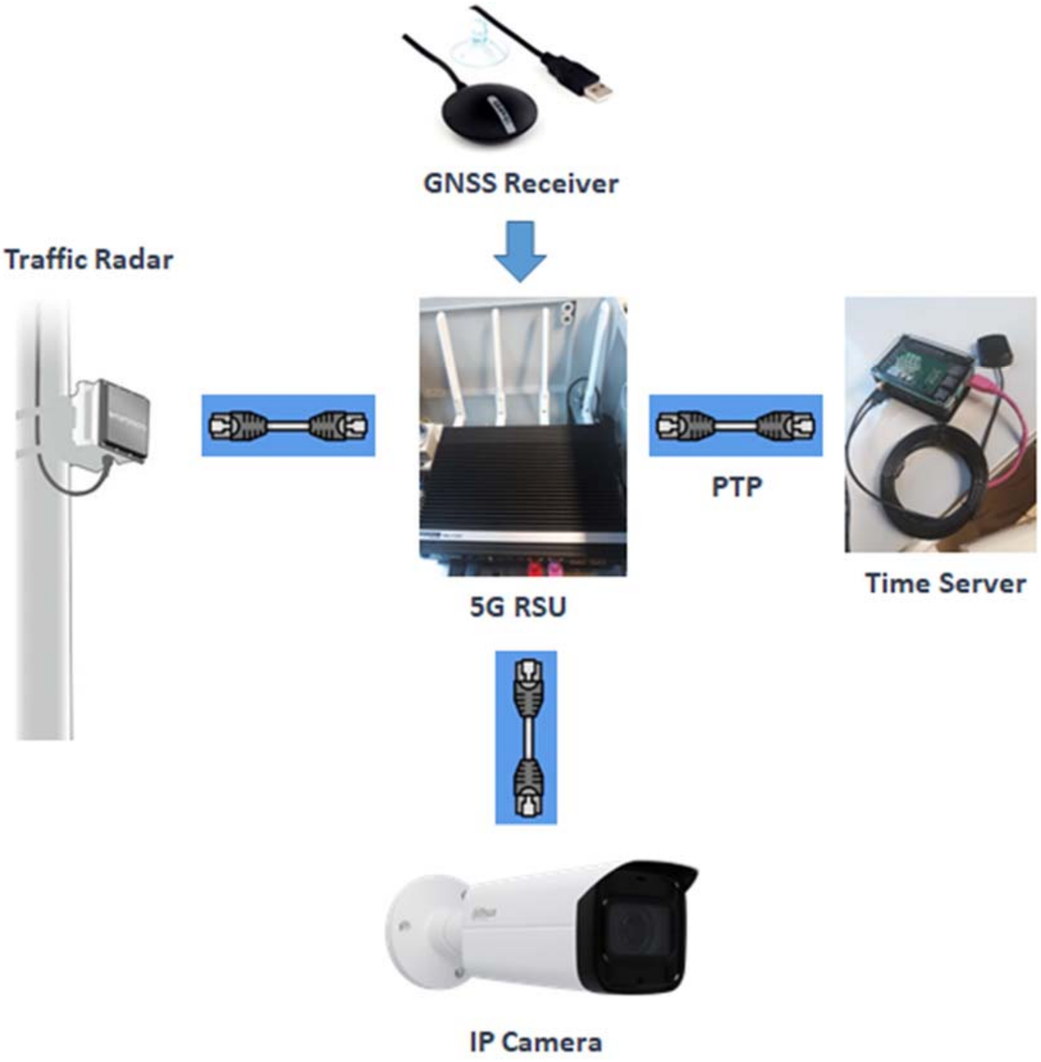


Figure 21: 5G RSU and roadside infrastructure equipment (PT side)

The 5G RSU runs a Linux-based machine and is configured with two different IP addresses over the 5G interface, one to connect with the MQTT Broker in the MEC node and another one for remote access through the internet. Regarding the MEC connection, it is used to send the ETSI CPM packets, that are built based on the traffic raw data from the radar, to the MQTT broker. The RSU is connected to a specific inqueue topic of the broker, which is associated with the station ID of the RSU, the tile where the RSU is located and the message type (CPM).

The internet VPN connection is used for RSU and sensors' management, control, configuration and remote update. For instance, it allows access to the camera's video stream or to the traffic statistics and real-time detection from the radar through a remote Virtual Private Network (VPN) connection. In order to access all these devices from a single IP address reachable through the VPN, Source and Destination Network Address Translation (NAT) Port Mapping is used.

5.1.8. Remote control equipment

There are some specific hardware and software components that need to be installed in the control site premises in order to be able to drive remotely a vehicle. Basically, it is a PC with a powerful GPU card which will be linked to a VR HMD headset from which the user will be able to drive a vehicle in a mixed reality environment. This software is also integrated with the Remote Driving Protocol part in charge of the control and the telemetry.

The HMD application has been developed using Unity with customized Gstreamer plugins that allow to render different live video streams in the VR environment.

This application is integrated with a NodeJS application that provides the feedback of the telemetry and controls a Logitech G29 wheel and pedals sending the commands remotely to the MEC which will forward them to the vehicle.

A description of this setup as well as the installation instructions can be found in the annexes at the end of this document (10.7.1).

5.2. GR-TR

5.2.1. Platooning with "see-what-I-see" application

The platooning application requires data from different sensors, namely: vehicle radar, camera and GNSS. Detailed information about these sensors can be found in D3.2, section 4.2 (5G-MOBIX, Deliverable D3.2: Vehicle adaptation for CCAM use cases, 2020).

As it refers to the See-What-I-See video streaming application, the equipment is deployed by ICCS at the GR and TR edge server infrastructures with the client devices installed and integrated with the trucks. Both servers can be involved in the application's demonstration and functionality. The camera placed in the first truck will start the video streaming and the follower will receive it through the application servers installed and integrated with edge servers. The application's infrastructure is depicted below in Figure 22.

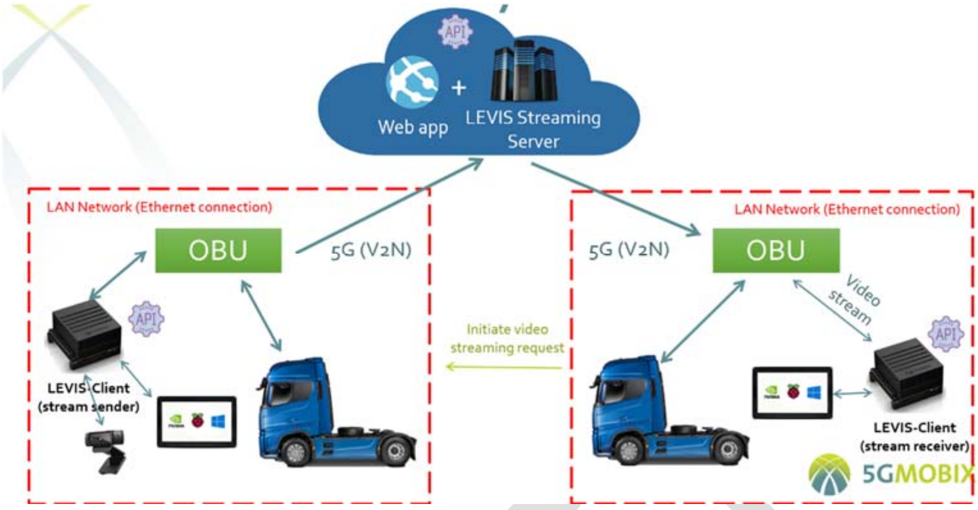

API
Web app
+
LEVIS Streaming Server
LAN Network (Ethernet connection)
OBU
5G (V2N)
API
LEVIS Client (stream sender)
Initiate video streaming request
LAN Network (Ethernet connection)
5G (V2N)
OBU
Video stream
API
LEVIS-Client (stream receiver)
5GMOBIX

Figure 22: Overview on See-What-I-See streaming application infrastructure

5.2.2. Assisted "zero-touch" border crossing application

In order to realize the "Assisted zero-touch border-crossing" user story at the GR-TR the following equipment have been deployed by WINGS at the customs sites of the two countries.

Origin country customs site (e.g., GR)

- Roadside infrastructure including still-frame camera for transmitting figures for license plate recognition, and a smart traffic lights which can receive instructions (green, red).
- One laptop/tablet/smartphone running the *customs agency GUI* where relevant information and warning messages are displayed (connection to the WINGS platform necessary)

Destination country customs site (e.g., TR)

- One laptop/tablet/smartphone running the *customs agency GUI* where relevant information and warning messages are displayed (connection to the WINGS platform necessary)
- Database/server with stored truck manifest/lists information to be transmitted to the neighbouring country for confirmation of measurements. May be co-located with the customs agency GUI laptop.

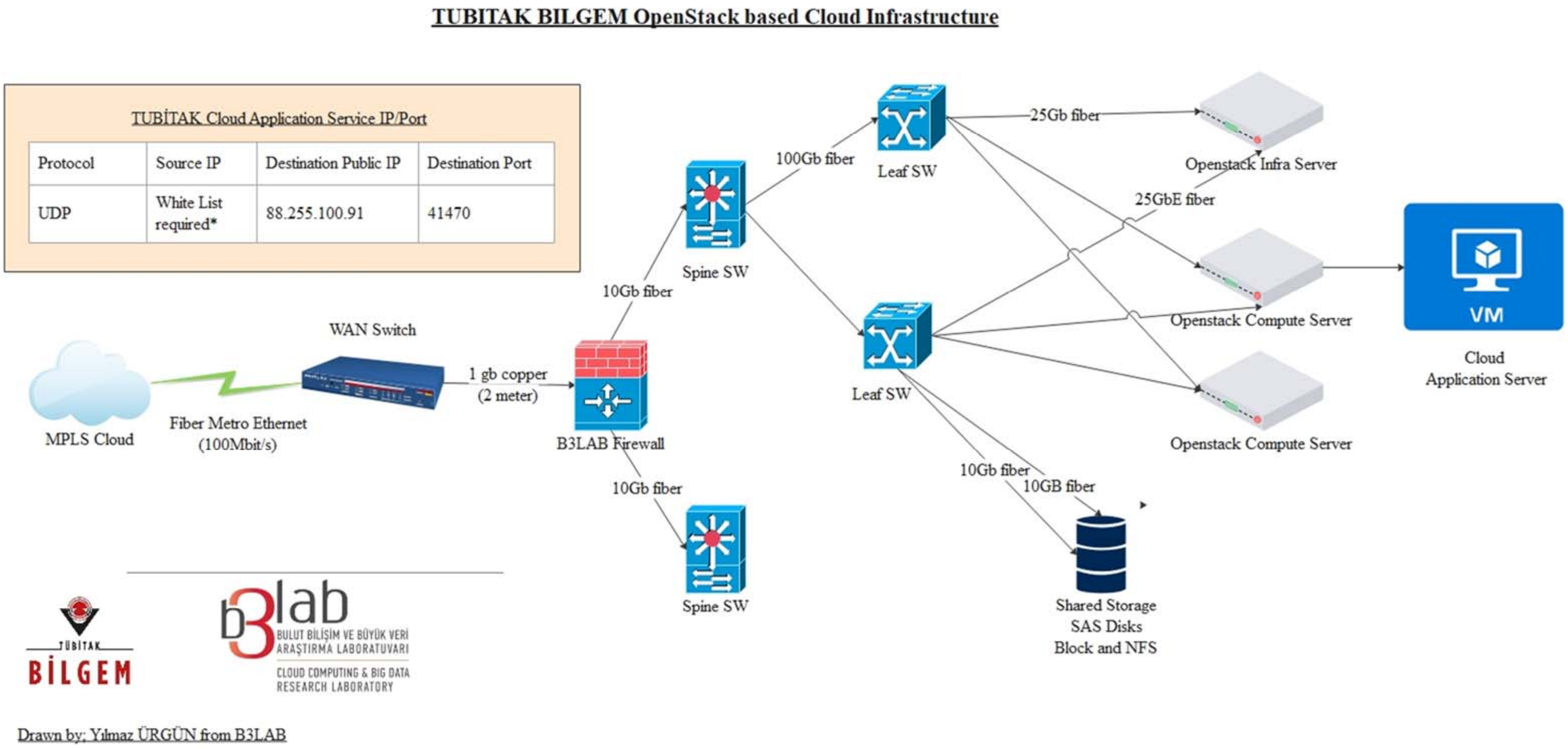
The sensors deployed at the customs sites are designed to capture and send real-time image frames for the licence plate recognition and to control the border bar and the traffic lights. The Raspberry pi-3, with built-in Wi-Fi connectivity, is used as a computation platform. The Quectel RM500Q chipset has been integrated into the road-side sensors to provide 5G connectivity. Power supply is provided from AC voltage. Once powered up, the RSU should start transmitting image frames, while continuous request data to adjust traffic lights and bar status.

5.2.3. Truck routing in customs site application

OBU and RSUs developed by IMEC are used for the Truck Routing user story. Detailed information about these components can be found in Section 4.3.2 for RSUs, and D3.2 (5G-MOBIX, Deliverable D3.2: Vehicle adaptation for CCAM use cases, 2020) for the OBU.

5.2.4. TUBITAK Cloud for Truck Routing

Truck Routing Application developed for Truck Routing user story are deployed at the TUBITAK Cloud. TUBITAK Cloud is an Openstack based (Pike version supported by NASA and Rackspace) cloud computing solution that provides platform and software services as well as standard infrastructure services. For this project, a virtual machine was created on the cloud and configured as Truck Routing Server. For safety issues, TUBITAK Cloud is located in a secured area and only authorized users can access the building. Cloud tenant's private networks are isolated with different VLANs. There is a firewall located between the cloud and the internet. Openstack Security groups are using tenant's inbound and outbound access as additional security layer. B3LAB Cloud is based on IaaS (infrastructure as a service) model, so tenants are responsible for application and other security. Also, whitelisting is required in order to send messages to virtual machines. For this user story, OBUs and RSUs are added to the whitelist and incoming messages from these entities to the cloud are forwarded to the Truck Routing Application Server. Figure 23 depicts the TUBITAK Cloud infrastructure.



TUBITAK BILGEM OpenStack based Cloud Infrastructure

TUBİTAK Cloud Application Service IP/Port

Protocol	Source IP	Destination Public IP	Destination Port
UDP	White List required*	88.255.100.91	41470

Drawn by: Yılmaz ÜRGÜN from B3LAB

Figure 23: TUBITAK Cloud architecture

6. PLATFORM AND APPS EVOLUTION

This section covers the activities conducted by the CBCs to evolve and enhance the individual CCAM platform with additional applications, interfaces or tools which in the end allows them to attain the specific use case functionalities of the corridors.

6.1. ES-PT

6.1.1. Cloud applications

ITS Centers

The ITS Centers in Spain and Portugal support the Advanced Driving and HD Maps use cases with monitoring, sensor data processing and HD Maps distribution. In the ES-PT CBC, two similar mirrored ITS Centers have been deployed, each one intended to serve vehicles traveling on its side of the border. These ITS Centers are composed of various modules:

Monitoring and Traffic Management module aims to be the control center of the entire cooperative ecosystem, allowing for the management, monitoring and actuation of the different connected and cooperative elements. This module allows to monitor connected vehicles' states, as well as triggering traffic events for vehicles that can receive them. This module has been developed by CTAG, and the same one is used for each ITS Centre.

HD Maps module aims to add new functionality to the ITS Center, enabling it to support the HD maps generation for autonomous vehicles. This module has been developed in parallel (two instances) by CTAG and A-to-Be, so that the Spanish ITS Centre deploys CTAG's module, and the Portuguese ITS Centre deploys A-to-Be's module. Although being developed in parallel, the HD maps module keeps a common line of implementation, which works as follows: the module receives a package with raw sensor data from a vehicle. This data is processed in order to extract the features of the road which are inserted into a JSON file. Afterwards, this JSON file is used in two ways: it is published over the network, in order to be accessed from any vehicle able to generate its own proprietary HD map; and this JSON is used too as an input in a second phase of processing in the cloud, which generates an updated HD map, that is later distributed between vehicles using this kind of map format. The component that uses the JSON file to produce and distribute an updated HD Map in the cloud is the same in both Portugal and Spain and is provided by CTAG.

ATMS integration module is an additional component not planned initially but added with the July 2020 amendment. It will only exist on the Portuguese side and will integrate InfraPT's ATMS with the 5G network so that alerts from InfraPT's ATMS can be disseminated as DENM messages to vehicles. This will be used to trigger the HD Maps Scenario that is part of the Advanced Driving user stories and starts with the Alert of a

Roadworks situation. This integration will be achieved using the Datex II standard with a component provided by A-to-Be called A-to-Be Mobics.

In terms of integration, it is important to consider the data flows of the ITS Center components between each other, in the same country and between countries, with the MEC Brokers, directly to vehicles and to other partner systems. We present this analysis in Figure 24:

Monitoring and Traffic Management and **HD Maps modules**:

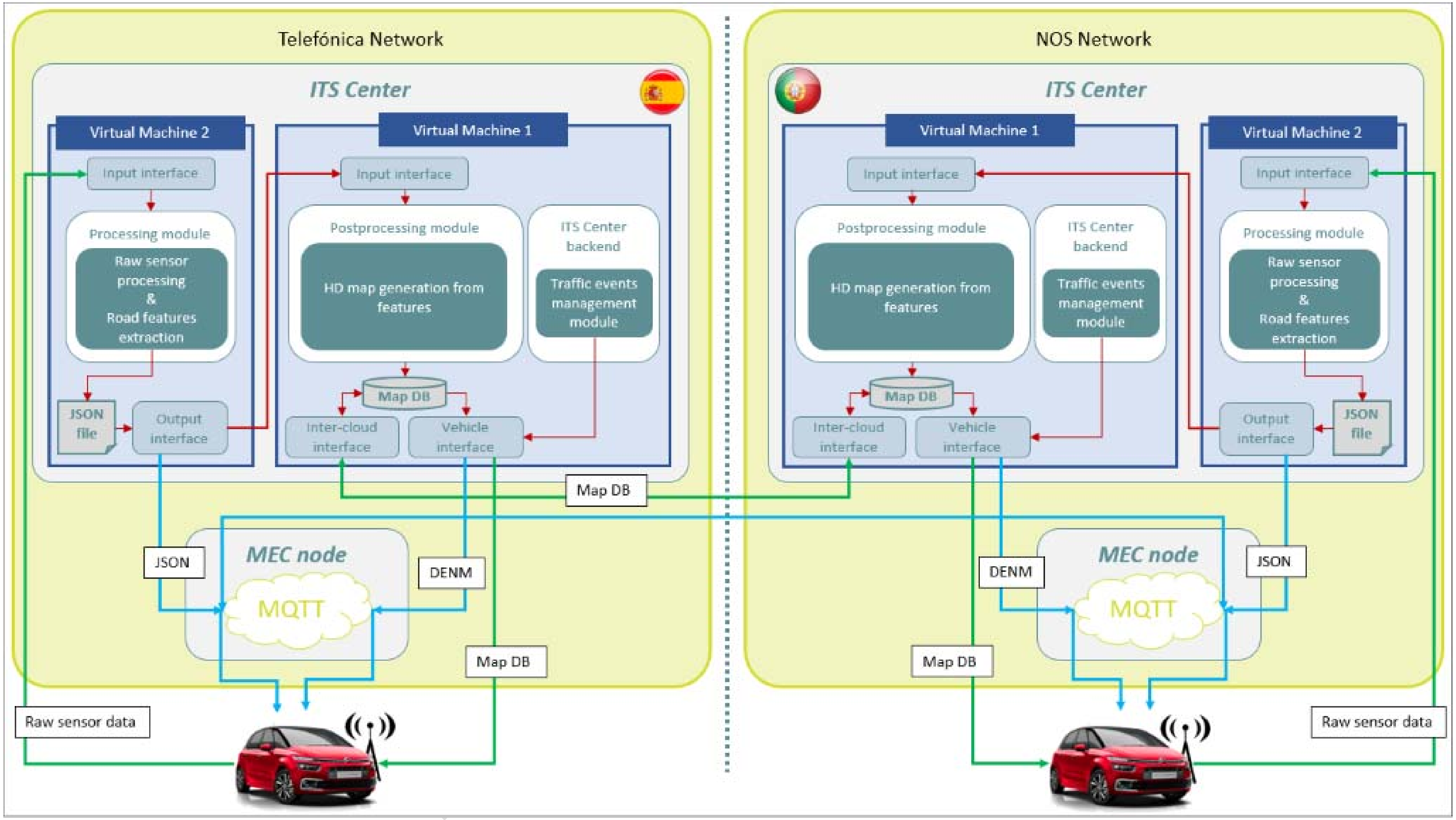


Figure 24: Cloud architecture for Monitoring & Traffic Management and HD Maps modules

Within this architecture, each ITS Center instance is composed of two different elements (i.e. two virtual machines): the first Virtual Machine (VM), VM1, consists of the ITS Centre core (backend, monitoring application, traffic events management module, interfaces with other entities, and so on), whereas the second VM (VM2) is dedicated to processing the data from vehicles and to compose the HD Map's JSON output containing the extracted features of the road.

ATMS integration module:

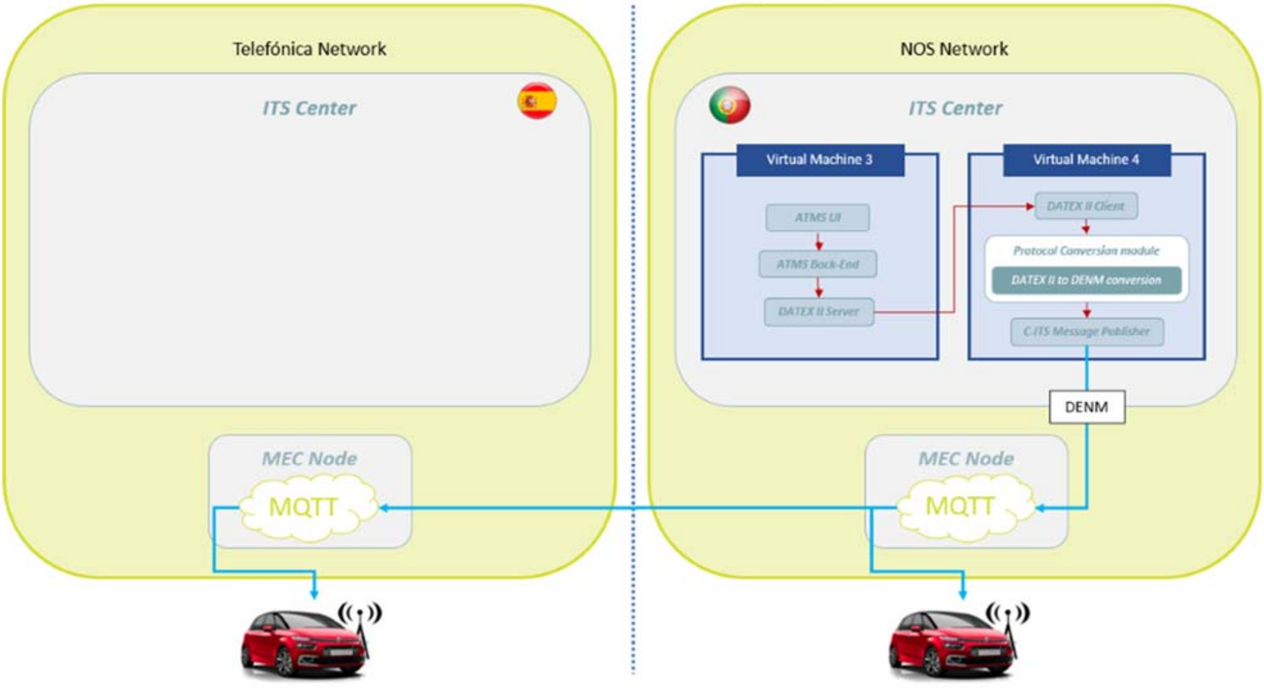


Figure 25: Cloud architecture for ATMS integration module.

Within this architecture, the PT side ITS Center instance is composed of two different elements (i.e. virtual machines): the first VM (VM3) consists of the ATMS Operation and Communications platform (UI, backend, DATEX II server), whereas the second VM (VM4) is used for converting the event data from the ATMS into DENM messages and disseminating these to the vehicles, as depicted also in Figure 25.

We can identify the following data flows and respective integrations in the following table:

Table 8: Data flows and integration

Data Flows	ITS Centre Module	Other party	Protocol / API
C-ITS Message reception	Monitoring & Traffic Management	MEC Broker	MQTT with DENM, CAM and CPM messages
C-ITS Message dissemination	Monitoring & Traffic Management	MEC Broker	MQTT with DENM messages
Raw Sensor data upload	HD Maps	AD Vehicle	SFTP with ADTF .dat files
HD Maps JSON ITS Centre exchange	HD Maps	Monitoring & Traffic Management	SFTP with HD Maps JSON files
HD Maps JSON dissemination to vehicles	HD Maps	MEC Broker	MQTT with HD Maps JSON messages
Map DB ITS Centre Exchange	Monitoring & Traffic Management ES / PT	Monitoring & Traffic Management PT / ES (other country)	SFTP with HD Map DB files
Map DB update	Monitoring & Traffic Management	AD Vehicle	SFTP with HD Map DB files
Reception of Datex II Situations	ATMS integration module	InfraPT ATMS	Datex II
C-ITS Message dissemination	ATMS integration module	MEC Broker	MQTT with DENM messages

6.1.2. MEC applications

MQTT Brokers and Geoservers

This subsection describes the MQTT Brokers and GeoServers in the ES-PT CBC MECs.

In order to exchange messages between connected network devices, such as vehicles, roadside infrastructure, VRUs or ITS Centers, MQTT brokers are installed on each side of the ES-PT border. There is one broker in the ES MEC node and another one in the PT MEC, as shown in Figure 26.

The messages published on the brokers are organized according to the message type, so there are different topics for each type, such as CAM, DENM, CPM or MCM. Besides the MQTT broker itself, there is also a GeoServer application running in each MEC node, which is responsible for handling the messages published by all connected elements. The strategy used to distribute only the relevant information to each subscriber is to split the map in geographical tiles or quadtrees. As a result, the broker topics names are defined as follows:

<general_topic_header>/**<message_type>**/**<tile>**/<StationID>

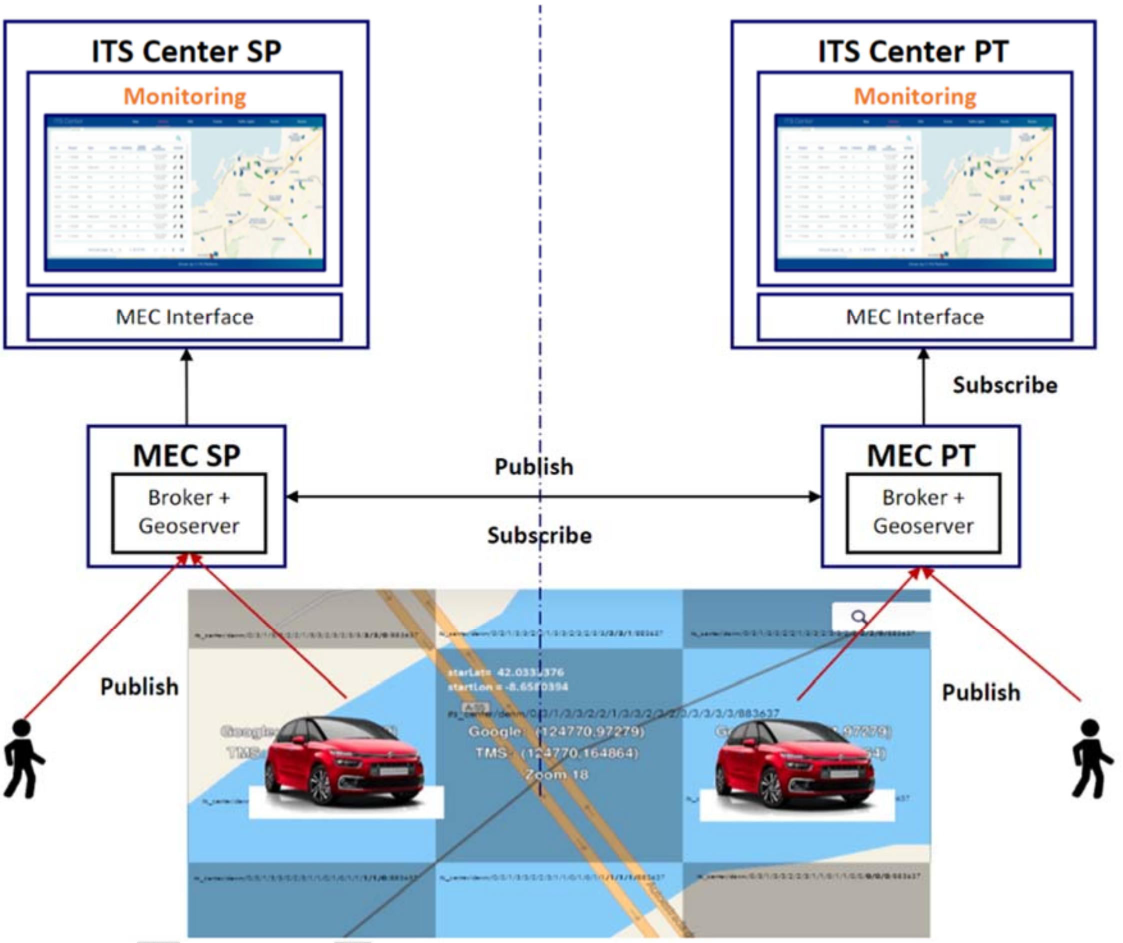


Figure 26: MEC MQTT Brokers and GeoServers in ES-PT CBC

The tiling structure is based on the Google XYZ standard[1], that uses Mercator projection and allows the calculation of the relevant geographic tiles for a specific location. Figure 27 shows the world divided into four tiles/squares on the left, corresponding to zoom level 1, and level 2 on the right, where the four initial

[1] https://developers.google.com/maps/documentation/javascript/coordinates

tiles are again divided into four smaller squares, resulting in a total of 16 tiles. This process is repeated the number of times necessary to obtain the desired zoom level.

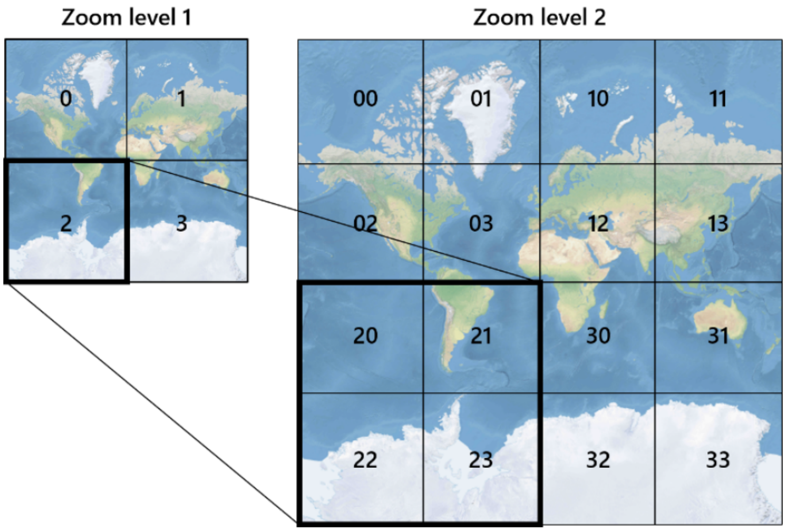


Figure 27: Tiling Structure

For interurban environments, a smaller zoom level is often more convenient, as the speed of the vehicles is usually higher. On the other hand, the zoom level increases for urban environments, where a smaller quadtree size is needed in order to receive, therefore process, a smaller number of messages.

The maximum level of zoom is 18 that corresponds to a 150 meters side tile approx. Example:

- <generic_topic_header>/cam/0/986
- <generic_topic_header>/cam/0/1/986
- …
- <generic_topic_header>/cam/0/3/1/3/3/2/2/3/1/1/0/1/1/0/0/0/0/0/986

The quadtree to publish in, will be computed by the connected device (vehicles, RSUs, …) based on the position of the cooperative alert/message that is needed to notify. Once the quadtree is obtained, the alert will be published in the corresponding topic as explained above. These messages are published in the inqueue topics of the broker, for instance:

- its_center/inqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/2/3/3/3/3/3/986

After the messages are published in the broker, they will be received and processed by the GeoServer application that subscribes to all inqueue topics. Then, the GeoServer calculates the adjacent tiles, and republish the message in all of them, in order to make it available for connected vehicles that are in the surroundings of the tile where the message was published. However, in this case the messages are published in the outqueue topics:

Same tile:

- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/2/3/3/3/3/3/986

Adjacent tiles:

- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/2/3/3/3/3/2/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/3/2/2/2/2/2/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/3/1/1/0/1/0/1/1/1/1/1/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/2/3/3/3/3/0/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/3/1/1/0/1/0/1/1/1/1/0/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/2/3/3/3/3/1/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/1/3/3/2/3/3/2/2/2/2/0/986
- its_center/outqueue/cam/0/3/1/3/3/2/2/3/1/1/0/1/1/0/0/0/0/0/986

For their part, vehicles and VRUs will dynamically subscribe to the topic that corresponds to the tile that they are currently in. In this manner, they will receive the CAM, DENM, CPM and MCM messages that are published in that tile and its adjacent ones. This option was selected instead of relying on vehicles' subscriptions to the tiles they judge of interest, because the GeoServer has a more general overview of the traffic system and thus can better manage the publishing tiles in order to fulfil specific message dissemination purposes.

To illustrate the process, Figure 28 depicts a VRU alert published in a quadtree, that will be republished by the GeoServer in its adjacent quadtrees. This alert will be received by the vehicles subscribed to any of those quadtrees.

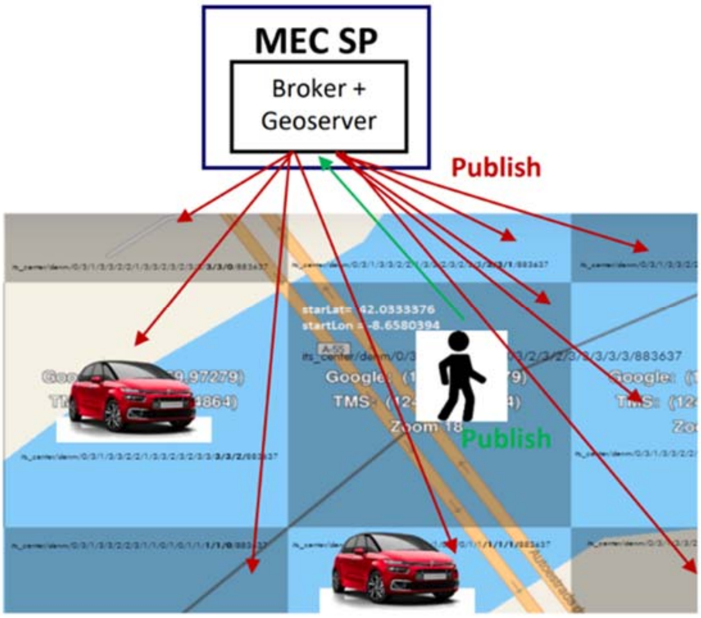


Figure 28: GeoServer republishing strategy

There is also an exchange of messages between MEC nodes, in order to provide service continuity in the cross-border areas, so that vehicles and VRUs connected to one of the MQTT brokers can also receive information published in the broker of the other country as shown in Figure 29. This interconnection is attained through the GeoServer application that subscribes to specific topics called "inter_mecs" in the other's country MQTT broker and republishes those received messages in outqueue topics of the co-located broker. The topic naming in this case is the following:

ES MEC publishing topics = PT MEC subscribing topics: its_center/inter_mecs/sp/ <message_type> /<tile>/<StationID>ES MEC subscribing topics = PT MEC

publishing topics: its_center/inter_mecs/pt/ <message_type>/<tile>/<StationID>

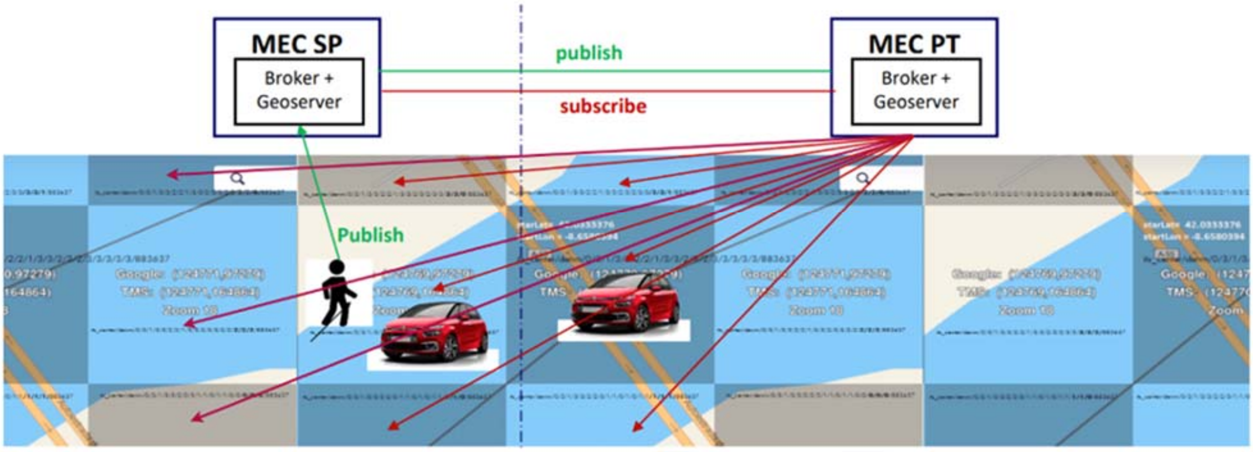


Figure 29: Interconnection between MEC brokers collision detection for connected VRU

Collision detection for VRU is an application that will be running in the MEC with the function of detecting possible collisions or dangerous situations between vehicles and Connected VRUs. This application is using CAM messages that are only shared by connected vehicles and connected VRUs (smartphone VRU app). Figure 30 shows the architecture of the Application. Figure 31 presents a schematic of the connectivity between elements that belongs to the Collision detection for Connected VRU.

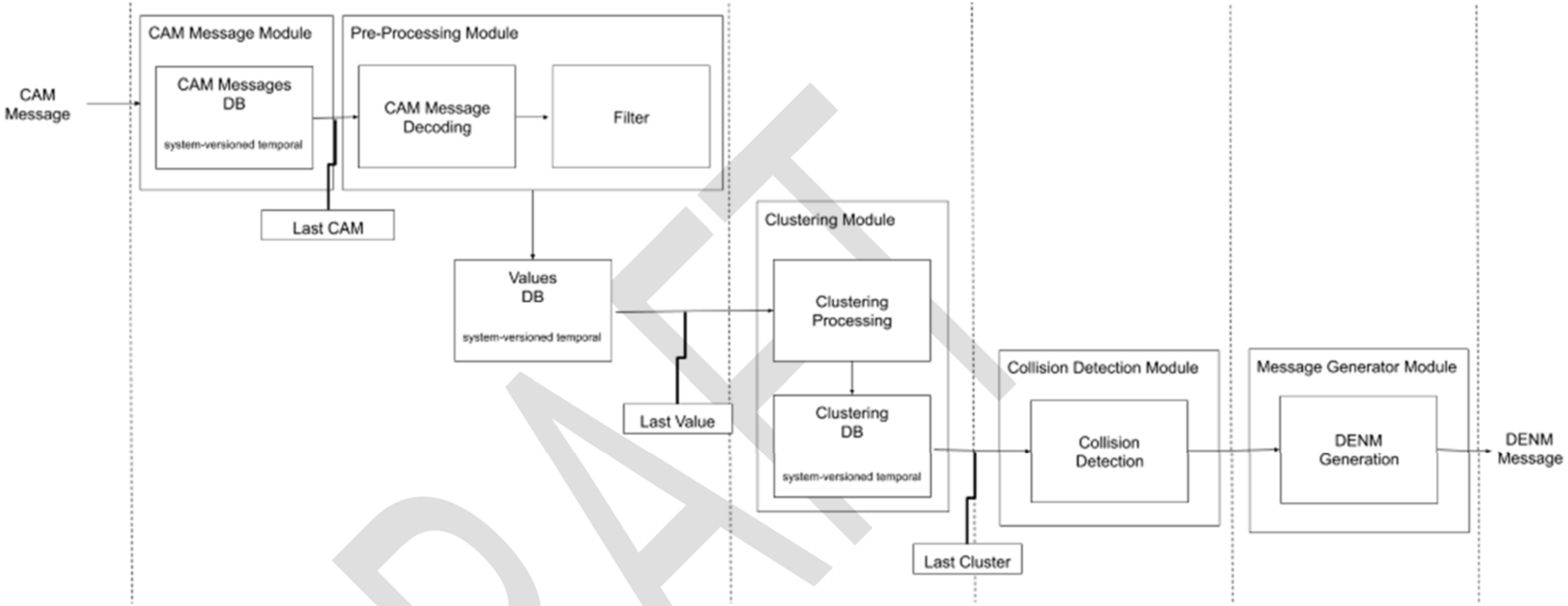


Figure 30: Architecture for Collision Detection for Connected VRU

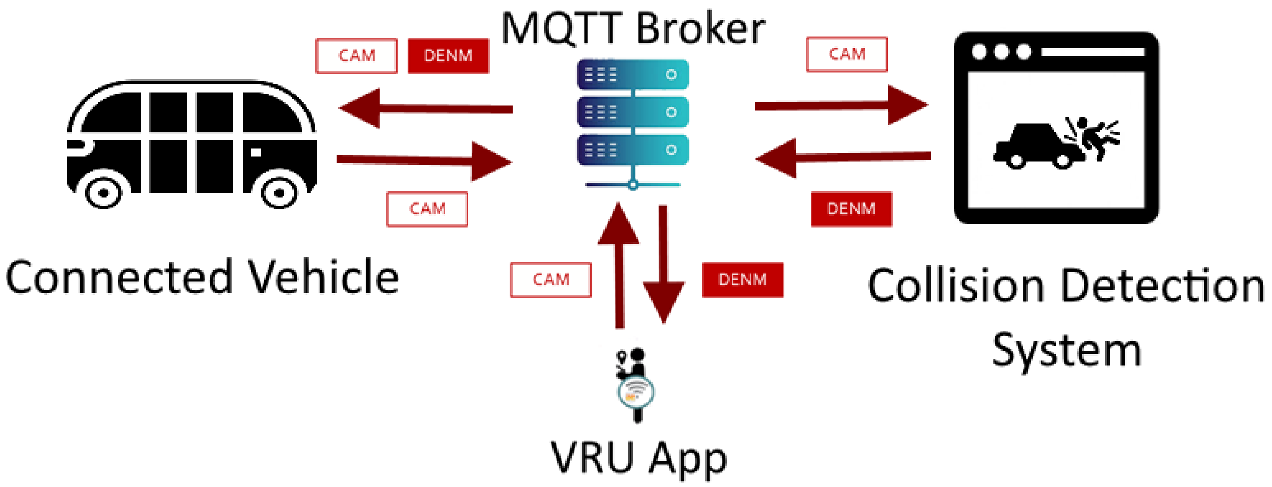


Figure 31: Schematic of Collision Detection System connectivity

The application was designed to have five modules: CAM message, Pre-Processing, Clustering, Collision Detection and Message Generator. These modules are developed considering a microservice architecture. In the CAM message module, the application will collect the CAM messages that are shared through MQTT Broker. These messages are stored in an internal Database, used for analysis and debugging purpose. In the following module, Pre-processing Module, the CAM message is decoded and stored based on the ITS station code. The most recent CAM message are linked to the register of the ITS station. If it is the first CAM for this specific ITS station, the application creates a new registry. After, the GPS position is filtered through a Kalman Filter (based on previous GPS values). The final result is saved in a database that will be available to the following modules. The database is updated based on the activity of the ITS station: if the application does not receive a CAM message in a specific interval (between 10 and 20 seconds) the ITS station is deleted from the database.

The Clustering module uses the last positions of all active ITS to create two clusters, one for vehicles and another for VRUs. This clusters are tested in the Collision detection module. To trigger a collision warning, a vehicle and a VRU must be at least at 5 meters and less than 5 seconds between them. If these two conditions are true, the modules Message Generator generate a DENM message considering the interception point. The DENM message is sent to the MQTT Broker.

The collision detection system development has been finished. The modules were initially developed as standalone components. The Collision detection module was implemented, and the algorithm and thresholds were tested using synthetic messages created for this purpose.

In the following development cycles, the microservice architecture was implemented using the modules already developed and tested. The application is then deployed in order to run in a virtual machine in the MEC. After the installation, we proceeded with the integration and validation tests of the solution using firstly synthetic data, and finally real data.

Remote Driving application

The Remote Driving Application running in the MEC is a NodeJS development with the purpose of redirecting UDP messages associated to the Remote Driving Protocol. Another additional piece of software is required to be installed in a PC which has a direct connection to the MEC instance and that is used as a cockpit in order to be able to remotely drive the vehicle and to get the telemetry data information into the HMD display. The installation instructions are described in section 10.7.1 related to the cockpit installation guide.

The following scenarios take place from an end-to-end perspective:

1.We are in the "master" MEC (the Spanish one) and we receive an UDP message from the cockpit. The message is directly sent to the vehicle in case the vehicle is attached to a cell under Spanish operator coverage (Telefonica). If this is not the case, the message is forwarded to the Portuguese MEC so that it delivers it to the Vehicle.

2.We are in the "slave" MEC (the Portuguese one) and we receive an UDP command from the other MEC (coming from the cockpit). In this case we send it right away to the vehicle.

3.We are in the "master" MEC (the Spanish one) and we receive and ACK UDP message (it could come from the other MEC or from the vehicle). The message is delivered right away to the cockpit.

4.We are in the "slave" MEC (the Portuguese one) and we receive and ACK message from the vehicle. The message is passed to the Spanish MEC so that this former MEC delivers it to the cockpit.

Explicit messages are sent from the vehicle and propagated back to the Remote Driving App running in the MEC so that the instance knows whether it should send the messages directly to the vehicle or to the other MEC. In the case of being a "slave" instance of the Remote Driving app in the MEC, no messages will arrive either from the other MEC or from the vehicle if the vehicle is connected to the other country.

There are three type of specific messages that are:

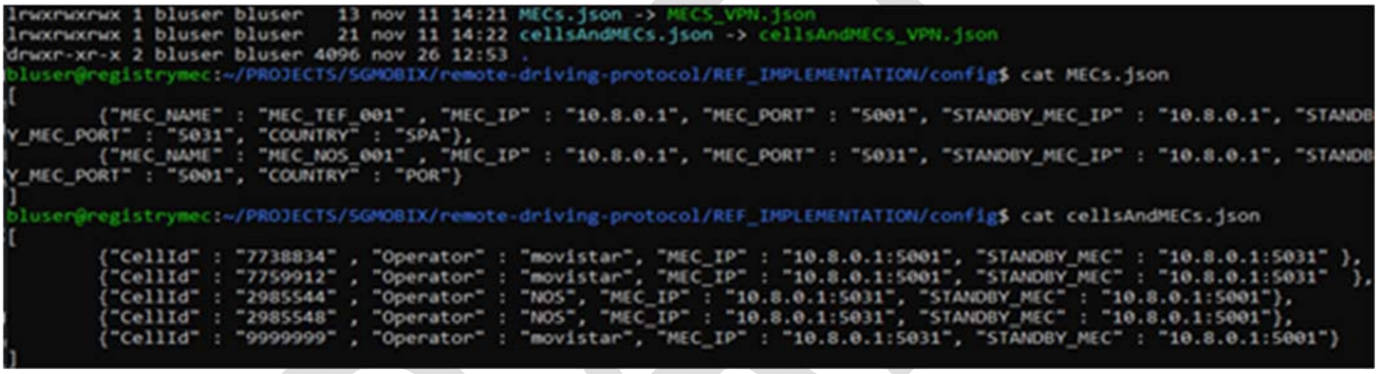
- CMD_REG_CAR: a vehicle registers to interchange messages with the MEC.
- CMD_REG_DRV: a cockpit registers to interchange messages with the MEC

- CMD_SWITCHOVER: a roaming event has occurred. If the vehicle moves from one country to the other, the behaviour of the application needs to change so that it either delivers the messages directly to the vehicle or to a slave MEC.

At start-up, the instance needs to connect to the Registry VM in order to register itself and get the information on the topology and their own IPs and ports used for listening as well as the ones for its counterpart MEC instance.

Registry application

It takes care of registering the vehicles and drivers which are active actors of the remote driving protocol. It also holds the configuration files for the system: MECS.json and cellsAndMECs.json (see Figure 32).



```
lrwxrwxrwx 1 bluser bluser   13 nov 11 14:21 MECs.json -> MECS_VPN.json
lrwxrwxrwx 1 bluser bluser   21 nov 11 14:22 cellsAndMECs.json -> cellsAndMECs_VPN.json
drwxr-xr-x 2 bluser bluser 4096 nov 26 12:53 .
bluser@registrymec:~/PROJECTS/5GMOBIX/remote-driving-protocol/REF_IMPLEMENTATION/config$ cat MECs.json
[
        {"MEC_NAME" : "MEC_TEF_001" , "MEC_IP" : "10.8.0.1", "MEC_PORT" : "5001", "STANDBY_MEC_IP" : "10.8.0.1", "STANDB
Y_MEC_PORT" : "5031", "COUNTRY" : "SPA"},
        {"MEC_NAME" : "MEC_NOS_001" , "MEC_IP" : "10.8.0.1", "MEC_PORT" : "5031", "STANDBY_MEC_IP" : "10.8.0.1", "STANDB
Y_MEC_PORT" : "5001", "COUNTRY" : "POR"}
]
bluser@registrymec:~/PROJECTS/5GMOBIX/remote-driving-protocol/REF_IMPLEMENTATION/config$ cat cellsAndMECs.json
[
        {"CellId" : "7738834" , "Operator" : "movistar", "MEC_IP" : "10.8.0.1:5001", "STANDBY_MEC" : "10.8.0.1:5031" },
        {"CellId" : "7759912" , "Operator" : "movistar", "MEC_IP" : "10.8.0.1:5001", "STANDBY_MEC" : "10.8.0.1:5031"  },
        {"CellId" : "2985544" , "Operator" : "NOS", "MEC_IP" : "10.8.0.1:5031", "STANDBY_MEC" : "10.8.0.1:5001"},
        {"CellId" : "2985548" , "Operator" : "NOS", "MEC_IP" : "10.8.0.1:5031", "STANDBY_MEC" : "10.8.0.1:5001"},
        {"CellId" : "9999999" , "Operator" : "movistar", "MEC_IP" : "10.8.0.1:5031", "STANDBY_MEC" : "10.8.0.1:5001"}
]
```

Figure 32: Registry application

It has the following REST API:

- /action/listCars: provides the client with the list of registered vehicles that can be driven.
- /action/driveCar: notify the system that a given driver (cockpit) will drive a vehicle.
- /action/lookupMEC: tells the client which is the MEC associated to a given CellID.
- /action/listMECs: lists the Remote Driving MEC instances IPs and Ports.
- /action/listCellsAndMECs: provides the client with the complete mapping of CellIDs and MEC instances.

And it also enables a socket.io registration process for the vehicles to join as well as for notifying a switchover event (roaming).

6.2. GR-TR

This section covers the implemented functionalities for user stories of the GR-TR cross-border corridor.

6.2.1. Platooning with "see-what-I-see" application

This user story is accomplished by two different application servers. For the platooning application, the Turkcell servers are used, while for the see-what-I-see application (SWISA), the ICCS servers are used. Trucks are provided by FORD and network providers are Cosmote & Turkcell (networks deployed by Ericsson GR & TR) for 5G connectivity. On-board units are provided by IMEC for both applications.

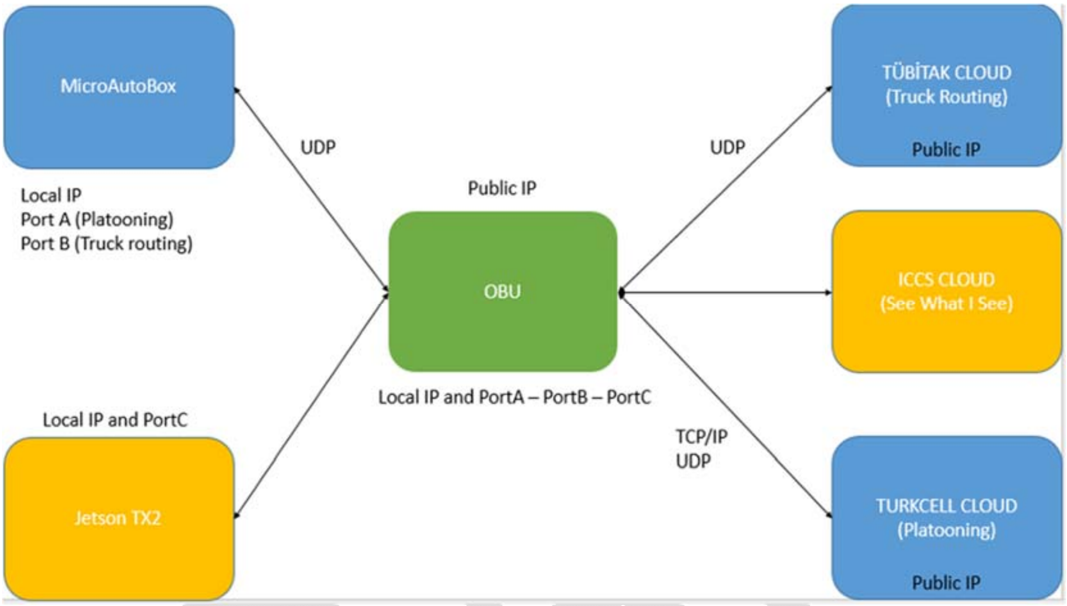


Figure 33: MABX_OBU Server Connectivity Boundary Diagram

To accomplish the platooning application, a server PC located in Turkcell – İstanbul Core is used. MQTT broker is established to accomplish message traffic between platoon leader and follower. A vehicle controller unit (MicroAutoBox), IMEC OBU and server connectivity boundary diagram can be seen in Figure 33.

Figure 34 shows the message flow between LEVIS clients and ICCS Cloud:

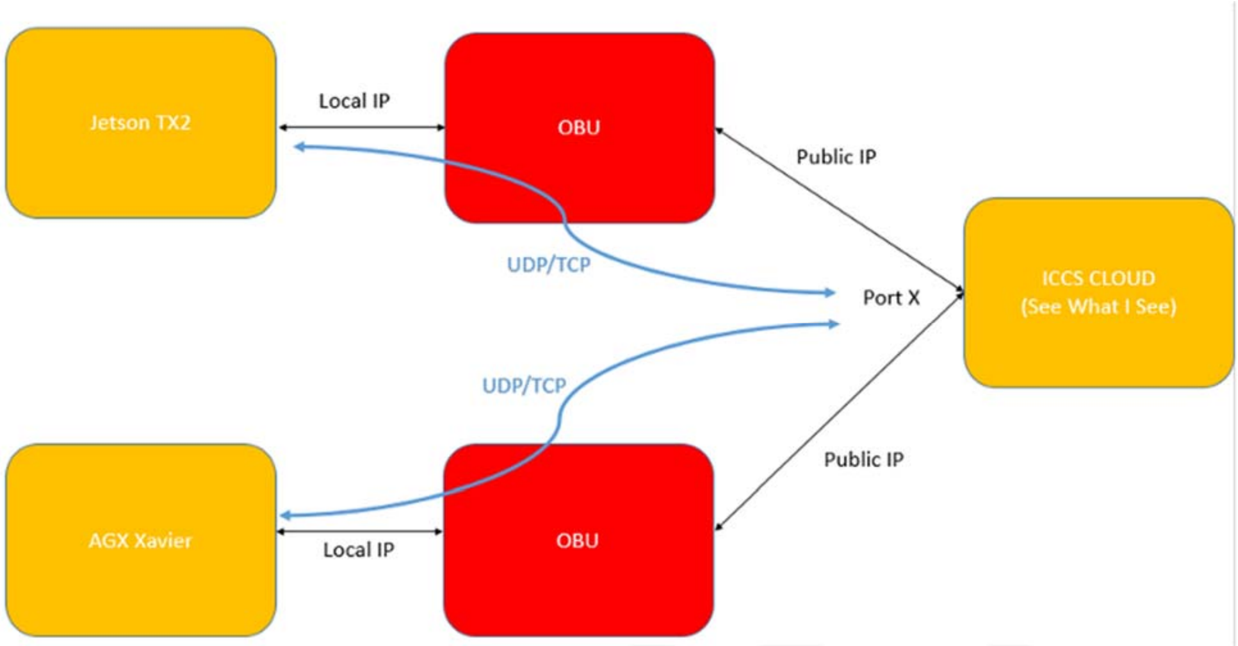


Figure 34: LEVIS Client – OBU – SWIS Server Connection Boundary Diagram.

Information flow

The information flow for Platooning with see-what-I-see application is the following:

1. The leader driver creates a platoon and his request is received by the Platooning server
2. The platooning server saves the leader vehicle IP.
3. The follower vehicle is also registered to the same platooning server. The follower vehicle wants to join a platoon and searches for it.
4. The platoon server matches the vehicles.
5. The follower vehicle approaches the leader vehicle and the follower driver initiates a join maneuver.
6. After join is completed, the platoon members share speed, acceleration, position and intention (Split, merge, dissolve etc.) with each other through the Platooning server.
7. Any time after the platoon is established, the follower vehicle driver requests video stream from the leader vehicle.
8. After request is received by the ICCS server, the RTSP server is started.
9. The ICCS server request live a stream from the leader vehicle LEVIS client.

10. The leader vehicle LEVIS client encodes video stream and sends it back to the ICCS server.

11. The ICCS server sends received encoded video streams to the follower vehicle LEVIS client.

12. The LEVIS client on the follower vehicle decodes the video stream and shows it on the HMI.

All communication flow is accomplished with IMEC OBUs and gNBs.

User Story UML Diagrams

The UML (Unified Modelling Language) sequence diagram of the platooning is presented in Figure 35:

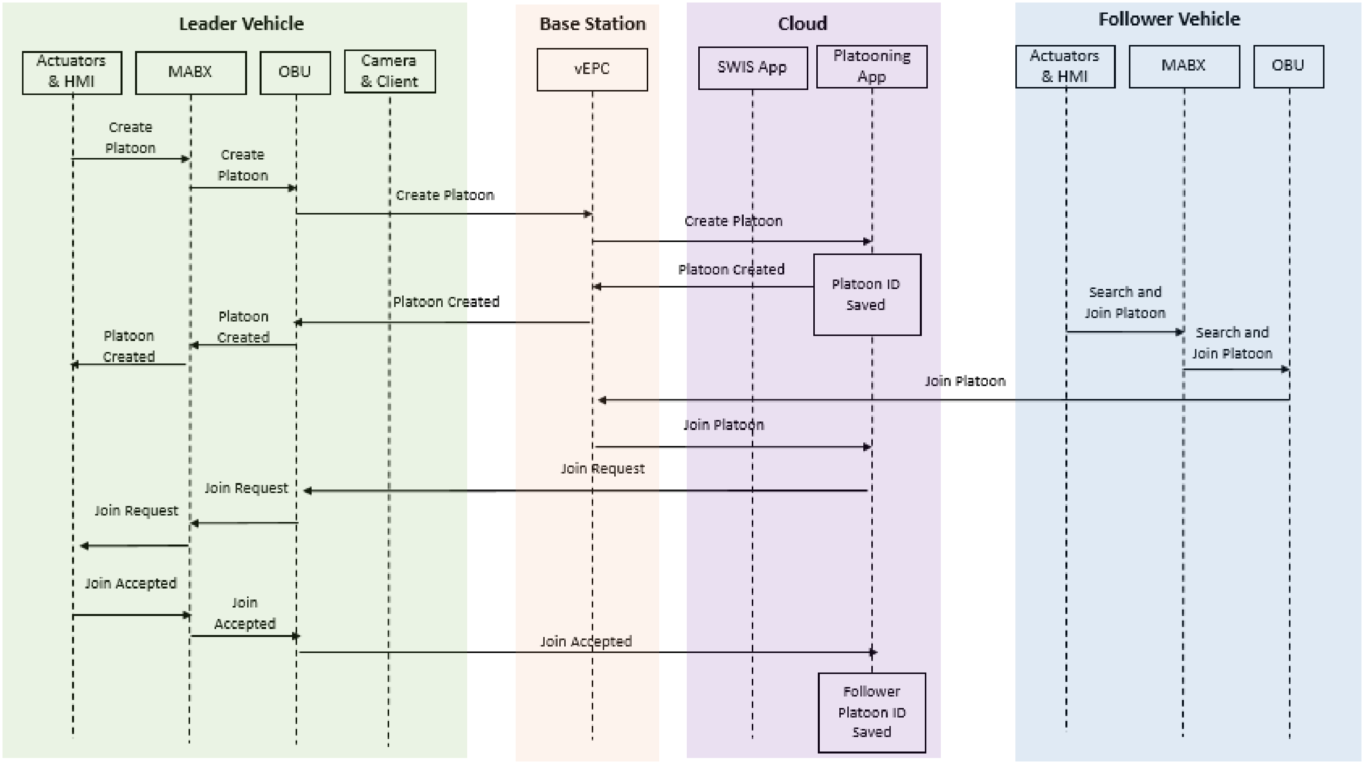


Figure 35: Platooning UML Sequence Diagram

The UML sequence diagram of the SWISA, is presented in Figure 36:

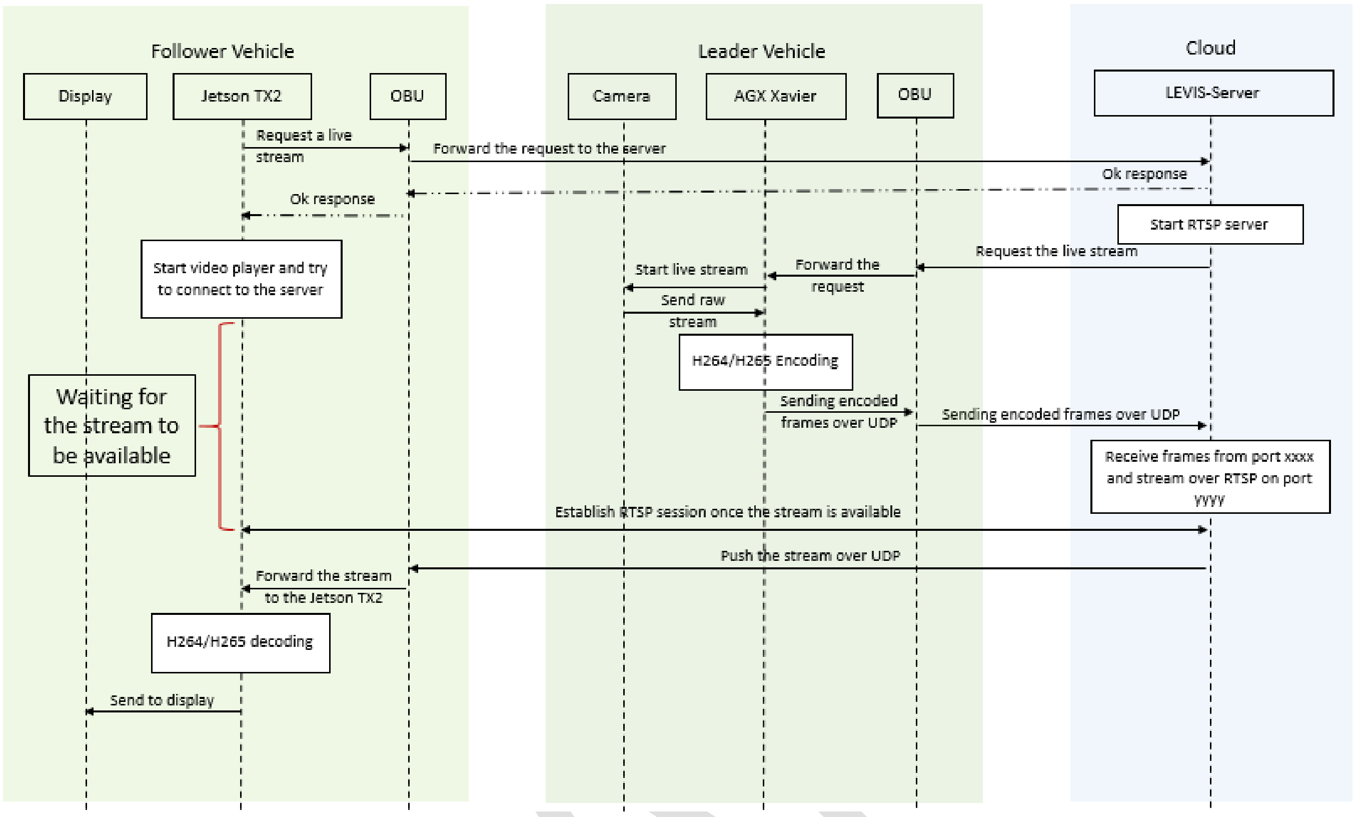

Follower Vehicle
Leader Vehicle
Cloud
Display
Jetson TX2
OBU
Camera
AGX Xavier
OBU
LEVIS-Server
Request a live stream
Forward the request to the server
Ok response
Ok response
Start RTSP server
Request the live stream
Forward the request
Start live stream
Send raw stream
Start video player and try to connect to the server
H264/H265 Encoding
Waiting for the stream to be available
Sending encoded frames over UDP
Sending encoded frames over UDP
Receive frames from port xxxx and stream over RTSP on port yyyy
Establish RTSP session once the stream is available
Push the stream over UDP
Forward the stream to the Jetson TX2
H264/H265 decoding
Send to display

Figure 36: SWISA UML Sequence Diagram

6.2.2. Assisted "zero-touch" border crossing application

The "Zero-touch border-crossing" user story driven by WINGS is deployed at the GR-TR CBC and is realized using the Cosmote & Turkcell networks (deployed by Ericsson GR & TR) for 5G connectivity and one Ford truck for the trials, which will be enriched with the OBU and sensors developed by WINGS. Besides the 5G network infrastructure and the FORD truck, the following entities are required for the realization of this user story:

- The WINGS analytics platform running either on the Cloud or in the MEC server
- The WINGS OBU and its integrated sensors transmitting information to the WINGS platform (described in deliverable D3.2 (5G-MOBIX, Deliverable D3.2: Vehicle adaptation for CCAM use cases, 2020))
- The WINGS RSU and the integrated sensors, installed at the customs site, transmitting information to the WINGS platform
- Three laptops/tablets/smartphones to act clients and to display the WINGS developed GUIs, one customs agency / officer and one in the truck addressing the driver. Both devices are connected and receive information from the WINGS platform.

The end-to-end high-level architecture of the "Zero-touch border-crossing" user story led by WINGS is depicted in Figure 37.

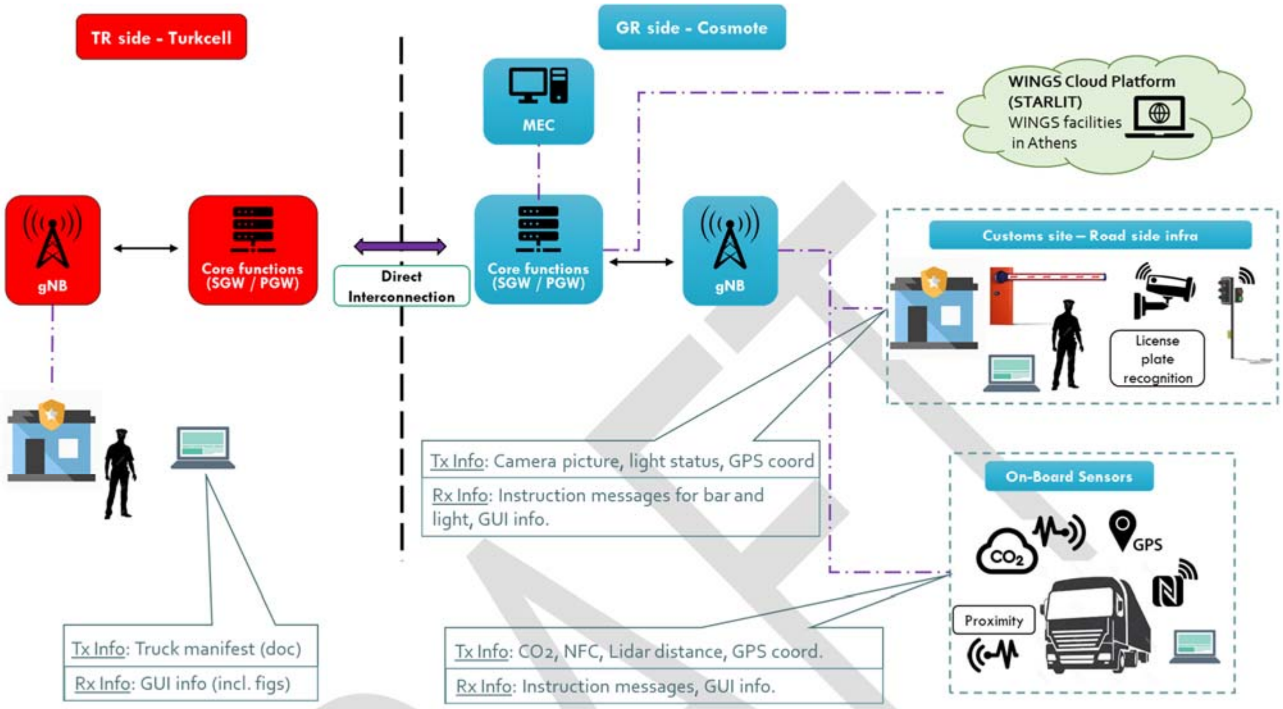

TR side - Turkcell
GR side - Cosmote
MEC
WINGS Cloud Platform (STARLIT)
WINGS facilities in Athens
gNB
Core functions (SGW / PGW)
Direct Interconnection
Core functions (SGW / PGW)
gNB
Customs site – Road side infra
License plate recognition
Tx Info: Camera picture, light status, GPS coord
Rx Info: Instruction messages for bar and light, GUI info.
On-Board Sensors
CO2
GPS
Proximity
Tx Info: Truck manifest (doc)
Rx Info: GUI info (incl. figs)
Tx Info: CO2, NFC, Lidar distance, GPS coord.
Rx Info: Instruction messages, GUI info.

Figure 37: WINGS "zero touch border crossing" architecture

Information flow

Depending on whether the WINGS platforms functionality resides in the remote Cloud located far away from the trial location (i.e., WINGS premises in Athens) or in the MEC/edge close to the trial location, the different information flows to and from the truck / road-side sensors and the WINGS platform need to be established over the 5G network. For the proper functionality of the "Assisted zero touch border crossing" the following flows are necessary:

- CO2 measurements from OBU to WINGS platform (mMTC service)
- NFC readings from OBU to WINGS platform (mMTC service)
- GPS readings from OBU to WINGS platform (mMTC service)
- Proximity measurements from LiDAR / OBU to WINGS platform (uRLLC service)
  - Need for quick reaction to the detection of a person in front of the moving truck
- Still-frames (pictures) from road-side camera to WINGS platform (semi-eMBB service)
- Instructions from WINGS platform to smart border bar and smart traffic light (mMTC service)
- GUI information (including maps & license plate picture) from WINGS platform to the two customs agencies GUIs (semi-eMBB service)

- GUI information (including maps & ECU info) from WINGS platform to the driver GUI (semi-eMBB service)

Based on the above identified transmitted and received information from each entity of the user story, information flows can be identified depending on the location of the truck at each side of the borders and the location of the WINGS platform functionality (Cloud or MEC).

Cloud Based functionality (Scenario A)

Figure 38 depicts the required (double-sided) information flows for the case that the WINGS platform functionality resides in the Cloud and the truck (with the OBU) is transmitting from the Home-PLMN (H-PLMN). In this case the WINGS platform should be accessible via a public IP, hence internet access is required via the Home-PGW.

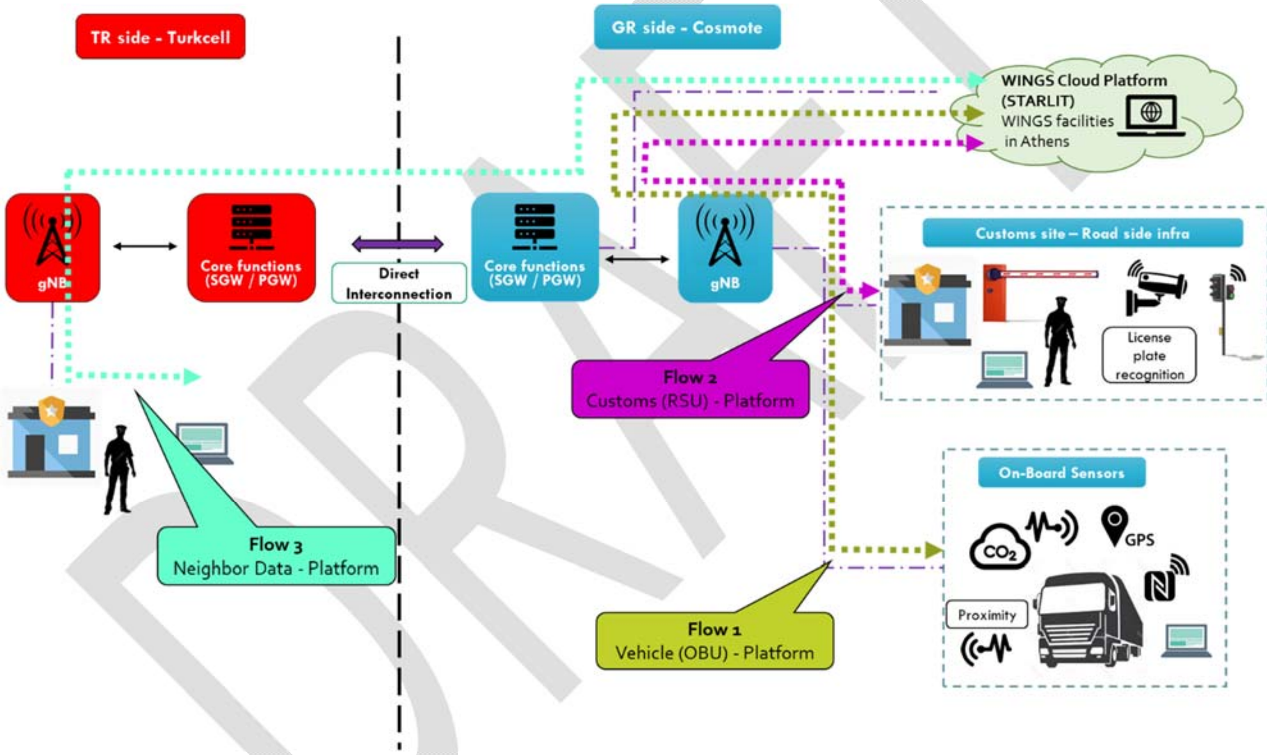


Figure 38: Information flows for the case of Cloud based functionality with the truck on the home network

Figure 39 depicts the required information flows for the case that the WINGS platform functionality still resides in a publicly accessible cloud, but the truck is now transmitting from the Visiting-PLMN (V-PLMN). The direct interconnection between the Cosmote and Turkcell 5G-NSA networks should be utilized to get the data to the WINGS platform residing in the H-PLMN, with a minimal latency.

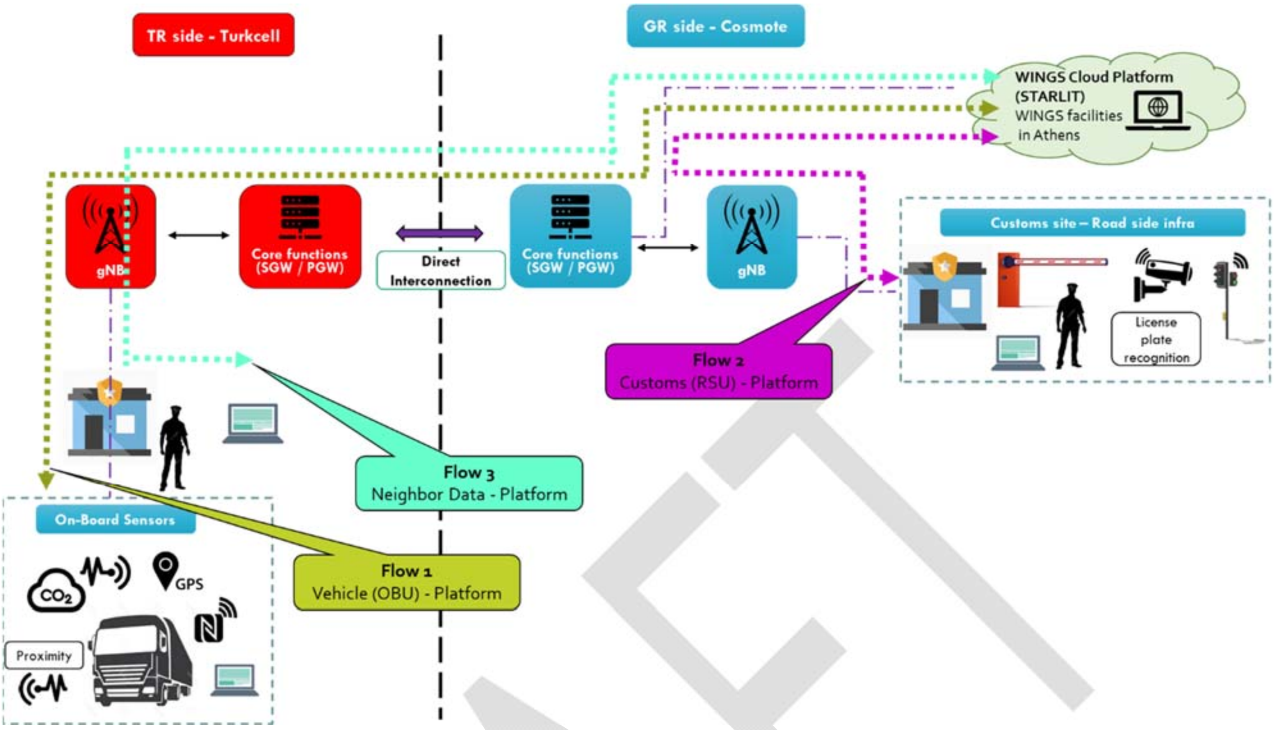

TR side - Turkcell
GR side - Cosmote
WINGS Cloud Platform (STARLIT) WINGS facilities in Athens
gNB
Core functions (SGW / PGW)
Direct Interconnection
Core functions (SGW / PGW)
gNB
Customs site – Road side infra
License plate recognition
Flow 2 Customs (RSU) - Platform
Flow 3 Neighbor Data - Platform
On-Board Sensors
CO2
GPS
Proximity
Flow 1 Vehicle (OBU) - Platform

Figure 39: Information flows for the case of Cloud based functionality with the truck on the visiting network

MEC / Edge Based functionality (Scenario B)

Figure 40 depicts the required (double-sided) information flows for the case that the WINGS platform functionality resides in the MEC/Edge and the truck (with the OBU) is transmitting from the H-PLMN. In this case the data can be directly forwarded to the co-located MEC, and all generated instructions can use the same way back.

Figure 41 depicts the required information flows for the case that the WINGS platform functionality still resides in the co-located MEC / Edge, but the truck is now transmitting from the Visiting-PLMN (V-PLMN). The direct interconnection between the Cosmote and Turkcell 5G-NSA networks should be utilized to get the data directly to the WINGS platform residing in the H-PLMN MEC, hence significantly reducing latency. Public internet access is not required in this case either.

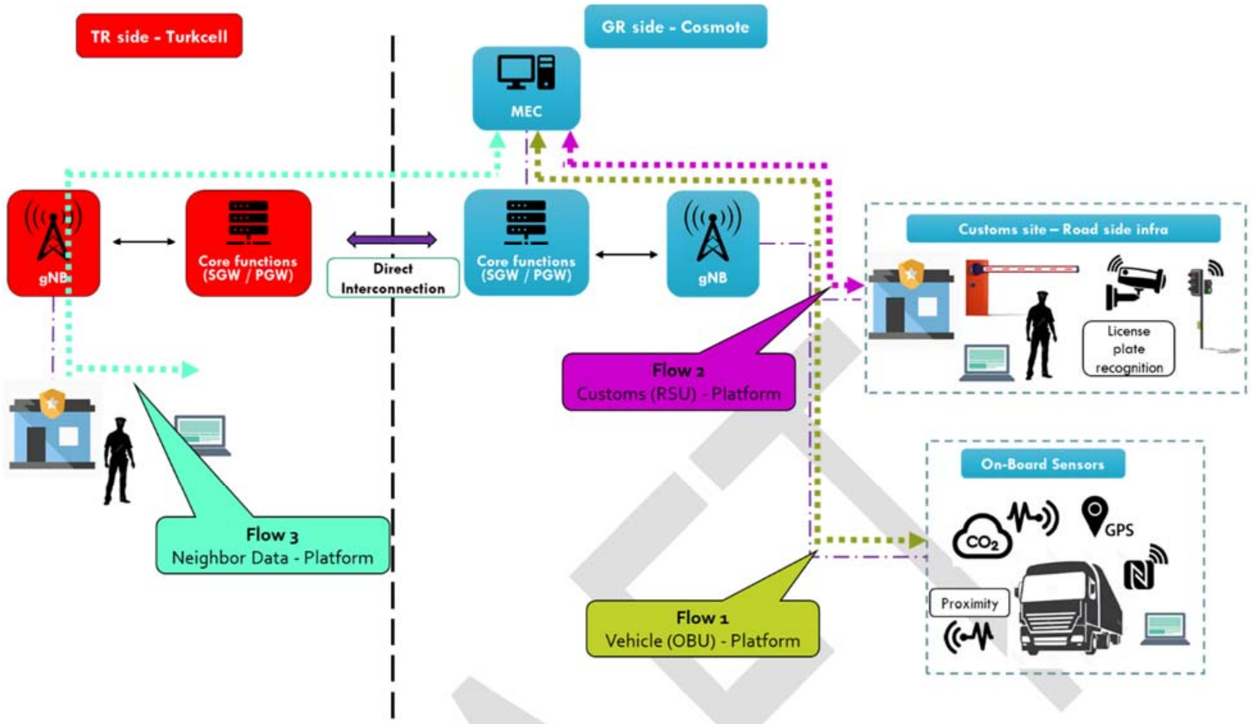


Figure 40: Information flows for the case of MEC/Edge based functionality with the truck on the home network

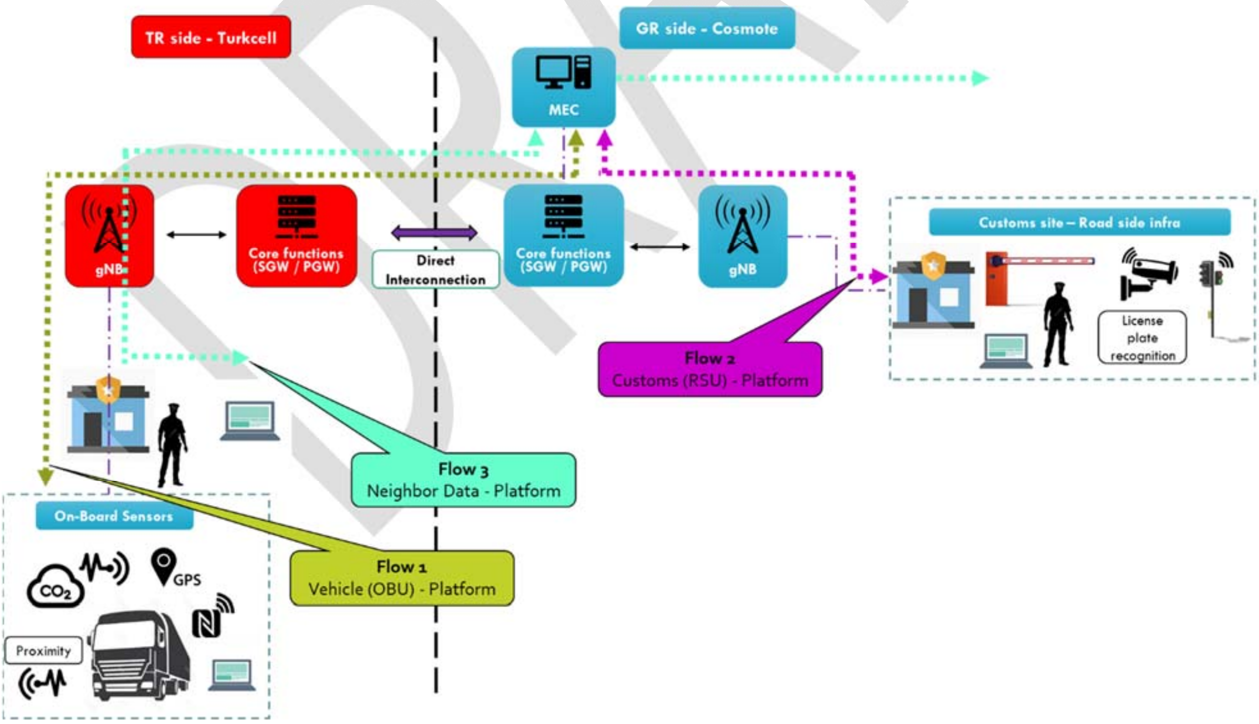


Figure 41: Information flows for the case of MEC/Edge based functionality with the truck on the visiting network

Platform functionality during HO in LBO mode

During the Local Break-Out (LBO) scenario a service interruption is expected to take place during the execution of an inter-PLMN HO, i.e., when the Ford truck is crossing the border, the truck's OBU will lose its connectivity to the original 5G network and will attach to the visiting network after some time. This scenario causes the following main effects:

a) Connectivity among the OBU and the WINGS platform will be interrupted for a certain period of time. During that period, the OBU will not be able to upload information to the WINGS server or receive information and driving instructions from the server.

b) Once connectivity is established with the V-PLMN the OBU will obtain a new IP address

c) Once connectivity is established with the V-PLMN, the data routing between the OBU and the WINGS server will have changed, as the truck will now be served by the instance of the server residing in the V-PLMN's edge. This also means that there will be a change in the IP of the serving WINGS server (from the perspective of the OBU)

d) Once the truck is attached to the V-PLMN, the necessary information (status transfer, measurement history, etc.) from the WINGS server residing at the H-PLMN edge will have to be transferred to the V-PLMN instance of the WINGS server.

In order to deal with the above effects and to avoid the shutdown or malfunction of the WINGS application, the measures shown in Table 9 have been taken and the relevant mechanisms have been implemented in the WINGS application server and the WINGS OBU.

Table 9: Measures in support of LBO on the WINGS Application and OBU

WINGS mechanisms to cope with LBO HO		Deals with Effect
WINGS Platform / Server	Imminent HO detection mechanism added, based on i) truck GPS signal and ii) mean distance between the GPS coordinates of the incoming truck and customs' agent.	a), b), c), d)
	Upon the detection of an imminent HO the WINGS platform notifies the OBU that after loss of connectivity, the OBU will obtain a new IP address from the IP address pool X.	b)
	Upon the detection of an imminent HO & Edge server location, the WINGS platform notifies the OBU that once it obtains a new IP address the WINGS server will be accessible via the new IP address Y.	c)
	Upon loss of connectivity with the OBU & after an imminent HO detection, the H-PLMN initiates transfer of the necessary information to the V-PLMN instances using the OBU SIM IMEI, to identify the truck irrespective of its IP.	d)

	Once the OBU has registered to the V-PLMN instance using its Unique OBU ID, the V-PLMN instance of the platform will re-associate this Unique OBU ID with the new V-PLMN IP address of the truck.	d)
	As real-time information on the custom agent's location will not be available for the time of connectivity interruption (HO), the WINGS H-PLMN server instance notifies the OBU to: a) Slow down the truck to no more than 20 km/h for the period of no connectivity b) Trigger autonomous braking if necessary, only based on the on-board LiDAR/proximity sensor readings, in order to maintain a minimum level of VRU protection. Upon connection establishment with the V-PLMN, normal operation resumes.	a)
WINGS OBU	Once the OBU has obtained a new IP address at the V-PLMN, it will contact the WINGS server at the V-PLMN and will register using the IMEI.	b), d)
	Upon an imminent HO detection and a loss of connectivity (during HO) the OBU starts storing all measurements (buffering) and will transmit all of them to the V-PLMN server instance upon reconnection and registration.	a), c)
	The OBU switches to "isolated" mode, for the duration of the connectivity interruption. During "isolated" mode, the OBU itself processes the reading of the LiDAR/proximity sensor and issues local commands to the Ford truck (if necessary) to brake, upon detection of an obstacle. This will maintain a minimum level of VRU protection, but with less accuracy.	a)

User Story UML Diagrams

To better reflect the information flow among the different nodes participating in this use case, and to clarify the functionality of each of the nodes/equipment in the use case, a few Unified Modelling Language (UML) diagrams have been created from different perspectives. Figure 42 represents the Use case & inter-PLMN oriented UML sequence diagram while Figure 43 represents the Components/sensors-oriented sequence UML diagram.

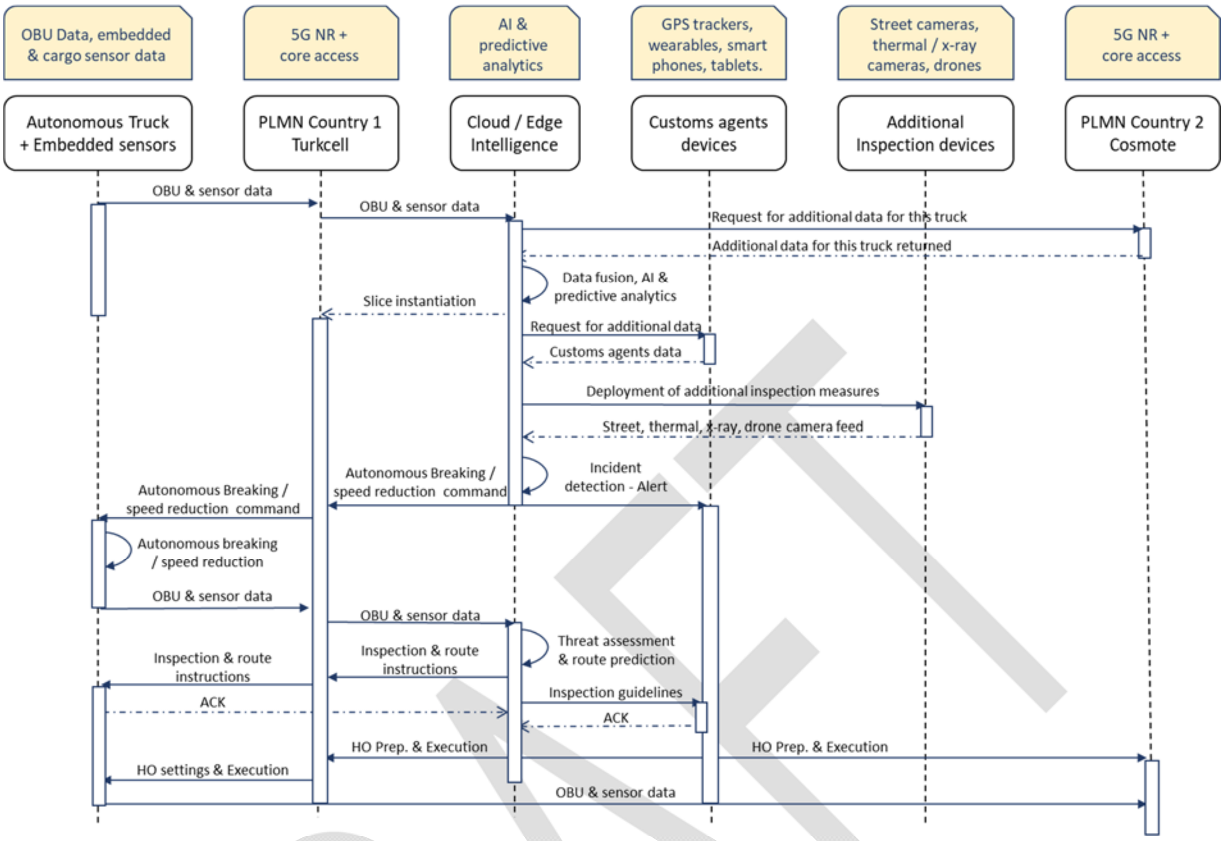

OBU Data, embedded & cargo sensor data
5G NR + core access
AI & predictive analytics
GPS trackers, wearables, smart phones, tablets.
Street cameras, thermal / x-ray cameras, drones
5G NR + core access
Autonomous Truck + Embedded sensors
PLMN Country 1 Turkcell
Cloud / Edge Intelligence
Customs agents devices
Additional Inspection devices
PLMN Country 2 Cosmote
OBU & sensor data
OBU & sensor data
Request for additional data for this truck
Additional data for this truck returned
Data fusion, AI & predictive analytics
Slice instantiation
Request for additional data
Customs agents data
Deployment of additional inspection measures
Street, thermal, x-ray, drone camera feed
Autonomous Breaking / speed reduction command
Autonomous Breaking / speed reduction command
Incident detection - Alert
Autonomous breaking / speed reduction
OBU & sensor data
OBU & sensor data
Threat assessment & route prediction
Inspection & route instructions
Inspection & route instructions
Inspection guidelines
ACK
ACK
HO Prep. & Execution
HO Prep. & Execution
HO settings & Execution
OBU & sensor data

Figure 42: Assisted "zero-touch" border-crossing – Use case & inter-PLMN oriented UML Diagram

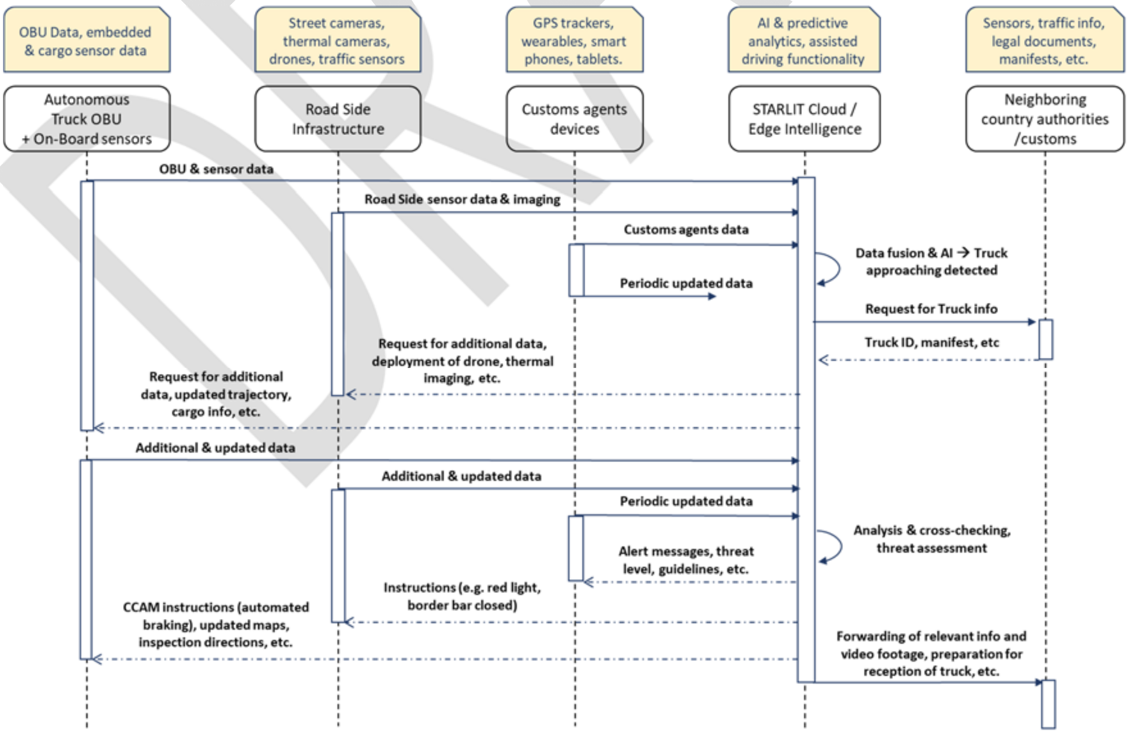

OBU Data, embedded & cargo sensor data
Street cameras, thermal cameras, drones, traffic sensors
GPS trackers, wearables, smart phones, tablets.
AI & predictive analytics, assisted driving functionality
Sensors, traffic info, legal documents, manifests, etc.
Autonomous Truck OBU + On-Board sensors
Road Side Infrastructure
Customs agents devices
STARLIT Cloud / Edge Intelligence
Neighboring country authorities /customs
OBU & sensor data
Road Side sensor data & imaging
Customs agents data
Data fusion & AI → Truck approaching detected
Periodic updated data
Request for Truck info
Truck ID, manifest, etc
Request for additional data, deployment of drone, thermal imaging, etc.
Request for additional data, updated trajectory, cargo info, etc.
Additional & updated data
Additional & updated data
Periodic updated data
Analysis & cross-checking, threat assessment
Alert messages, threat level, guidelines, etc.
Instructions (e.g. red light, border bar closed)
CCAM instructions (automated braking), updated maps, inspection directions, etc.
Forwarding of relevant info and video footage, preparation for reception of truck, etc.

Figure 43: Assisted "zero-touch" border-crossing – Components/sensors-oriented UML diagram

Assisted "zero-touch" border crossing application details

The WINGS platform running the "Assisted zero-touch border crossing" functionality can be deployed on a laptop, a server or in the Cloud. The main functionality implemented for this User Story is presented in Section 6.2.2, while the platform HW/SW requirements and the characteristics of the accompanying application are presented here. Independently of the location, the following system specifications are required to guarantee the SW smooth functionality of the Assisted "zero-touch" border-crossing platform:

- Processor: Intel Core i5-7300HQ - 2.50 GHz
- Hard Disk: 256 GB SSD
- RAM: 16GB
- Wireless Type: 802.11ac
- USB 3.0 Ports
- Operating System: Windows 10 (additionally Linux Ubuntu 18.04)

WINGS platform is running on a Java server with the following specifications:

- Java: Version 1.8.0_211
- Java (TM) SE Runtime Environment (build 1.8.0_211-b12)

Camera

A Raspberry Pi camera module V2 was selected. The sensor has a native resolution of 8 megapixel and has a fixed focus lens on board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

Login & Registration Form

A login and a registration form are included as shown in Figure 44. According to the user's input the appropriate dashboard is shown. If wrong credentials have been entered, a message pops up and prompts the user to re-enter them. There is also a link to a registration form in case the user hasn't created an account yet. In the registration form the user has the option to choose his role to either "Driver" or "Custom". Access is granted to the platform only after verification of the new user's credentials from the platform administrators. This mechanism introduces an additional layer of security to the WINGS platform, as only authorized users will be allowed to use the platform and will hence have access to the information provided therein.

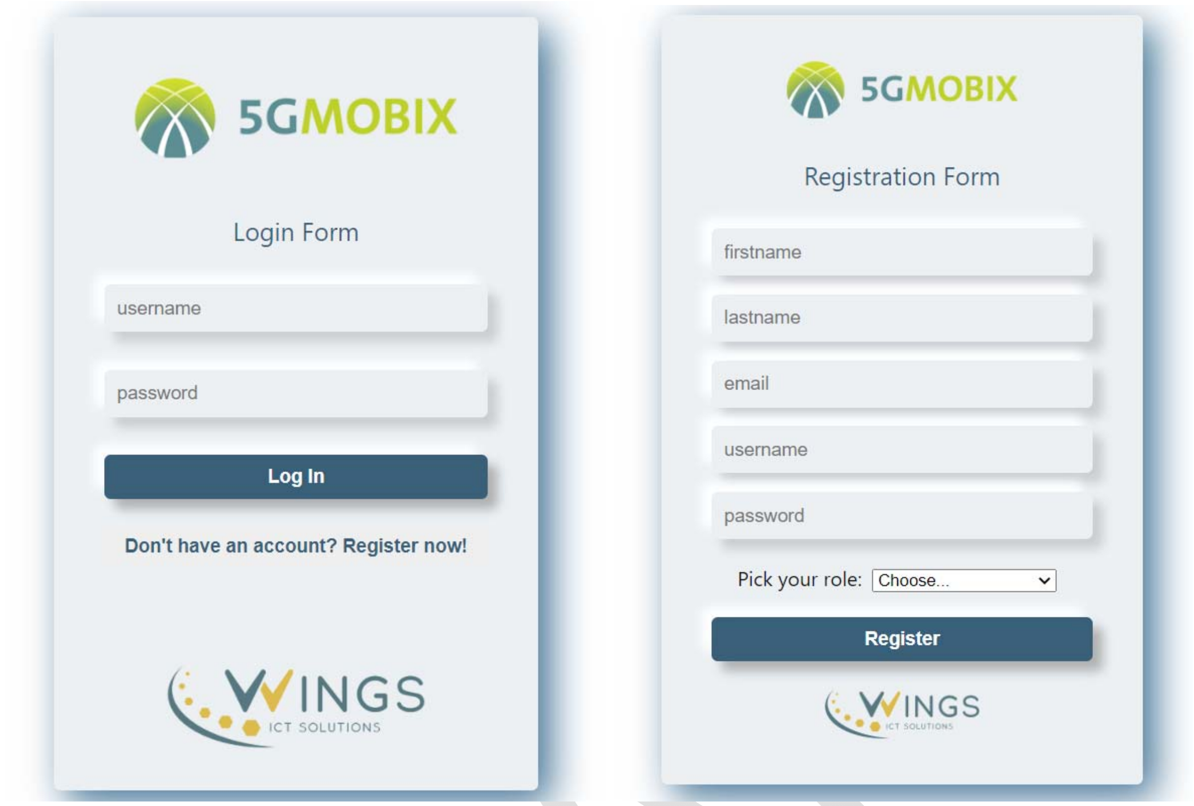
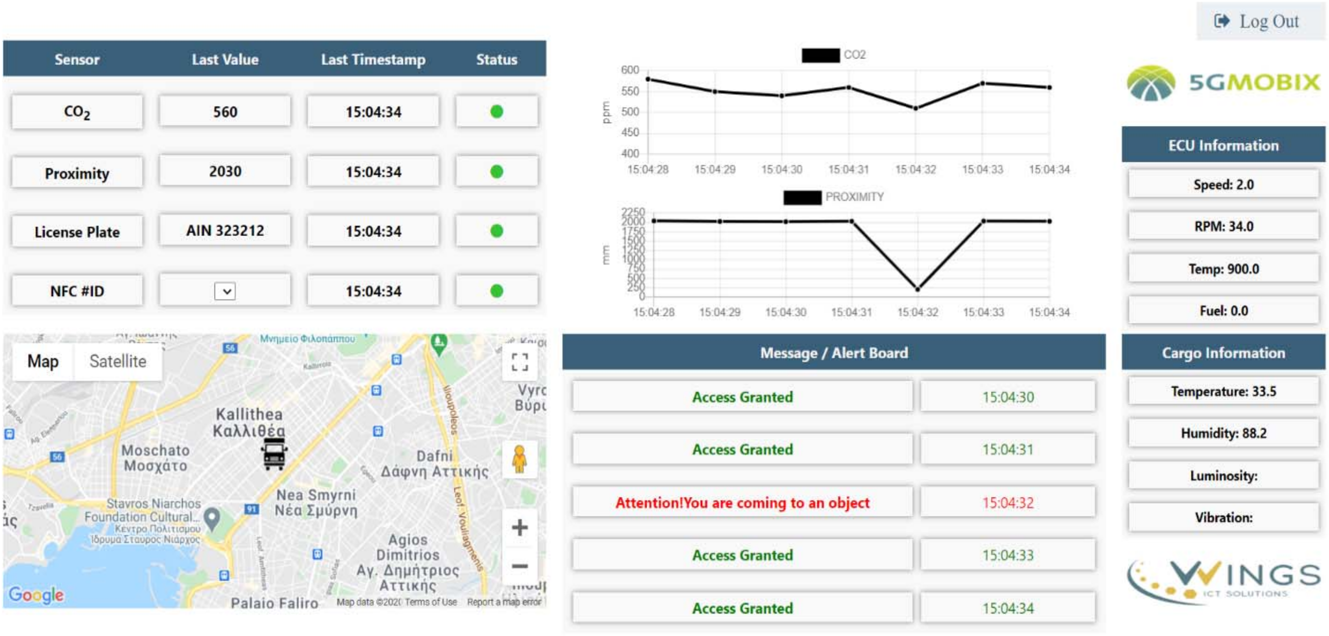


Figure 44: Assisted "zero-touch" border-crossing application – Login screen

Driver GUI

The driver's GUI displays useful information and sensors metrics for the driver (see Figure 45). An HTTP GET request is sent to the GUI's endpoint, that has been described before in order to get all information regarding the incoming trucks. The driver's GUI displays useful information and sensors metrics for the driver, in more details it is consisted of:

- A logout button.
- A table of sensors data: Co2, Proximity, NFCs.
- Charts with sensors values: Co2 and proximity values are represented on charts.
- A table of ECU information: Speed, RPM, Temperature and fuel.
- A table of Cargo Information: Temperature, Humidity, Luminosity and Vibration.
- An interactive map with truck live coordinates.
- Message board: Messages and warnings based on decision algorithms are displayed.



Log Out

Sensor	Last Value	Last Timestamp	Status
CO_2	560	15:04:34	●
Proximity	2030	15:04:34	●
License Plate	AIN 323212	15:04:34	●
NFC #ID	⌄	15:04:34	●

ECU Information
Speed: 2.0
RPM: 34.0
Temp: 900.0
Fuel: 0.0

Message / Alert Board	
Access Granted	15:04:30
Access Granted	15:04:31
Attention!You are coming to an object	15:04:32
Access Granted	15:04:33
Access Granted	15:04:34

Cargo Information
Temperature: 33.5
Humidity: 88.2
Luminosity:
Vibration:

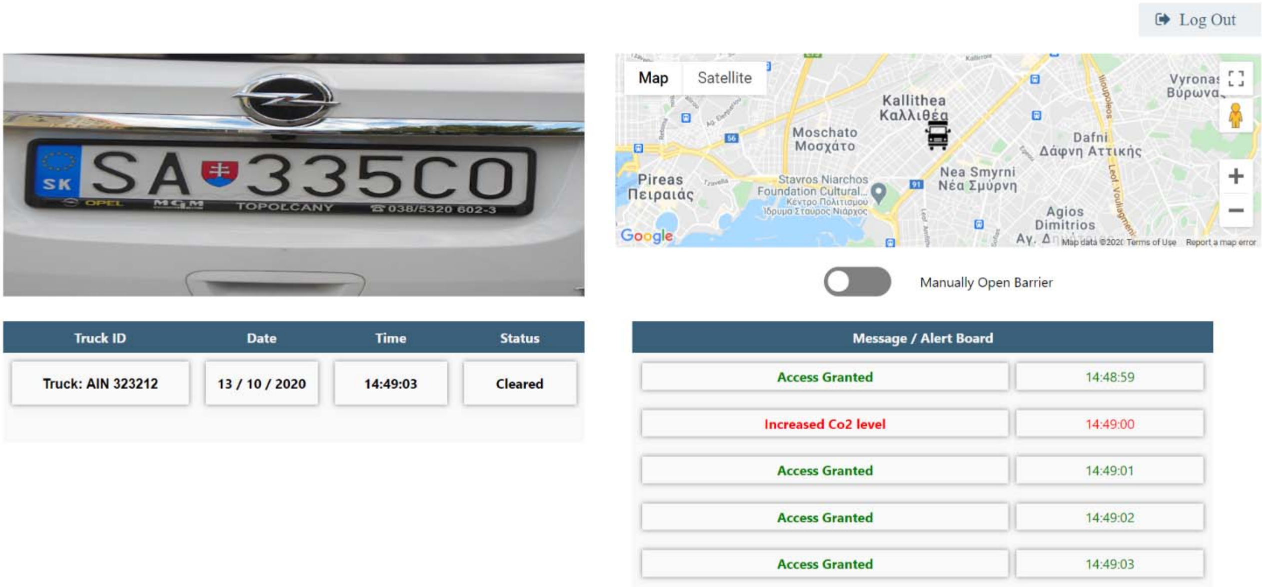
Figure 45: Assisted "zero-touch" border-crossing application - Driver GUI

Customs GUI

The Customs' GUI displays useful information and sensors metrics for the Customs agents. An HTTP GET request is being performed to GUIs endpoint that has been described before in order to get all information regarding the incoming trucks. A logout button is also displayed.

On the first screen, (see Figure 46) a table of the trucks that are approaching the borders is displayed. In a second screen (see Figure 47), a customs agent may drill down to a specific truck in order to see detailed information such as:

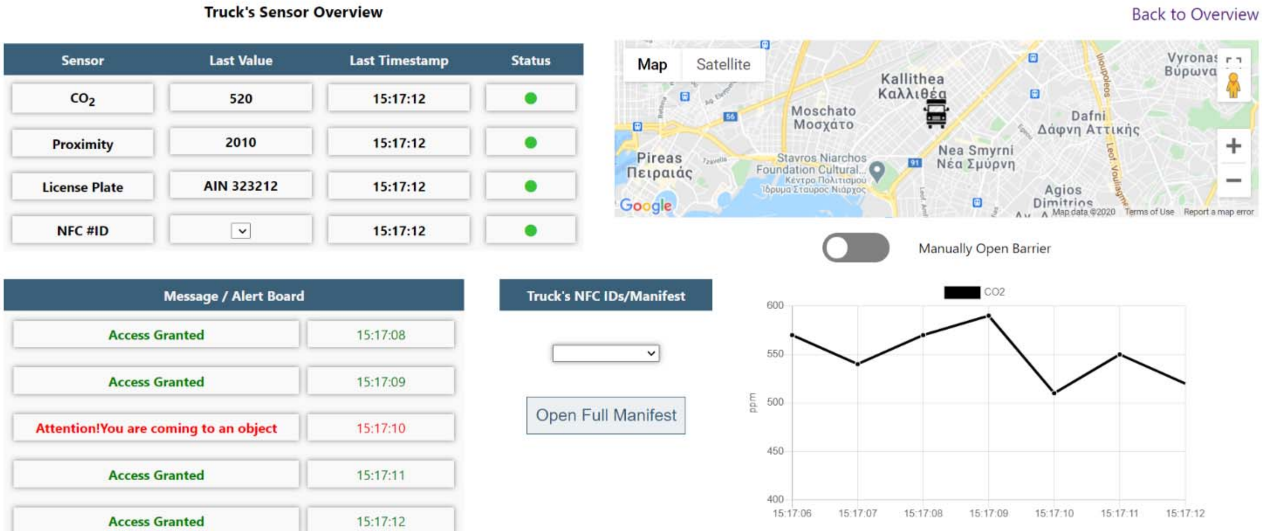
- A table of sensors data: Co2, Proximity, NFCs.
- Charts with sensors values: Co2 and proximity values are represented on charts.
- A video streaming.
- An interactive map with truck live coordinates.
- Message board: Messages and warnings based on decision algorithms are displayed.
- A manifest panel.
- A dropdown list with all the scanned NFCs.



Truck ID	Date	Time	Status
Truck: AIN 323212	13 / 10 / 2020	14:49:03	Cleared

Message / Alert Board	
Access Granted	14:48:59
Increased Co2 level	14:49:00
Access Granted	14:49:01
Access Granted	14:49:02
Access Granted	14:49:03

Figure 46: Assisted “zero-touch” border-crossing application – Customs agent GUI – View 1



Truck's Sensor Overview

Sensor	Last Value	Last Timestamp	Status
CO_2	520	15:17:12	●
Proximity	2010	15:17:12	●
License Plate	AIN 323212	15:17:12	●
NFC #ID		15:17:12	●

Message / Alert Board	
Access Granted	15:17:08
Access Granted	15:17:09
Attention!You are coming to an object	15:17:10
Access Granted	15:17:11
Access Granted	15:17:12

Truck's NFC IDs/Manifest

Figure 47: Assisted “zero-touch” border-crossing application – Customs agent GUI – View 2

6.2.3. Truck routing in customs site application

The Truck Routing user story is realized at TR customs by using the Truck Routing Application, Ford truck, IMEC OBU&RSUs and Turkcell Network. The Truck Routing Application has been developed by TUBITAK BILGEM and deployed to the TUBITAK Cloud located at TUBITAK headquarters (Kocaeli / Turkey). The Ford truck drives autonomously from the entrance of the Customs to the X-Ray building using the OBU and three RSUs provided by IMEC. Each RSU includes a LiDAR sensor and the Truck Routing Application uses this sensor information in order to route the Ford truck over cloud via 5G. Figure 48 depicts the general architecture and components of the Truck Routing user story.

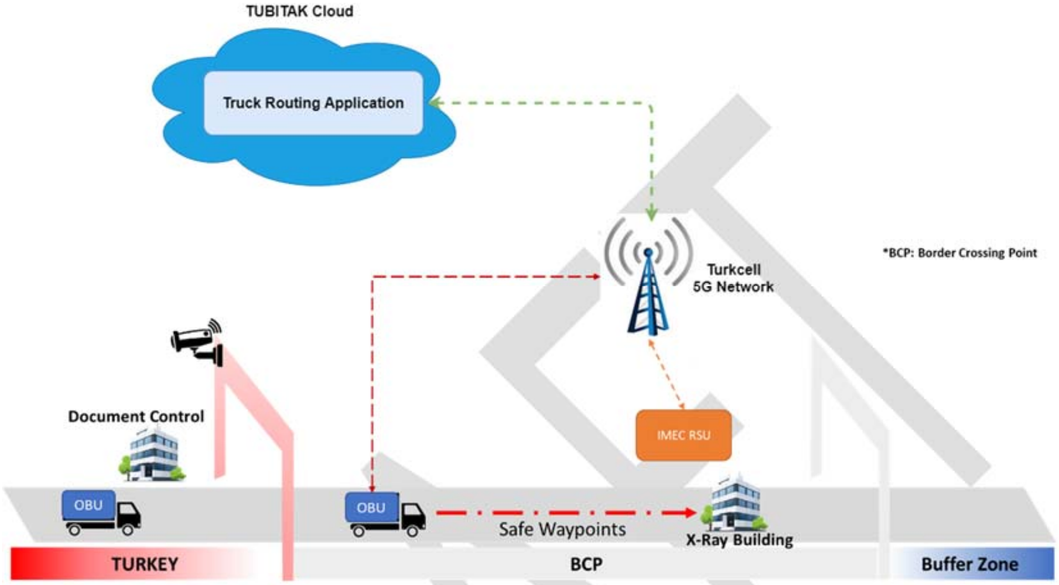


Figure 48: Truck routing user story components

Information flow

Since the Truck Routing Application runs on the TUBITAK Cloud, the different information flows between the OBU, RSU and the Truck Routing Application need to be established over the 5G network. The description of the information flow between cloud application, OBU and RSUs for the Truck Routing user story is presented below:

- When a Ford truck enters the TR custom, a routing request is sent from its OBU to the Truck Routing Application.
- A routing request confirmation message is sent from the Truck Routing Application to the OBU.
- A request is sent to the RSUs from the Truck Routing Application. RSUs will start publishing LiDAR data when this request arrived.
- The truck sensors and state information is sent continuously from the OBU to the Truck Routing Application.
- The LiDAR data is sent continuously from the RSUs to the Truck Routing Application.
- Calculated safe waypoints are sent from the Truck Routing Application to the OBU.
- When the truck reaches the X-ray building, a routing ending message is sent from the Truck Routing Application to the OBU.

- A request will be sent to the RSUs from Truck Routing Application. RSUs will stop publishing LiDAR data when this request arrived.

Figure 49, Figure 50 and Figure 51 depict the message flow for the Truck Routing user story. RSUs and OBU are directly sending messages to the TUBITAK Cloud via the Turkcell 5G network.

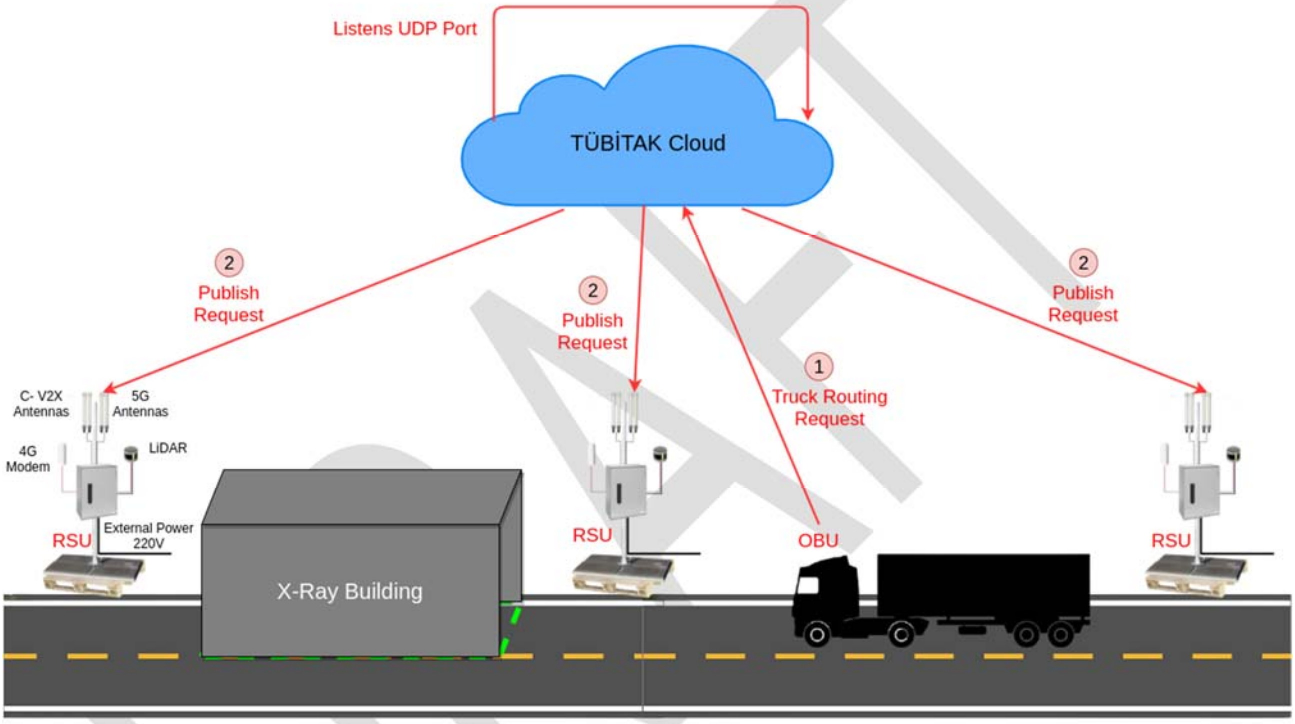


Figure 49: Message flow of truck routing starting phase

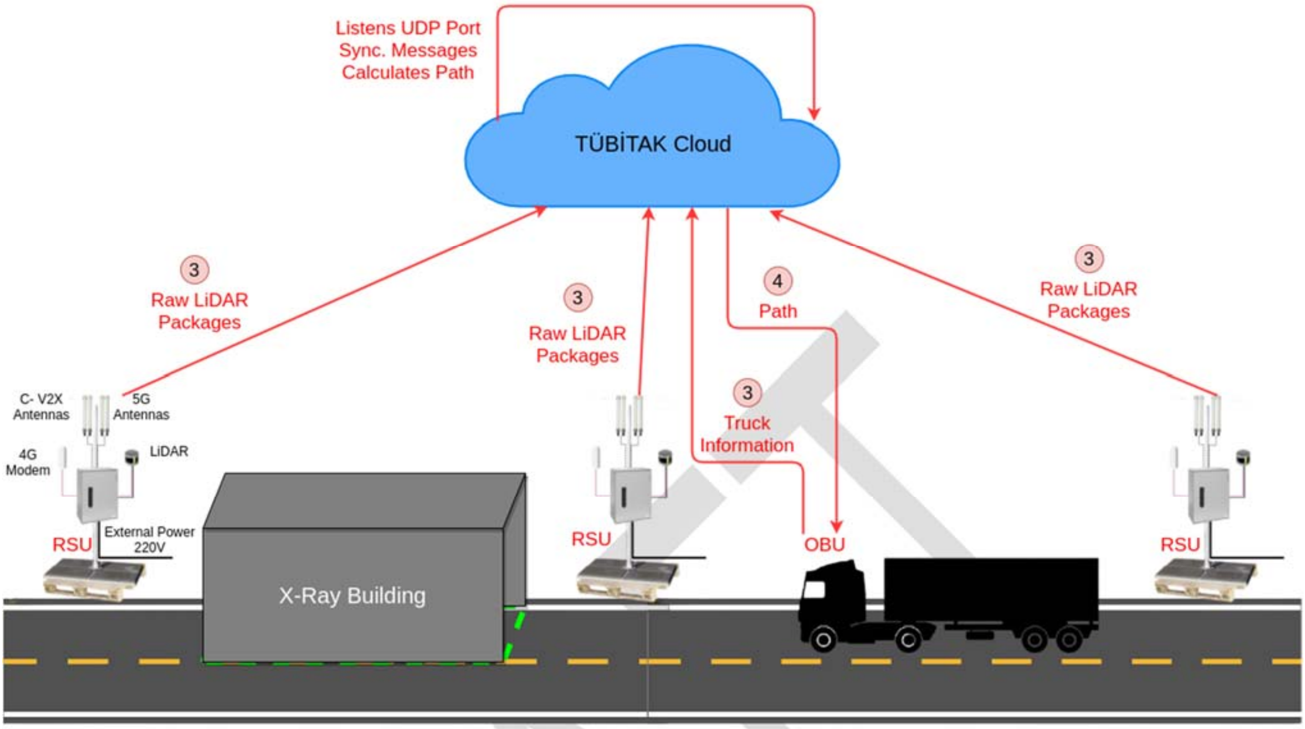


Figure 50: Message flow of truck routing executing phase

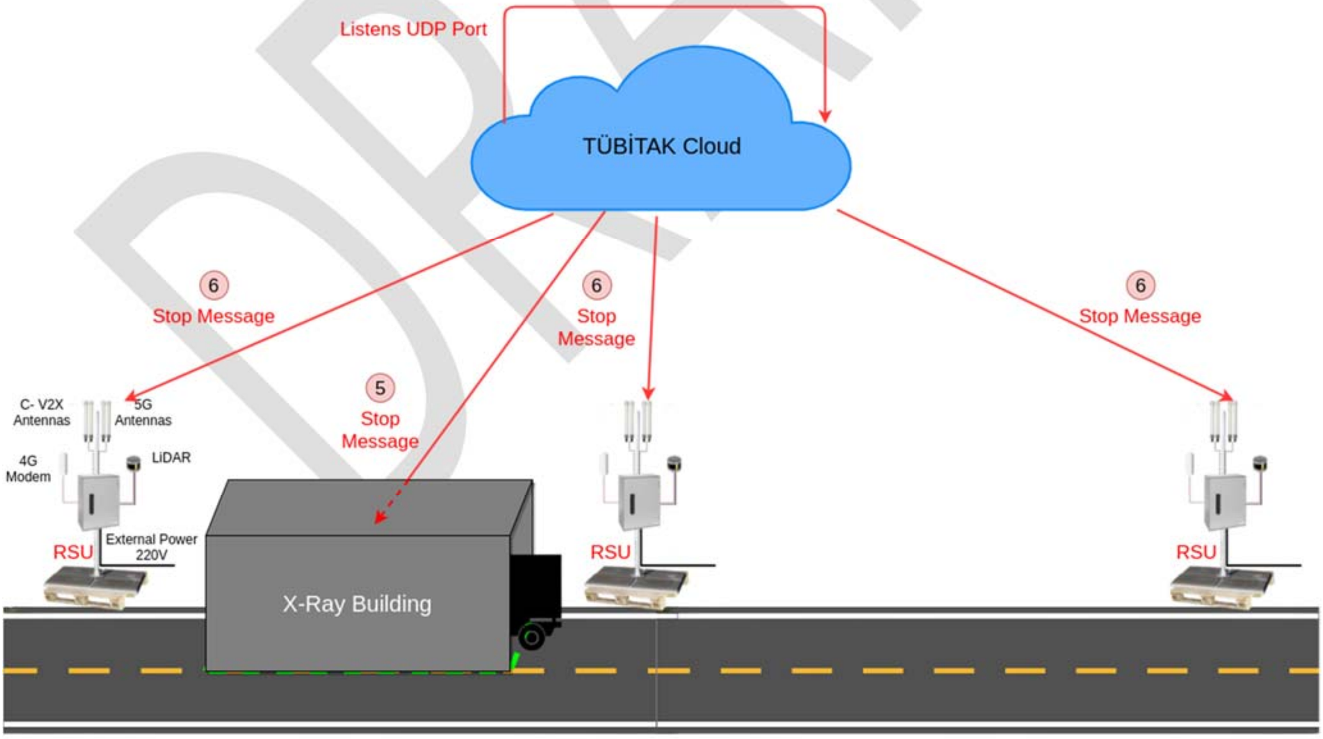
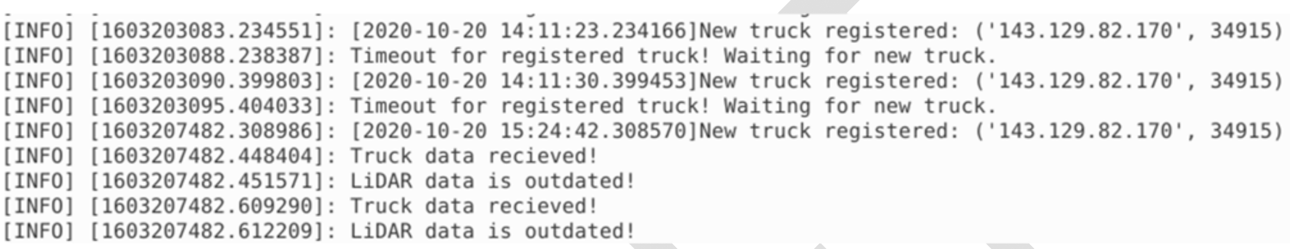


Figure 51: Message flow of truck routing ending phase

Server

The LiDAR drivers listens for LiDAR data while the Truck Routing Application's server node listens the UDP port for incoming OBU messages. When a Ford truck sends a routing request via the Dynamixel OBU, the truck is registered and the "start broadcasting LiDAR data" message is sent to the RSUs. After that, incoming messages from the RSUs and OBU are synchronized and sent to the processing nodes. If the OBU does not send any messages for a certain period, the registration is deleted and must be re-registered. Also, if RSUs do not send new LiDAR messages, the old LiDAR data is marked as "stale" and the forwarding is stopped until fresh LiDAR data arrives. Figure 52 depicts server info messages on the console.



```
[INFO] [1603203083.234551]: [2020-10-20 14:11:23.234166]New truck registered: ('143.129.82.170', 34915)
[INFO] [1603203088.238387]: Timeout for registered truck! Waiting for new truck.
[INFO] [1603203090.399803]: [2020-10-20 14:11:30.399453]New truck registered: ('143.129.82.170', 34915)
[INFO] [1603203095.404033]: Timeout for registered truck! Waiting for new truck.
[INFO] [1603207482.308986]: [2020-10-20 15:24:42.308570]New truck registered: ('143.129.82.170', 34915)
[INFO] [1603207482.448404]: Truck data recieved!
[INFO] [1603207482.451571]: LiDAR data is outdated!
[INFO] [1603207482.609290]: Truck data recieved!
[INFO] [1603207482.612209]: LiDAR data is outdated!
```

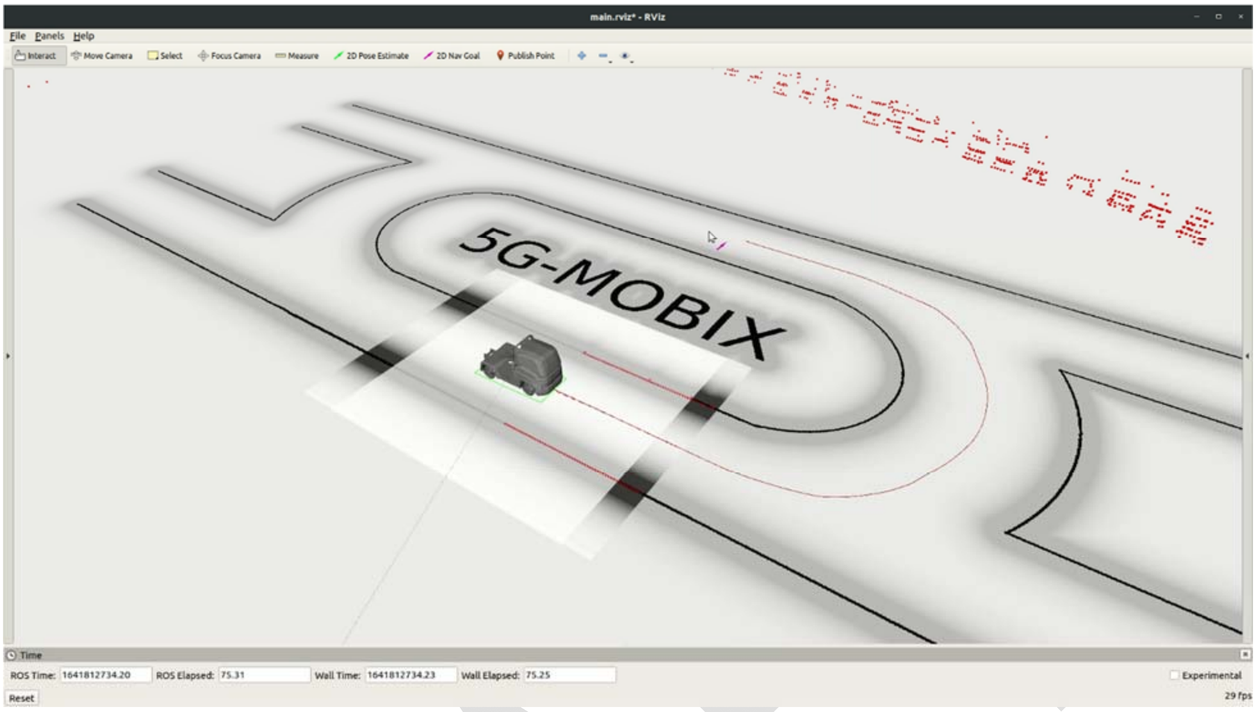
Figure 52: Example of server node info messages on console

Localization

The local dynamic map node generates an updated map of the customs area. After the LiDAR data is converted into a point cloud in the LiDAR driver, an object detection algorithm filters the objects from the data. These objects and truck location are then sent to the local dynamic map node. The generated map is used by the path planning algorithm.

Path planning

The path planning module uses two different path planning algorithms. First, the global path is planned using the A* algorithm on the static map. However, vehicle dynamics and dynamic obstacles are not considered in the global path. Then, the local path is planned periodically using TEB Local Planner. The local planner considers the vehicle dynamics and the local dynamic map while planning a path. The output of the local planner is then transformed into safe waypoints and sent to the OBU over the server node in a message containing 250 waypoints with the (x, y, z V) format, where V indicates the speed of the vehicle while x, y and z indicate the position of the waypoint. Under normal conditions, the vehicle will drive at a maximum speed of 10 km/h. Figure 53 depicts the path planning application on the local dynamic map.



ROS Time: 1641812734.20 ROS Elapsed: 75.31 Wall Time: 1641812734.23 Wall Elapsed: 75.25 □ Experimental

29 fps

Figure 53: Example of path planning output on the local dynamic map

User story UML diagrams

To better reflect the information flow among the different components participating in this user story, and to show the Truck Routing Application architecture, two UML sequence diagrams were designed. While Figure 54 represents the message flow between the Truck Routing user story components, Figure 55 shows the Truck Routing Application workflow on the server.

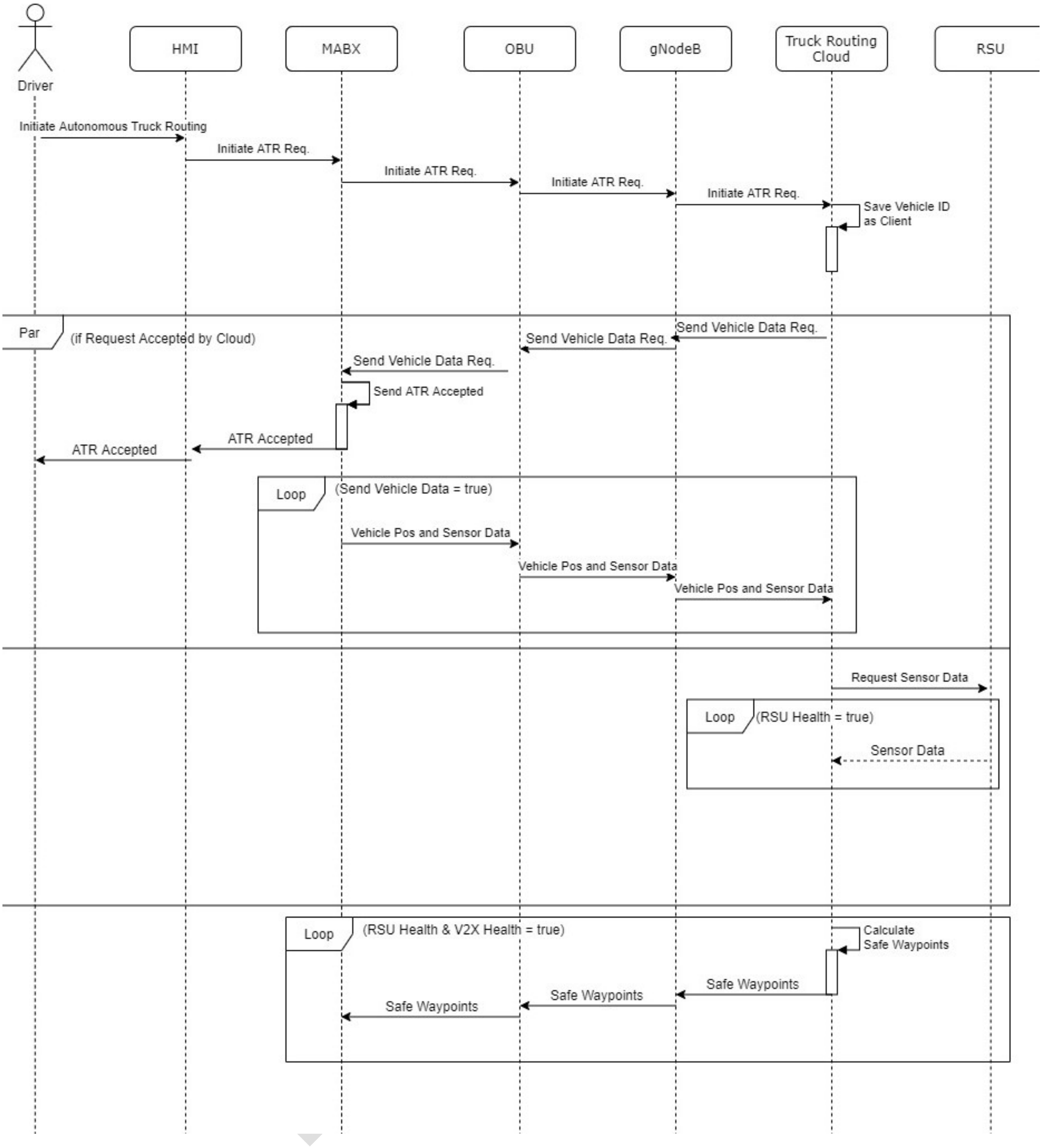


Figure 54: UML diagram of message flow between components in the Truck Routing user story

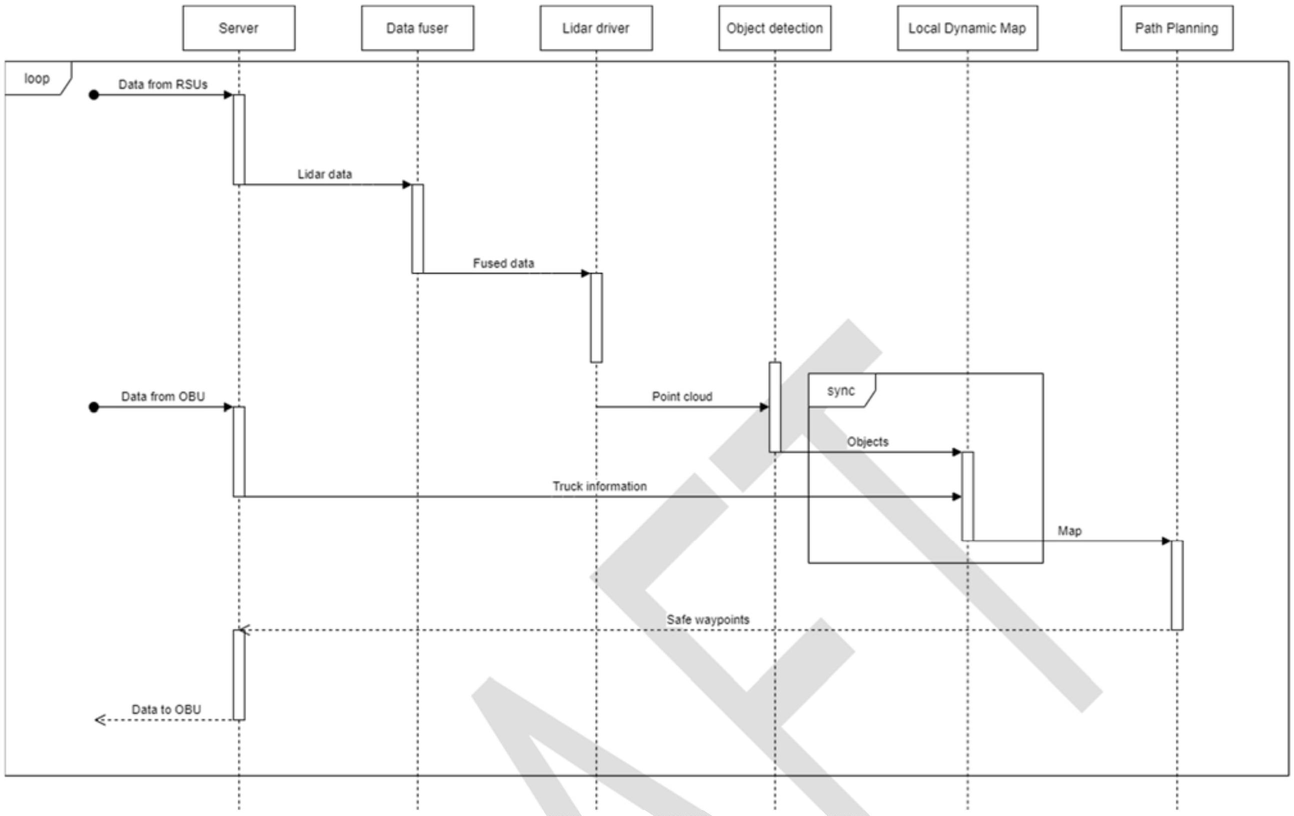


Figure 55: UML diagram of Truck Routing Application workflow

Truck routing hardware and software requirements

The platform HW/SW requirements and the characteristics of the accompanying Truck Routing Application are presented in the following. The Truck Routing Application is deployed to a virtual machine on the TUBITAK Cloud. The TUBITAK virtual machine specifications are the following:

- RAM: 16 GB
- Storage: 80 GB
- vCPU: Dynamically allocated from cloud
- OS: Ubuntu 18.04 LTS

The Truck Routing Application developed as Robot Operating System (ROS) package for modular design. The ROS is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. The software specifications for the Truck Routing Application are detailed next.

- ROS Melodic Morenia
- Python 2.7
- C++11

7. INFRASTRUCTURE INTEGRATION AND TESTING ACTIVITIES

7.1. ES-PT CBC

In the ES-PT CBC, the CCAM functionalities are not provided by single elements as described above but are provided by sets of coupled elements. So, in this section we describe the process of integration and testing of those sets of elements. This process involves the following stages:

- Testing HW elements / SW applications in testing environment individually.
- Testing HW elements / SW applications in testing environment all together.
- Integration works of HW elements on site / Deployment of application.
- Testing HW elements / SW applications in real environment.

7.1.1. Traffic Radars (Sensor + RSU)

The 5G traffic radars used in the ES-PT corridor consist of 2 independent elements that work together: a traffic radar that detects vehicles in the area of interest, and a 5G RSU that generates the appropriate ITS messages based on the sensor detections and provides the 5G connectivity.

In a first phase, the partners involved (CTAG, IT) had to verify that the sensors were working properly: calibrate the radars, check that detections are correct and that the messaging meets the requirements stipulated for interfacing with RSU. In the same way, the RSUs have also had to be tested to verify that they have 5G connectivity, and that the applications are running under the established requirements.

The next step has been to interconnect the sensors with the RSUs in a controlled test environment, thus verifying that the detections made result in the transmission of the appropriate ITS messages, and that these messages contain the correct information for each detection. A messaging and communication protocol between the radar and the RSU was defined. The I2V messages that were decided to be used were CPMs, but as they were not yet fully defined at the beginning of the project, it was decided to start the implementation with DENMs. Once the implementation with DENMs was done, it was tested by injecting logs from the radar and looking at the output to the MEC and when we obtained favourable results, we moved on to testing on the CTAG tracks. When the CPM standard allowed us to do so and the functionality with the DENMs was operational, we moved on to the new messaging and tested again on the table by injecting the radar logs and then on the CTAG tracks, which allowed us to move on to the real scenarios.

For the integration and deployment phase, it has been necessary to adapt the sites to house the CCAM equipment, in some cases requiring the construction and installation of specific supports to place the equipment in the appropriate position. The relevant permits have been requested from the appropriate regulatory bodies (DGT, Ministry of Fomento) both for the installation of these supports and for the deployment of the equipment on the actual road sites.

Finally, the last phase of the deployment consisted in checking that once the equipment was installed, the functionality was fulfilled. To this end, the connectivity of the RSUs to the project's 5G network was checked, the correct detection of the vehicles circulating on the motorway was verified, and finally it was confirmed that the whole system was transmitting the ITS messages correctly and with the correct information from the detections.

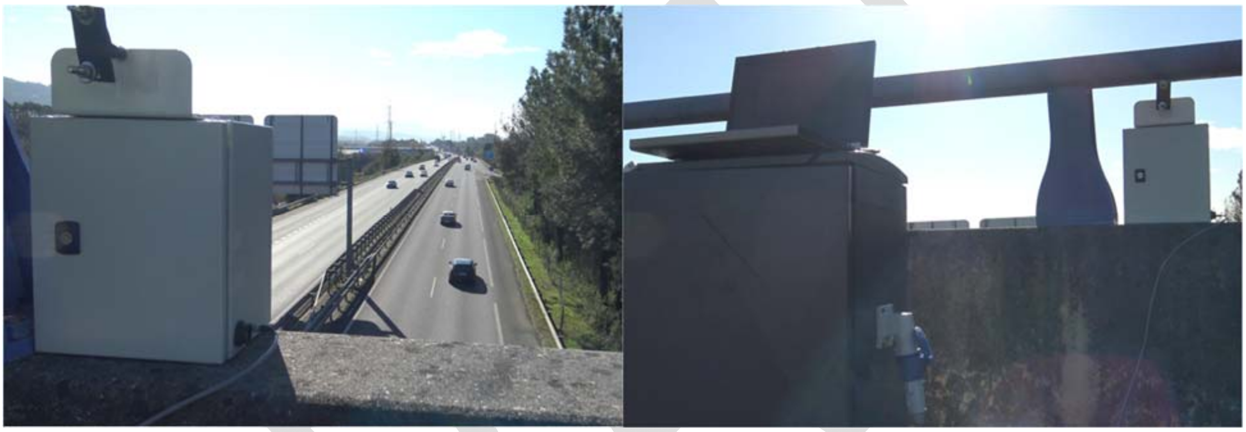


Figure 56: CTAG traffic radar installed in the AP9-A55 lane merge in Spain

7.1.2. Pedestrian detectors (Sensor + RSU)

Pedestrian detectors used in the ES-PT CBC are composed of different elements, following a similar scheme as the traffic radars described before: a sensor which is responsible for capturing the scenes and detecting elements, and a 5G RSU which acts as a connectivity provider with support for standardized ITS messages.

The first step in the integration is to check the correct functionality of the sensors. For the Portuguese case, Siemens had to calibrate its radar, check that detections are adequate, and that the messaging used meets the requirements of the interface with the RSU. In the Spanish case, the CTAG's sensor is a set of camera and processing unit which must be calibrated and configured, tested in a controlled environment to check that the detections and the response to the RSU interface are correct, as can be seen in Figure 57.

In addition, the RSU is tested in a controlled environment to verify that it provides 5G connectivity, and that the running applications meet the stipulated requirements.

Systems are then completed by connecting the sensor to the RSU in a laboratory environment so that full functionality can be verified.

For the integration and deployment phase, supports had to be built and installed to house the CCAM infrastructure. The relevant permits had to be requested beforehand from the competent authorities (Ministry of Fomento, Infrastructuras de Portugal).

To complete the deployment, tests have been carried out with the equipment installed on the actual site. These tests are intended to verify that the equipment complies with the specified functionality once it has been deployed, so that the connectivity of the RSUs with the project's 5G network is checked, the correct detection of elements in the scene is verified, and that the complete system transmits the expected ITS messages through the RSU with the appropriate form and content.

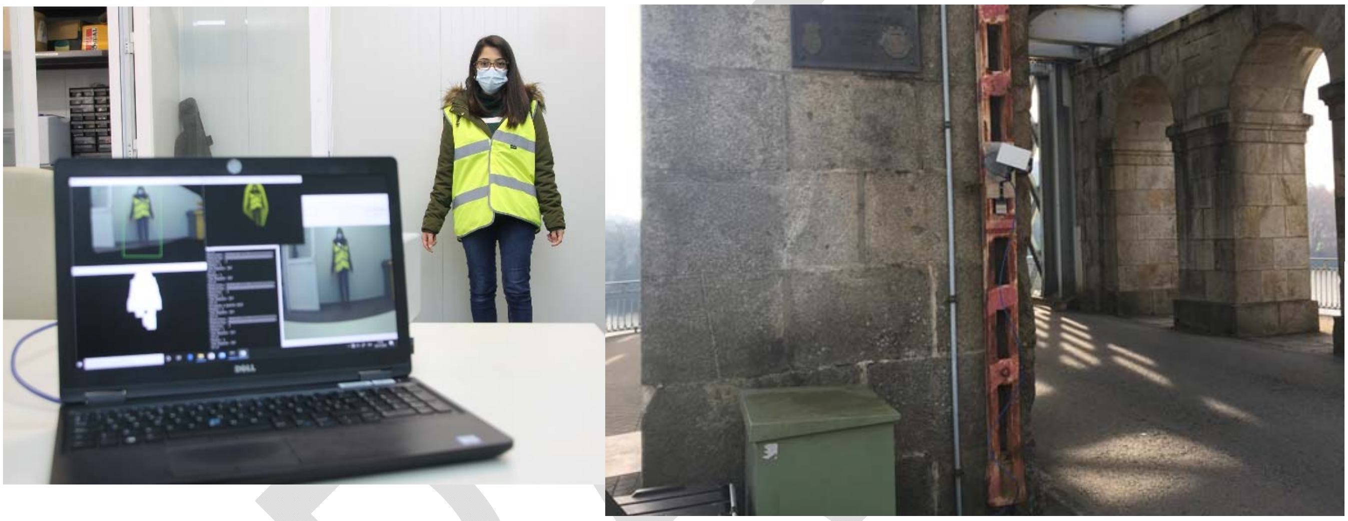


Figure 57: Laboratory and real environments tests of the Spanish pedestrian detector by CTAG

Specifically, for the VRU use case, an application for VRU detection has been developed within CTAG vision modules. The application has been prepared to detect events in the image and subsequently compare them again with the objects in the area that has changed. The detected objects are processed by a classifier that is able to differentiate between the different types of VRU and that by applying machine learning is able to improve the identification algorithm. In addition, the algorithms have been optimised and improved to work in real time and minimise communication delays with the IoT module.

In Figure 58 the VRU detection application behaviour is described. The left image white image areas with possible events can be observed. In the second one image, the classifier acts on the event zone, detects the VRU and marks in yellow the VRUs in the danger zone. There are other pedestrians in the image that are not selected. Finally, in the right image, the VRU can be observed already detected with a bounding box around it.

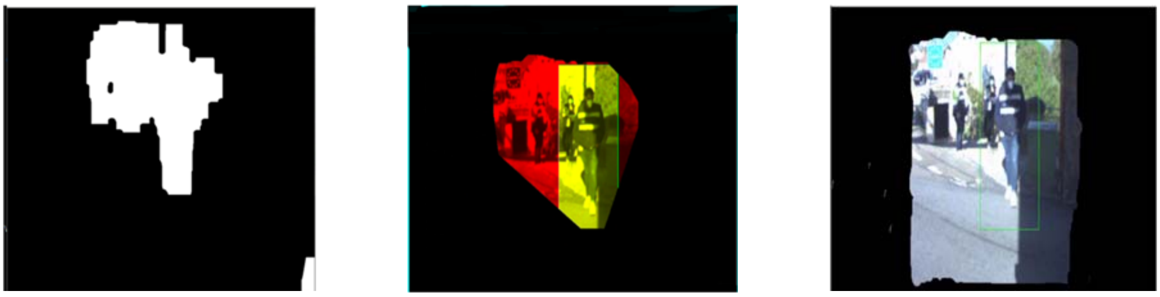


Figure 58 VRU detection event

At the communication level, a messaging and communication protocol was defined between the camera and the RSU, and a messaging simulator was created to be able to carry out independent tests. After the tests with the simulator, tests were carried out on the table by injecting real logs from the camera and after these, the tests were passed on to real tests on the CTAG tracks.

7.1.3. VRU warning system (mobile + MEC)

The VRU warning system consists of two elements (the mobile application and the MEC application) that must be tested and integrated individually in a first phase, for being integrated and tested together in a second phase. For test the VRU smartphone application, several function tests have been successfully conducted.

7.1.4. MQTT brokers interconnection

In this subsection, the integration and testing activities of the ES and PT MEC brokers are presented for the Advanced Driving and HD Maps use cases.

Advanced Driving use cases

The general architecture is defined as follows (see Figure 59):

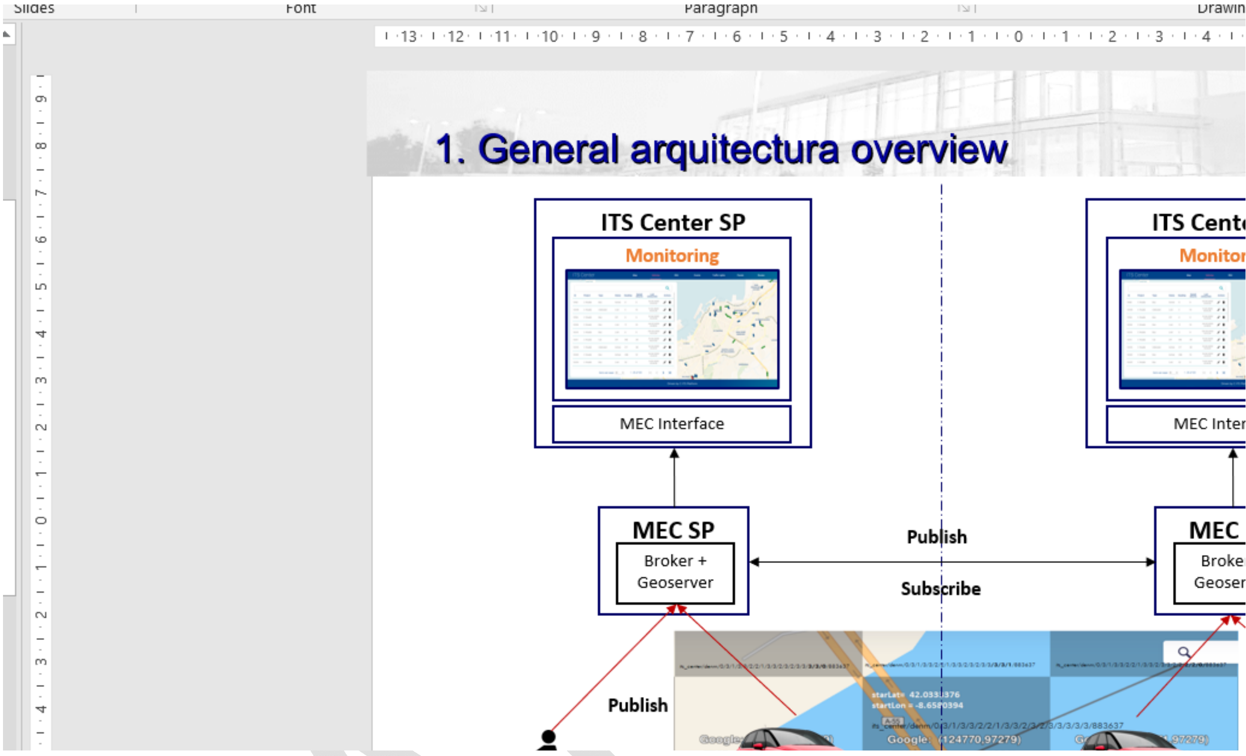
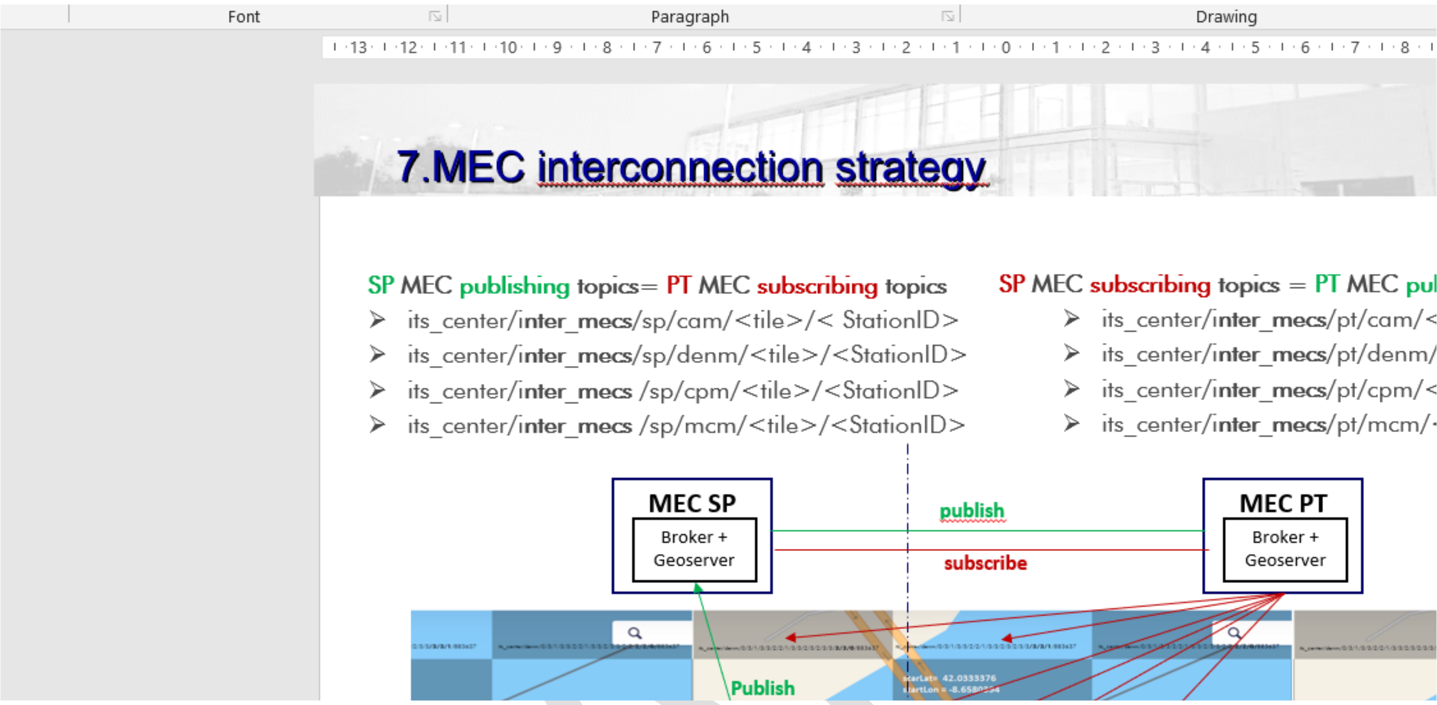


Figure 59: MEC interconnection general architecture

Connected elements such us Vehicles, VRUs, etc. publish ETSI encoded messages to the MEC broker they are connected to. The MEC Geoserver will republish the messages in the corresponding adjacent tiles, and the vehicles that are subscribed to those tiles will receive them. In the same way, the Geoserver deployed in the other MEC will subscribe to the tiles located in the border, in order to receive, not only the messages published in its country, but also the messages published in the other side of the border. This mechanism guarantees that the border is covered by both Geoservers, so that the vehicles will receive the messages published in the border regardless the MEC they are connected to (interconnection assured via *inter_mecs* topics in both brokers) (see Figure 60).



7.MEC interconnection strategy

SP MEC publishing topics= PT MEC subscribing topics

- its_center/inter_mecs/sp/cam/<tile>/< StationID>
- its_center/inter_mecs/sp/denm/<tile>/<StationID>
- its_center/inter_mecs /sp/cpm/<tile>/<StationID>
- its_center/inter_mecs /sp/mcm/<tile>/<StationID>

SP MEC subscribing topics = PT MEC pu

- its_center/inter_mecs/pt/cam/<
- its_center/inter_mecs/pt/denm/
- its_center/inter_mecs/pt/cpm/<
- its_center/inter_mecs/pt/mcm/

Figure 60: MECs Interconnection topics

Finally, the messages published in the border surroundings are republished by both Geoservers in the dedicated channels.

Tests carried out and tests to be performed

The tests carried out in October 2020 consisted of publishing and receiving CAM messages in the MQTT broker, and verifying that those messages were received by a OBU connected to the other MEC:

- ES OBU connected to ES MEC publishes in the broker -> PT OBU connected to PT MEC receives the information.
- ES OBU connected to PT MEC publishes in the broker -> PT OBU connected to ES MEC receives the information.
- PT OBU connected to PT MEC publishes in the broker -> ES OBU connected to ES MEC receives the information.
- PT OBU connected to ES MEC publishes in the broker -> ES OBU connected to PT MEC receives the information.

In all tests, the information was correctly received and decoded by the OBUs, showing that both MEC brokers and Geoservers are functioning well and the interconnection between ES and PT systems is working. The tests also proved that data and message interoperability at the device level is guaranteed.

The previous interconnection tests were performed in a test environment from both sides, Spanish and Portuguese. The aim was to assure that the setup works at application level. It will be necessary to perform those tests again in the final environment.

HD Maps use case

For this use case, the interconnection between MECs, consists in just one MQTT topic/channel created for sending a JSON file that contains the necessary parameters and information to perform the maps database update. This is in order to update the vehicles connected to the MECs, just in case they can create their own maps database, instead of using the one generated and published by the ITS Center they are connected to. This integration architecture is visible in Figure 24.

This scenario is planned to be used in the trials only in by the PT Connected Vehicle, but all OBUs will be able to receive the JSON messages. In order to verify that all is correct at application level, the verification will be as follows:

- Spanish OBU connected to Spanish MEC receives a JSON file published by the Portuguese MEC.
- Spanish OBU connected to Portuguese MEC receives a JSON file published by the Portuguese MEC.
- Portuguese OBU connected to Spanish MEC receives a JSON file published by the Spanish MEC.
- Portuguese OBU connected to Portuguese MEC receives a JSON file published by the Spanish MEC.

7.1.5. Remote Driving

Remote driving on a 5G network demands a set of essential requirements to be understood. This user story implies to know what quality the network can provide at each moment in two terms: video quality and network latency in the data channel.

Moreover, the consistency in our investigations must be endorsed by reputable international organizations and hence this use case two prestigious international forums have been chosen, 5GAA and VQEG.

5GAA is the largest group of manufacturers of automobiles, 5G networks and devices in relation to autonomous driving. In this forum Nokia is working on the cases of "Tele-operated Driving", since it is contributing in the 5G-MOBIX project with the architecture of the system and documenting the result of measurements in the remote driving user story. On the other hand, the quality of the low latency driving video will be evaluated by the VQEG, a group of world video experts that carries out systematic tests of video quality in various formats (also making proposals and recommendations to the ITU).

Tests have been performed to research the impact of latency in ToD for different latencies in a real car and a closed circuit. Furthermore, Nokia is working on a detailed study of circuit deviations as shown below. The key to this study is to know the relation between the speed of the car and the latency of the network to know if the remote driving is being in safety way or by contrast the vehicle must stop.

Next step is, once the speed, latency and video quality values are known, the implementation of an automatic speed control mechanism in the car and in the cockpit that, depending on the network latency at all times, will limit the maximum driving speed, or whether it will completely stop remote driving.

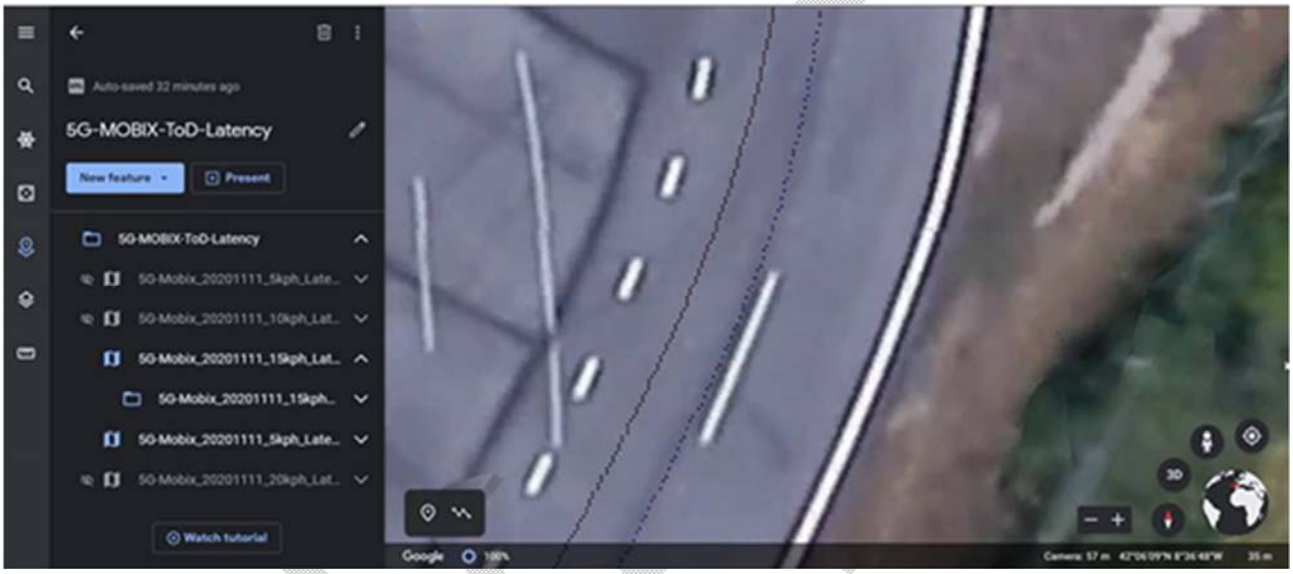

Auto-saved 32 minutes ago
5G-MOBIX-ToD-Latency
New feature
Present
5G-MOBIX-ToD-Latency
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_10kph_Lat...
5G-Mobix_20201111_15kph_Lat...
5G-Mobix_20201111_15kph...
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_20kph_Lat...
Watch tutorial
3D
Google
100%

Figure 61: Latency study in remote driving (Example 1)

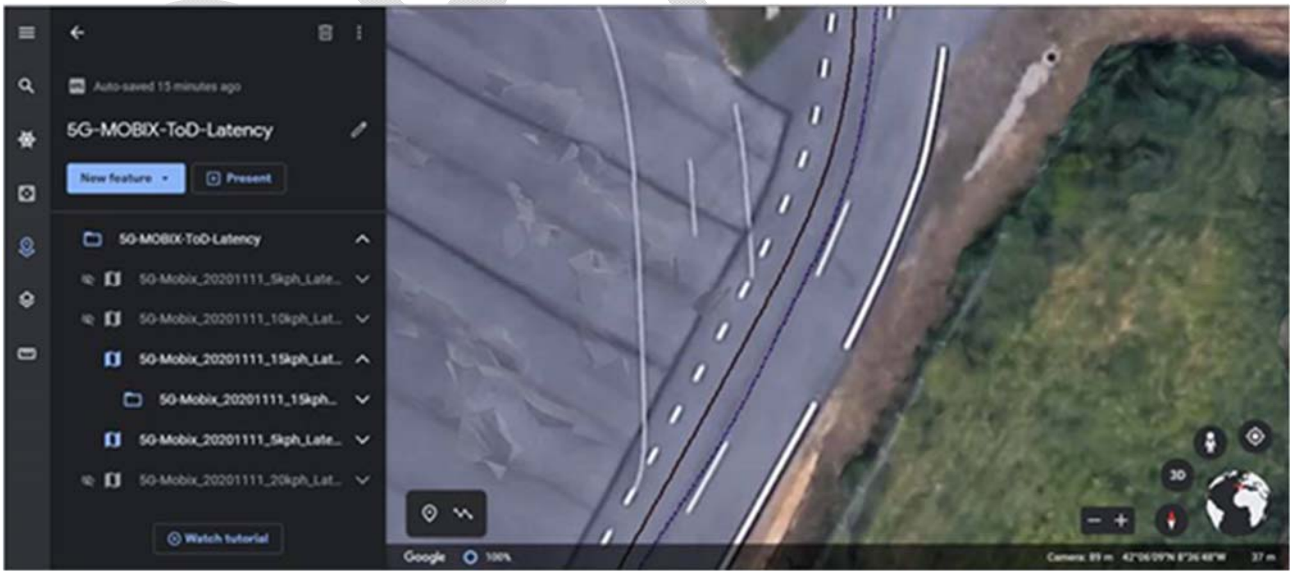

Auto-saved 15 minutes ago
5G-MOBIX-ToD-Latency
New feature
Present
5G-MOBIX-ToD-Latency
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_10kph_Lat...
5G-Mobix_20201111_15kph_Lat...
5G-Mobix_20201111_15kph...
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_20kph_Lat...
Watch tutorial
3D
Google
100%

Figure 62: Latency study in remote driving (Example 2)

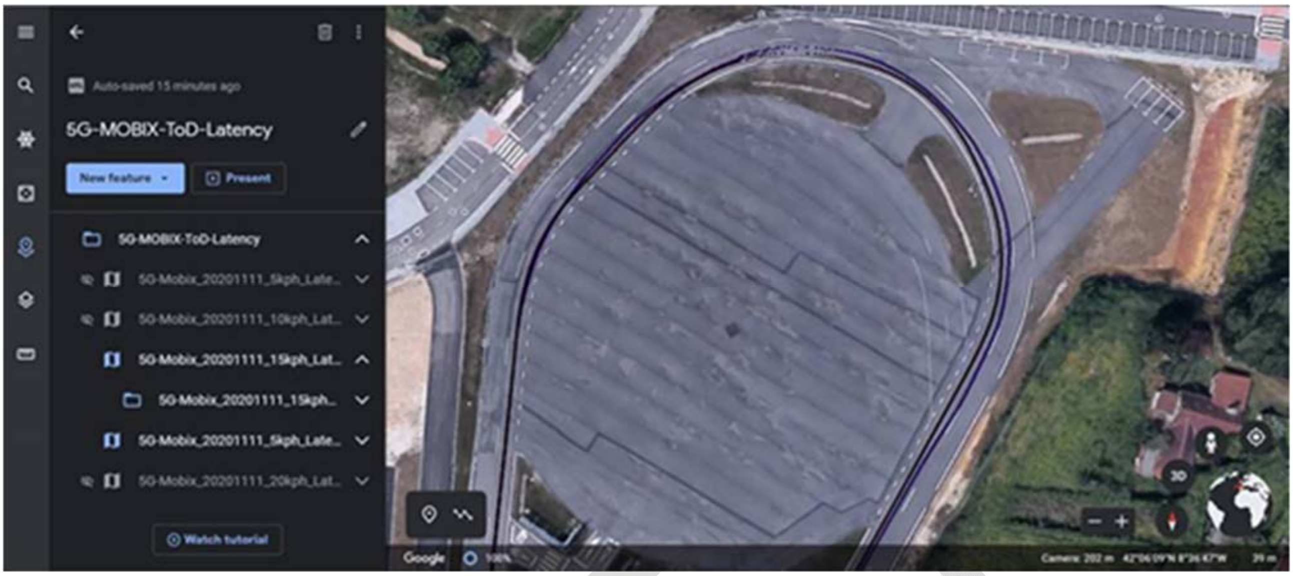

Auto-saved 15 minutes ago
5G-MOBIX-ToD-Latency
New feature
Present
5G-MOBIX-ToD-Latency
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_10kph_Lat...
5G-Mobix_20201111_15kph_Lat...
5G-Mobix_20201111_15kph...
5G-Mobix_20201111_5kph_Late...
5G-Mobix_20201111_20kph_Lat...
Watch tutorial
Google
3D
Camera: 202 m 42°06'09"N 8°36'47"W
39 m

Figure 63: Latency study in remote driving (Example 3)

The first tests have been performed, as shown in Figure 61, Figure 62, and Figure 63. The driving values without glasses, and with glasses and latency of 150 milliseconds have been obtained and displayed in a map. It should be noted here that the outcome of remote driving trials will be reported within WP4, however, the first tests confirm that the main issue of ToD lies in the curves.

Regarding the description of the hardware elements and software applications and the process of testing, integration and deployment of the components, several iterations were executed starting from a more unitary testing perspective and evolving towards an end-to-end deployment execution.

Phase 1. Testing HW elements / SW applications in testing environment individually.

From an individual point of view, 4 full-HD resolution cameras and a Jetson Xavier NX board have been tested. The tests basically consist in sending four UDP video streams acquired by the cameras and rendering them using the same board or other PC.

In a first iteration, correct behavior is tested using localhost connections in the same board. In this way we can measure the latency due to the encoding and decoding process plus application management. In a second iteration, the board is connected to a different IP address than the cameras and the latency measurements, as well as the ideal bitrates are measured. In a third stage, the same tests are executed within a mobile 4G environment and a MEC which is a replica of the final MEC via a VPN connection.

Then in other testing area, we have the part that comprises a Unity AR application which combines real video streams with a virtual cockpit. In this testing area, the correct rendering in several HMD devices is done. The interaction with physical cockpit device of this application (Logitech one) is still pending and will be achieved during the forthcoming months.

Phase 2. Testing HW elements / SW applications in testing environment all together.
A remote driving application has also been developed. This application is distributed in three different places: the cockpit, the MEC and the OBUs.

The following integration tests have been done: sending commands and receiving feedback from the OBU. What is missing is performing a test with a real OBU connected to a vehicle so that the commands go to the CAN bus of the vehicle and the data of the vehicles sensors is propagated back.
In order to be able to test all the SW applications together an iCreate robot has been used as a vehicle. The cockpit application has been executed in the real Logitech G29 device and a MEC instance which is a replica in terms of software of the one deployed in Vigo has been used. Finally, the Jetson Xavier board has been equipped with the four wide angle webcams. Everything has been connected using Wi-Fi network and the robot was driven like a normal vehicle using the full stack of software to be used in the final scenario. Latency has been measured with a 1ms resolution for the remote commands part and with 10ms resolution for the video part. In this test normal video (not mixed reality one) has been used to perform the driving operation of the robot. In any case, the same VPN server was used in order to transmit transparently all the flows in a controlled IP range.

Phase 3. Integration works of HW elements on site / Deployment of applications.
In a more advanced integration scenario, a real driving test with the car has been performed incorporating the video transmission and reception in the HMD. The HMD, in this case is an Oculus Quest which is running the application described in the previous paragraphs.

A customized Nvidia Jetson Xavier NX board with software that will allow to render real time UDP video from 4 wide angle cameras located in the car and finally, a 5G modem which is replaced in this case by a regular mobile phone connecting the board and the HMD via tethering. At this point the Jetson Xavier NX board acquires the video, which is then transmitted via 5G to the MEC and also received via 5G in the headset. We have just performed an initial integration testing of this, checking that the video is available in the HMD. For safety reasons we have executed a previous step of this test in which the cameras and HMD are connected using an internal Wi-Fi connection in the car. In this way we have more control on the video latency. Further tests including some tuning in the radio connection and the HMD video buffer side will be performed using just 5G connectivity. These tests have been done using the 5G infrastructure already deployed at CTAG premises.

Phase 4 Testing HW elements / SW applications in a real environment.
Testing the whole E2E Setup requires having a real OBU connected to the 5G Network at the same time as the Nvidia board with the cameras. A PC connected to the MEC with attached HMD and with the remote driving control protocol software running on it is required for the final integration tests. This requires fixed IP address provided at CTAG premises has connectivity to MEC instance running at Vigo location.

Such tests need to cover the following areas:

1.E2E ping/TCP/UDP connectivity from/to

- PC Cockpit - HMCU/OBU board
- PC Cockpit - NVidia Jetson board
- PC Cockpit - MEC Remote Driving instance (Spanish MEC Vigo)
- MEC (Spain) - MEC (Portugal)
- HMCU/OBU Board - MEC (Spain)
- 1.HMCU/OBU Board - MEC (Portugal)

2.Video streaming tests

- Stream 4 cameras full-HD 4MBit bitrate each to PC Cockpit via MEC
- Stream 3 cameras full-HD 4MBit bitrate each to PC Cockpit via MEC
- Stream 1 Camera full-HD 4Mbit + 3 HD Cameras 3Mbit
- Measure video latency tests
- Measure/Estimate approximate video packet loss

3.Remote driving protocol

- Check that **individual** control commands are arriving and acting into the vehicle
- Check that instrumentation measurements from the vehicle arrive to cockpitMeasure E2E latency for 2 previous bullets should be in the range < 100 milliseconds

All these tests have successfully performed in CTAG track as well as in the Old Bridge. The HMD being used is Oculus Quest2. The video frame loss has been measured both from transmission point of view in Jetson board as well as from the cockpit point of view while decoding video in the PC which is directly connected to the device ITS Center.

7.1.6. ITS Center

The ES and PT ITS Center logical structure and message flows were already presented in Subsection 6.1.1. Here we will present how the ITS Centers can be tested together. As explained in subsection 7.1.4, ITS Centers are not connected between each other for Advanced Driving Use Cases. The ITS Center are only used for monitoring purposes. In this way, both ITS Centers will subscribe to the Geoserver deployed in their country, this subscription includes CAM messages in the tiles located at the border on the side of the opposite country.

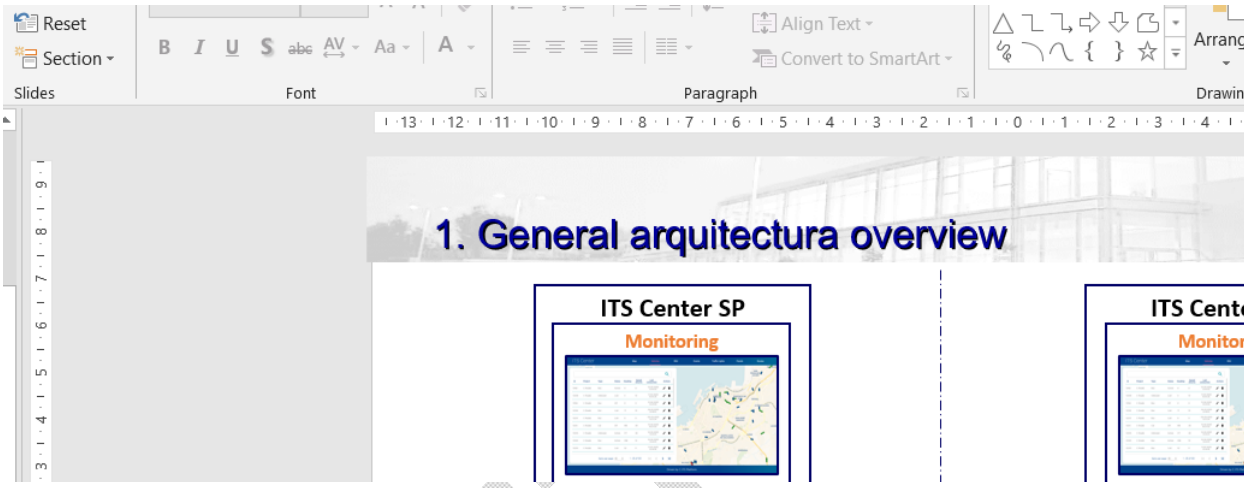


Figure 64: ITS Center connection to MEC

Test to be performed

There were no tests performed between both sides yet. But it is necessary to try out to register a Portuguese OBU in the Spanish ITS Center, and vice versa. The test to be performed must be:

- Spanish OBU publish CAM messages to the Spanish MEC, and Spanish and Portuguese ITS Centers monitors that OBU.
- Spanish OBU publish CAM messages to the Portuguese MEC, and Spanish and Portuguese ITS Centers monitors that OBU.
- Portuguese OBU publish CAM messages to the Spanish MEC, and Spanish and Portuguese ITS Centers monitors that OBU.
- Portuguese OBU publish CAM messages to the Portuguese MEC, and Spanish and Portuguese ITS Centers monitors that OBU.

HD Maps use case

As shown in section 7.1.4 MQTT brokers interconnection, the ITS Centers will only communicate between each other to send the HD Map database generated. This is to make the updated HD Map database available for download via SFTP to the vehicles that are connected to both ITS Centers. This integration is depicted in Figure 24.

In order to verify the interoperability between sites, the following tests must be carried out:

- OBU connected to Spanish ITS Center publishes sensor data, ITS Center generates maps database, and the OBU and Portuguese ITS Center download the maps database. Portuguese ITS Center makes it available for another OBU connected to it that will download it as well.

- OBU connected to Portuguese ITS Center publishes sensor data, ITS Center generates maps database, and the OBU and Spanish ITS Center downloads the maps database. Spanish ITS Center makes it available for another OBU connected to it that will download it as well.

ATMS integration module

The ATMS integration module at the Portuguese ITS Center will publish DENM messages to the Portuguese MEC. Depending on the location of the DENM message, these messages may be received by vehicles on both sides of the border or only at the Portuguese side of the border.

In order to verify the functionality of the ATMS integration module, the following tests must be carried out:

- ATMS integration module publishes a DENM message, located at the border, to the Portuguese MEC, a Portuguese and Spanish OBUs connected to the Portuguese MEC receive that message.

- ATMS integration module publishes a DENM message, located at the border, to the Portuguese MEC, a Portuguese and Spanish OBUs connected to the Spanish MEC receive that message.

- ATMS integration module publishes a DENM message, located away from the border at the Portuguese side, to the Portuguese MEC, a Portuguese and Spanish OBUs connected to the Portuguese MEC receive that message.

ATMS integration module publishes a DENM message, located away from the border at the Portuguese side, to the Portuguese MEC, a Portuguese and Spanish OBUs connected to the Spanish MEC don’t receive that message.

7.2. GR-TR

7.2.1. Assisted “zero-touch” Border-Crossing

The “Assisted “zero-touch” Border-Crossing” application developed by WINGS, has already passed all the necessary functionality and integration tests, required before the actual trials. These tests were focused on assessing the following aspects:

- Proper communication among the Application server, the OBU and the roadside Infrastructure developed by WINGS

- Proper reception and transmission of messages to and from the WINGS application.

- Correct output of the developed algorithms based on different data input.

- Proper logging of all necessary metrics according to the agreed format, allowing for post-processing and KPI calculations.
- Proper integration with the FORD truck and its onboarded IMEC OBU
- First insights into the performance of the application over the 5G testbed in terms of achieved throughput and end-to-end latency.

These tests happened in multiple stages in order to not hinder development and to circumvent the implications of COVID-19, which restricted travel, and hence the ability of the project partners to meet and perform tests together, in the designated test trucks and the actual cross-border corridor. The performed tests are listed below:

- Test 1
  - Date / Location: July 2020 / Cosmote 5G testbed lab (Athens, Greece)
  - Test target: Verify connectivity of the OBU and RSU with the 5G testbed. Verify proper message exchange between the OBU/RSU and the WINGS platform
- Test 2
  - Date / Location: October 2020 / Cosmote 5G testbed lab (Athens, Greece)
  - Test target: Verify the proper end-to-end functionality of the Assisted "zero-touch" border crossing use case over the Cosmote 5G testbed
- Test 3
  - Date / Location: June 2021 / GR side of the GR-TR cross-border-corridor (Kipoi, Greece)
  - Test target: Verify the proper functionality of the US in the actual GR-TR corridor using the Cosmote 5G network. Get access to early logs and measurements to gauge the performance of the network and the performance of the application.
- Test 4
  - Date / Location: December 2021 / Ford test truck with 5G coverage (Eskisehir, Turkey)
  - Test target: Remote end-to-end integration between the Ford truck, the IMEC OBU and the WINGS OBU, to verify proper exchange of messages and the WINGS OBU ability to autonomously instruct the truck to stop in case of a VRU detection.

Some early test results from the WINGS OBU performance from Test 1 and 2 are presented in Section 4.5 of Deliverable D3.2 of 5G-MOBIX (5G-MOBIX, Deliverable D3.2: Vehicle adaptation for CCAM use cases, 2020), while the full verification test results for this US are reported in deliverable D3.6 .

7.2.2. IMEC RSUs

After completing the assembly of the three RSUs that are installed at the Turkish customs area, IMEC performed a series of tests in its premises to ensure their expected correct operation. In that direction and as a first step, IMEC firstly tested each component of each RSU individually.

IMEC firstly ensured that each individual component can be remotely accessible and that they are correctly interconnected through the managed switch (Planet IGS-4215-4P4T). At the processing unit side (PCEngines APU3D4), IMEC installed the Ubuntu 18.04 operating system and all the required software that is used to create the necessary functionality for each use case within the GR-TR test site. Then, the GPS module (Navilock NL-8012U) that is used for time synchronization with the other ITS stations (e.g., OBUs and RSUs) has been tested. Concerning the 4G (Mikrotik wAP LTE kit) and 5G (Quectel RM500Q) modems, the wireless interfaces have been configured and it has been verified that the modems can be connected to the network. Due to the lack of a 5G network in Belgium during the testing period, the 5G modems have been tested only on a 4G network. The 5G connectivity is tested in FORD's and Turkcell's premises. Additionally, the 5G modems have been updated to the latest firmware version provided by Quectel. Regarding the C-V2X PC5 modules (Cohda MK6c), they have been updated to the latest firmware version provided by Cohda and they have been configured to communicate with the processing units of the RSUs. Moreover, it has been verified that the integrated GPS of each C-V2X PC5 module works as expected and that it can get a GPS fix. IMEC examined the correct operation of the short-range wireless communication link over PC5. Finally, the recovery system (PCEngines APU2E4) system has been tested to verify the remote management and recovery of the individual components of the RSUs. Furthermore, it has been confirmed that the managed PDU (Robot Electronics ETH008) is able to power cycle remotely the individual components in case they do not respond for any reason.

Afterwards, the operation of the RSUs as a whole and the end-to-end communication between the three RSUs have been tested and verified. To this end, IMEC used custom testing tools installed in each processing unit to transmit standardized V2X messages between the three RSUs wirelessly via both short-range C-V2X PC5 and long-range C-V2X Uu links.

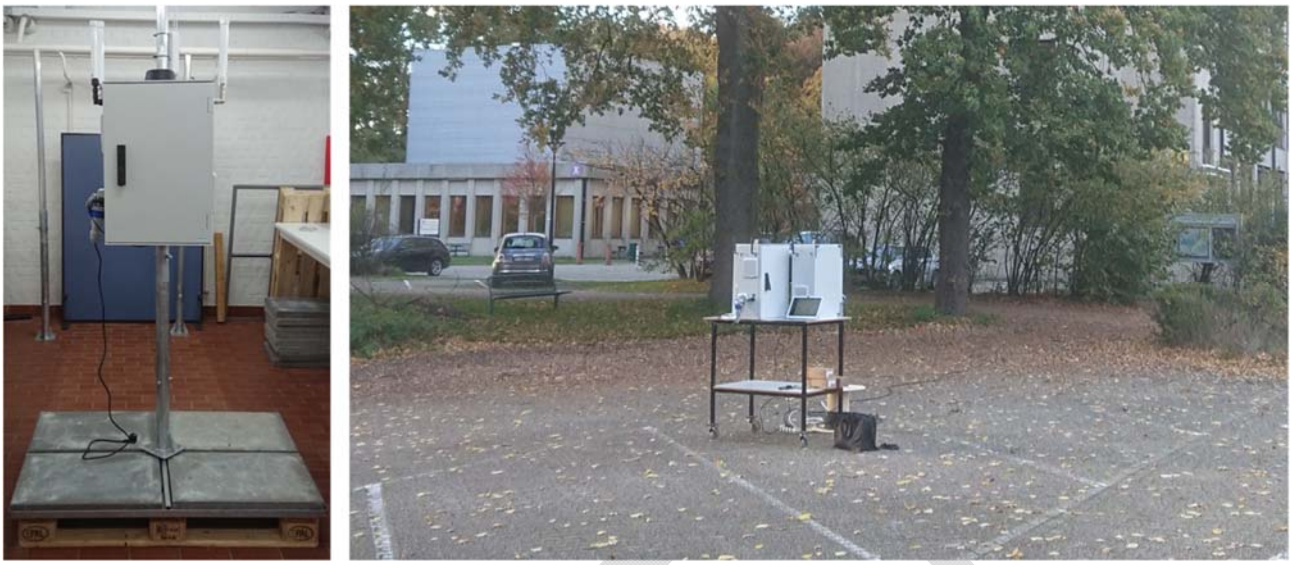


Figure 65: The IMEC RSUs are being evaluated before shipped to Turkey

Figure 65 shows the IMEC RSUs being evaluated before they were shipped to Turkey. The left side of the figure shows an RSU as it is being configured, while the right side of the figure depicts the RSUs placed outdoors, so that they can lock the required satellites and get a GPS fix in order to verify the successful end-to-end communication between them.

7.2.3. Truck routing in customs site application

Truck Routing end-to-end integration tests were carried out at Ford OTOSAN Test Field (Eskişehir, Turkey). During the tests, 1 Ford truck with IMEC OBU installed, 2 IMEC RSUs and the Truck Routing Application on the TUBITAK Cloud were used. 5G base stations that Turkcell installed on the field for the tests were used as the communication infrastructure. Tested and verified features are described below:

- Communication between RSUs and cloud application.
- Communication between OBU and cloud application.
- Lidar data fusion in the cloud
- Truck localization in the cloud
- Following the safe waypoints by the vehicle.
- Autonomous routing on obstacle-free straight road.
- Autonomous routing on obstacle-free curved road.
- Slowing down and stopping when encountering an obstacle during routing.

8. FUTURE WORK

8.1. ES-PT

The ES-PT cross border corridor has completed the integration and deployment of the roadside infrastructure elements, such as roadside sensors, MEC applications, ITS Center applications and so on. The verification phase has also been completed and it can ensure that every element works as expected and fulfils its intended functionality.

A first phase of tests has been carried out on the national scenarios on the Spanish side on the tracks and on the A55 and on the border scenario on the old bridge, which have allowed adjustments and corrections to be made and again to verify both the infrastructure and the rest of the sensors and applications deployed.

However, real road trials continue and it is possible under these circumstances that some bugs or unknown and unexpected issues may appear (especially in the high-speed New Bridge scenario). In this case, the partners involved in these issues will work on fixing the problems in the best way in order to make the trials work successfully. Apart from these potential issues, the T3.4 partners are involved in a continuous maintenance process during the whole trial periods (early trials and full trials), which involves regularly checking the equipment functionality and service availability, monitoring the equipment behaviour, and fixing any “expected” issue included in the assumptions of the design, such as that ones that may arise due to the normal use.

8.2. GR-TR

The GR-TR partners have developed, integrated and tested the infrastructure and applications needed to support the GR-TR. Currently the GR-TR partners are planning their on-site visit for the execution of the GR-TR trials as reported in the WP5 Test Cases and in the WP4 Trial planning. It is understood that during the trials potential bugs and issues may arise that will require the quick reaction of the respective partners in order to end up with fully functional trials. This process will entail troubleshooting and fixes on both HW and SW in order to accomplish the first trials, while later on upgrades on various infrastructure components and applications may be needed, based on the observation of early results. More specifically the next steps envisioned by the GR-TR partners during the trialling phase, from a WP3 perspective are:

- Intense support during trials for quick troubleshooting and fixes, as necessary.
- Monitoring of trial outcomes and fine-tune / reconfigure HW and SW as needed.
- Maintenance of HW/SW components during the entire duration of the trials.
- Upgrades in HW/SW components and applications based on trial results and early post-processing to improve performance.
- Testing and integration of new upgrades and continuous monitoring and support.

- Continuous feedback cycles: Trial results → Post-processing → Upgrades & fixes → testing and integration of upgrades/fixes → New trial session.

9. CONCLUSION

This document reports on the CCAM infrastructure development and deployment activities at the different cross-border corridors and trial sites. Furthermore, the provided information describes in detail the enhancements and evolution of the applications that implement the use case specific functionality of each of the sites. The described rollout activities are following the 5-phase plan of the 5G-MOBIX project and conclude the activities according to the previously laid out roll-out plan.

The document is structured in such a way that it first provides an overview on the planning and global alignment of the activities that were conducted by the individual sites. Following this, an overview on the CCAM infrastructure deployments, the overall CCAM components employed in the TSs and CBCs and integration contributions of each local trial site to cross-border corridors is provided.

The remaining sections provide a detailed description of the roadside infrastructure components, the deployment and enhancement activities that have been conducted, the evolved applications and platforms, as well as the integration and testing activities of the CBCs. The same rich contents for the trial sites from DE, FI, FR, NL, CN and KR is provided in the Annexes, following the same structure as the CBC sections. While the contributions of the specific TSs and CBCs in terms of CCAM infrastructure differ depending on the various use cases targeted by the individual efforts, this report summarizes the successful deployment of the diverse set of development achievements. Accordingly, this report is an important milestone of the project, providing insights on the CCAM infrastructure aspects utilized in the 5G-MOBIX trialling activities.

REFERENCES

5G-MOBIX. (2019). *Deliverable D2.2: 5G architecture and technologies for CCAM specifications.*

5G-MOBIX. (2019). *Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM.*

5G-MOBIX. (2020). *Deliverable D3.1: Corridor and Trial Sites Rollout Plan.*

5G-MOBIX. (2020). *Deliverable D3.2: Vehicle adaptation for CCAM use cases.*

5G-MOBIX. (2020). *Deliverable D3.3: Report on the 5G technologies integration and roll-out .*

5G-MOBIX. (October 2019). *Deliverable D2.1: 5G-enabled CCAM use cases specifications.*

A-200094, 5. T. (2020). *Working Group Sysem Architecture and Solution Development - V2X Application Layer Reference Architecture.*

ETSI EN 302 637-2. (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.*

ETSI EN 302 637-3. (2014). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service.*

ETSI GR MEC 035. (06-2021). *Multi-access Edge Computing (MEC);. Study on Inter-MEC systems and MEC-Cloud systems coordination.*

ETSI GS CIM 009. (V1.2.1. 2019-10). *Context Information Management (CIM); NGSI-LD API.*

ETSI GS MEC 003. (03-2022). *Multi-access Edge Computing (MEC); Framework and Reference Architecture.*

ETSI GS MEC 030. (04-2020). *Multi-access Edge Computing (MEC); V2X Information Service API.*

ETSI TR 103 562. (2019). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2.*

ETSI TS 122 185. (V16.0.0, 2020-08). *LTE, Service requirements for V2X services.*

ISO TS 21184. (n.d.). *Cooperative intelligent transport systems (C-ITS) — Global transport data management (GTDM) framework.*

Smartmicro. (2014). UMRR Automotive Type 29 Data Sheet. Retrieved from https://www.autonomoustuff.com/wp-content/uploads/2016/08/Smartmicro-Type-29-Data-Sheet.pdf

VAN HUIS, J., BAAN, J., & VAN LOON, A. (2009). *Traffic Monitoring using videobased vehicle tracking.* ITS Stockholm.

10. ANNEXES

The CCAM infrastructure developments and deployments conducted by the individual TSs were put here in the annex to keep the focus of the main document on the CBCs. The individual TS sections follow the same structure as the main document above.

10.1. Trial Site DE Roll-out

10.1.1. Introduction

The German trial site is located in the city center of Berlin and comprises two use cases, namely eRSU-assisted Platooning and EDM-enabled extended sensors with surround view generation. The overall architecture that describes the trial site's functionalities is depicted in Figure 66.

While the details of the presented architecture have been provided in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), Figure 66 has been updated to reflect the updated infrastructure that has been further enhanced since then. Most notable is the inclusion of the Near Edge solution operated by MobiledgeX[2], that is used for deploying the Vehicle Discovery Service and the EDM service, the 5G modems to provide connectivity to the 5G networks available at the DE TS, and the PC5 capable hardware by Cohda Wireless, which enables the communication via sidelink C-V2X at the DE TS. In the following we describe the different infrastructure components that have been developed and deployed for the use cases in more detail.

10.1.2. Overview of Infrastructure Components

To create the EDM-Service the DE TS roadside infrastructure features in total nine extended roadside units (eRSU), which have been deployed along the digitized road stretch to provide extended perception to vehicles. The eRSU components, as depicted in the Far Edge part of Figure 66, and in a road-centric perspective in Figure 72, include the following hardware:

- HD IP Camera Sensors.
- Far Edge compute Servers by Neousys.
- PC5 Sidelink C-V2X Hardware by Cohda Wireless, MK6c integrated in two eRSUs
- 5G modem from Quectel (Uu interface) or direct network connection via TUB backhaul network.

[2] https://mobiledgex.com/

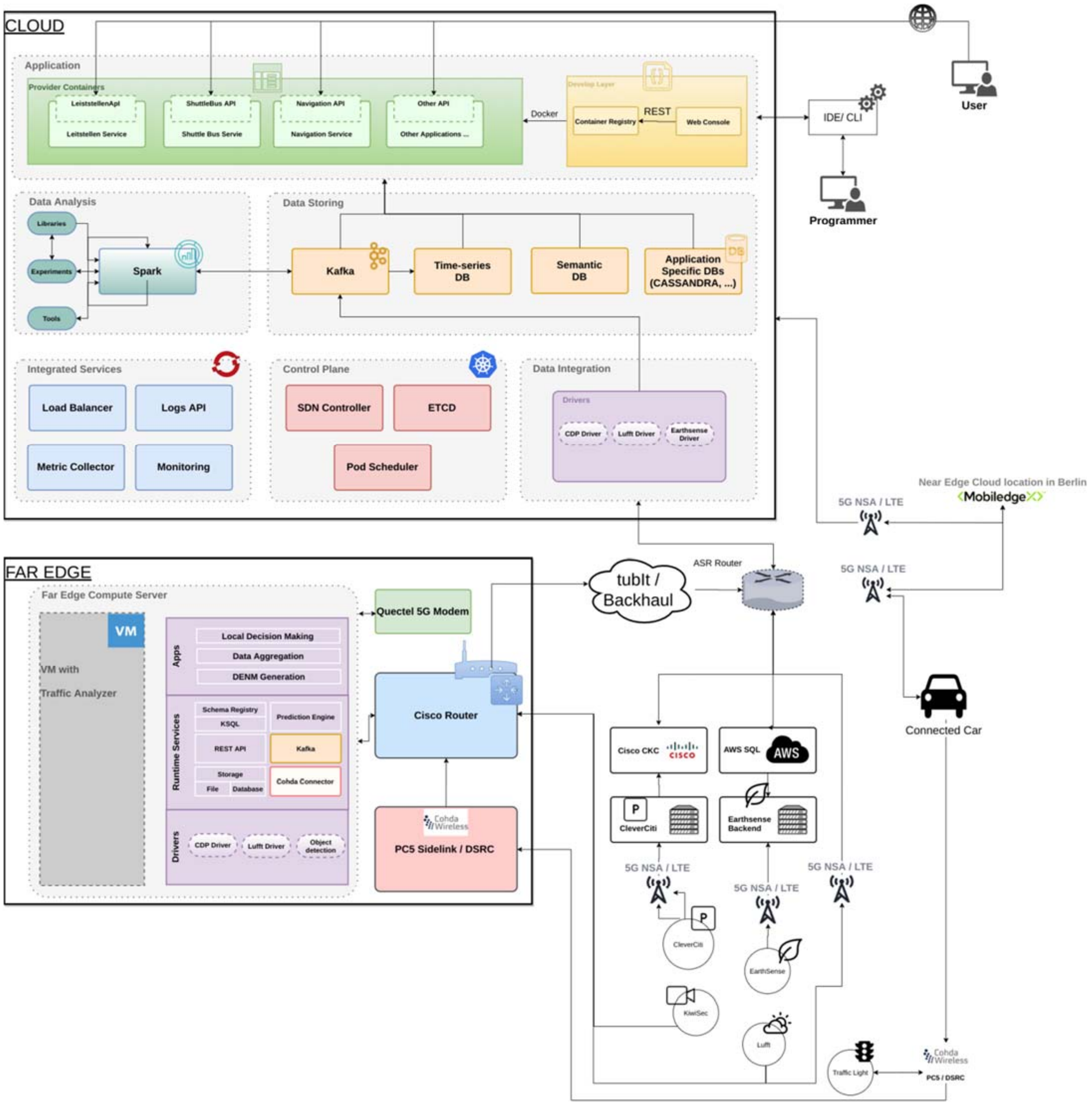


Figure 66: Overall architecture of DE TS

A vast range of sensor infrastructure feeding in the data from roadside, including road-condition, traffic light, parking lots, weather and environmental conditions, and intelligent streetlights is available from earlier projects to enrich the perception of CCAM as well. In Figure 66, these sensors are depicted on the lower right part.

10.1.3. Infrastructure Deployment and Enhancement Activities

In the following we describe the infrastructure deployment and enhancement activities necessary for the DE TS use cases.

Sensor Development and Deployment

For the eRSU-assisted platooning use case the eRSU provides a traffic analysis based on object detection from live HD video IP camera sensors (see Figure 67). In total twenty of these standard HD IP video cameras have been deployed, minimum two at each of the eRSU sites. The compressed video stream coming from the cameras is evaluated instantly in the eRSU, as described below.

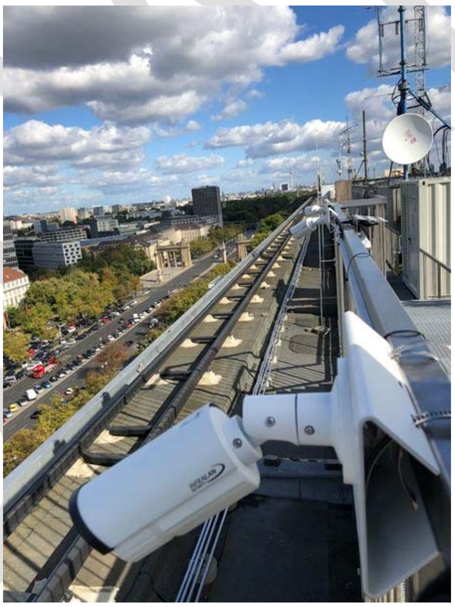


Figure 67: Camera Sensors deployed at TU Main Building

RSU / Edge Development and Deployment

The eRSUs, depicted in Figure 68, utilize their internal edge compute server (far edge Mobile Edge Computing (MEC)) to perform the traffic analysis and detect objects on a per frame basis and subsequently create an EDM based on this information. The object detection is very demanding in terms of CPU and GPU power and the following hardware specs were the minimum required to realize the object detection from two cameras simultaneously:

- Neousys Nuvo-6108GC
- Intel Core i7-7600
- 16GB DDR4 RAM
- Nvidia GeForce RTX2080 GPU
  512GB SSD

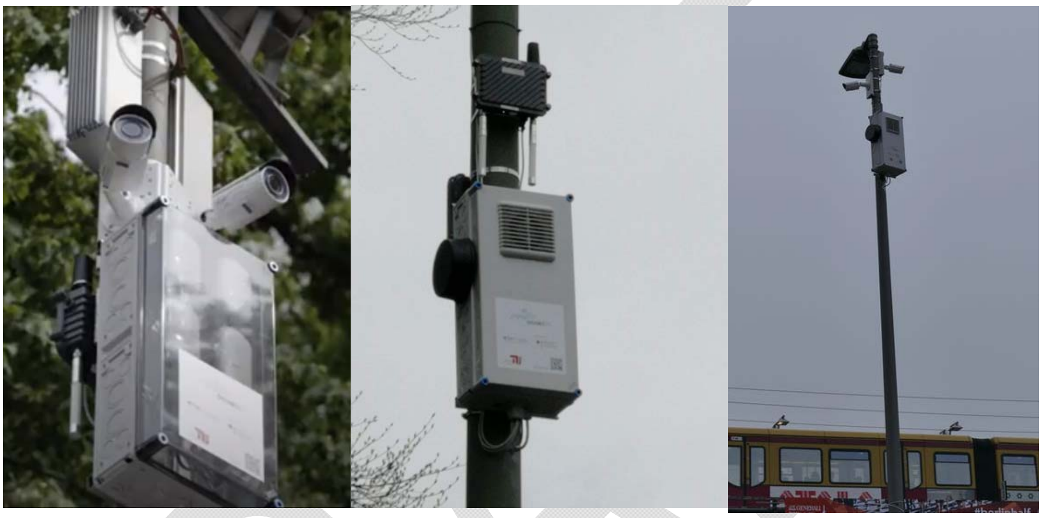


Figure 68: Three examples of the nine currently deployed eRSUs at the DE TS along Str. des 17. Juni in Berlin

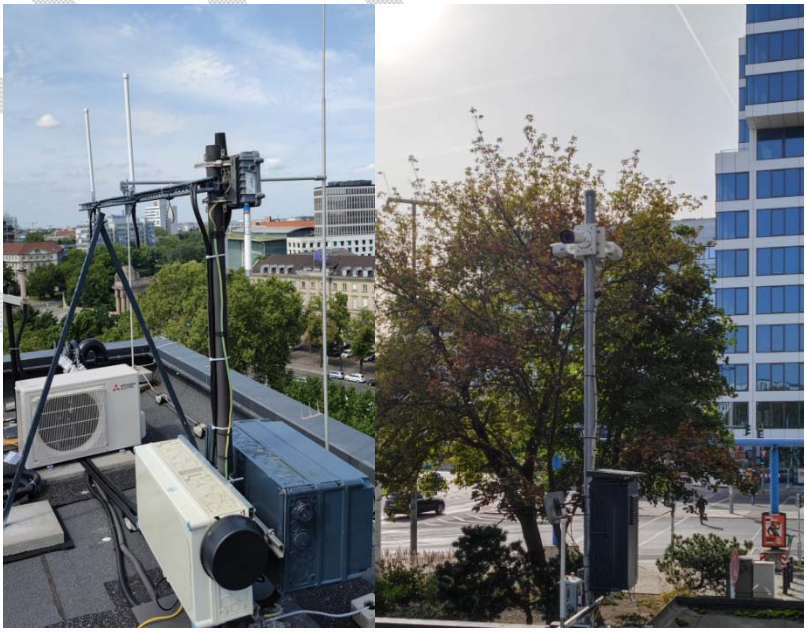


Figure 69: PC5 Deployment for DE TS based on Cohda MK6C

In addition to this equipment, two eRSUs have been extended with 5G modems from Quectel (RM500Q-GL) and C-V2X hardware from Cohda Wireless (MK6C). In order to utilize the 5G modem, the operating system of these eRSUs has been upgraded to Ubuntu Server 20.04 with LTS, which has a more recent Linux Kernel > 5.4.14 to support the Quectel RM500Q 5G modem. The current setup utilizes ModemManager tool to communicate with the 5G modem, however, the Quectel Manager has also been tested and works as expected.

To communicate with vehicles that are participating in the DE TS experiments, and for applications that need to exchange only small amount of data, the eRSU and the OBUs communicate directly through the PC5 interface provided by the Cohda MK6C EVK connected to the eRSU. The Cohda MK6C EVK device main component is the Qualcomm MDM 9150 C-V2X chipset, which implements the C-V2X functionality according to the 3GPP Release 14 and 15 specification. The MK6C unit also comes with a GNSS unit that is used to receive GPS signal of high precision.

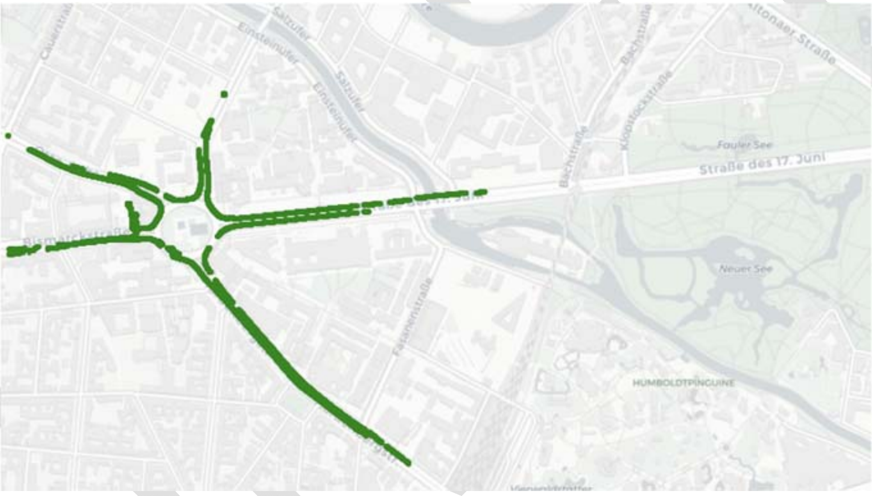


Figure 70: Coverage area for one of the two PC5 deployments at DE TS

In addition to communication via PC5 coverage, we have equipped two eRSUs to make use of the 5G Uu interface provided by the Quectel 5G RG500Q and the Quectel RM500Q-GL module, which are mounted in an evaluation board and connected to the eRSU. Subsequently the 5G Uu connection can be used for data exchange with the applications deployed at MobiledgeX near edge MEC or with the ITS center.

Cloud Deployment

The DAI Datacenter provides the Cloud components depicted in Figure 66, in order to operate, manage and provide the services from the integrated DE TS Far Edge. Most notable is the OpenShift-based Management of the Edge PCs, which is deployed on all eRSUs except the ones featuring a Quectel 5G modem, as the CentOS-based Linux from OpenShift does not support the recent 5G hardware chipsets.

The DAI Cloud infrastructure also hosts the ITS center application with the Dashboard and control center applications for the German trial site.

On top of that, the DAI Datacenter also provides the resources to deploy the core parts of TUB's designed distributed monitoring framework. The main components are a Prometheus server, an Elasticsearch database and a Grafana server. In the DAI Datacenter, all metadata and metrics generated by the components of the DE-TS are collected, processed and visualized, enabling the monitorization of the user stories and of the CCAM infrastructure as a whole.

The Monitoring system based on Prometheus and Grafana is hosted in TUB cloud gathering all the logs and metrics coming from applications (QoE), communication libraries (QoS), edge systems (SLA) and modems (network logs), of which Figure 71 gives an example. It shows a dashboard created with Grafana where the metrics are categorised in three different control and observation levels, which are radio, network and application levels.

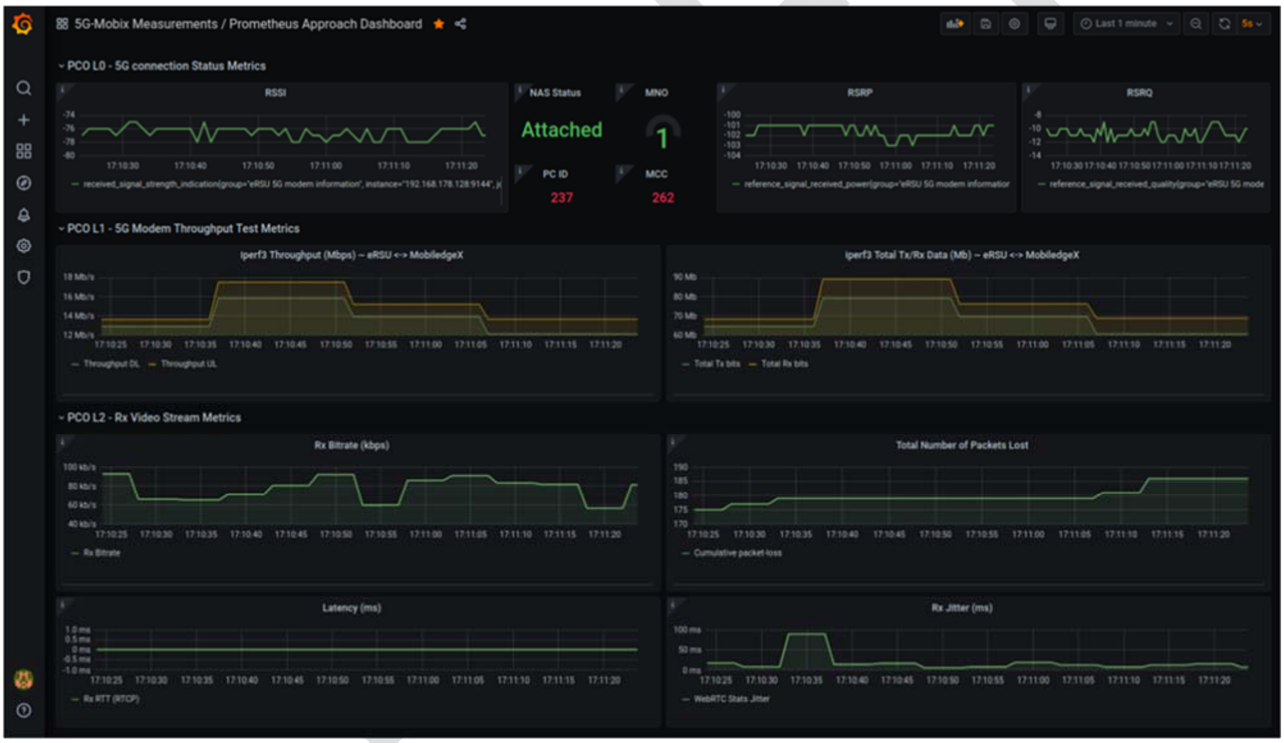

5G-Mobix Measurements / Prometheus Approach Dashboard
Last 1 minute
5s
PCO L0 - 5G connection Status Metrics
RSSI
-74
-76
-78
-80
received_signal_strength_indication{group="eRSU 5G modem information", instance="192.168.178.128:9144",
NAS Status
Attached
MNO
1
PC ID
237
MCC
262
RSRP
-100
-101
-102
-103
-104
reference_signal_received_power{group="eRSU 5G modem informatior
RSRQ
-8
-10
-12
-14
reference_signal_received_quality{group="eRSU 5G mode
PCO L1 - 5G Modem Throughput Test Metrics
iperf3 Throughput (Mbps) – eRSU <-> MobiledgeX
18 Mb/s
16 Mb/s
14 Mb/s
12 Mb/s
Throughput DL
Throughput UL
iperf3 Total Tx/Rx Data (Mb) – eRSU <-> MobiledgeX
90 Mb
80 Mb
70 Mb
60 Mb
Total Tx bits
Total Rx bits
PCO L2 - Rx Video Stream Metrics
Rx Bitrate (kbps)
100 kb/s
80 kb/s
60 kb/s
40 kb/s
Rx Bitrate
Total Number of Packets Lost
190
185
180
175
170
Cumulative packet-loss
Latency (ms)
1.0 ms
0.5 ms
0 ms
-0.5 ms
-1.0 ms
Rx RTT (RTCP)
Rx Jitter (ms)
100 ms
50 ms
0 ms
WebRTC Stats Jitter
17:10:25
17:10:30
17:10:35
17:10:40
17:10:45
17:10:50
17:10:55
17:11:00
17:11:05
17:11:10
17:11:15
17:11:20

Figure 71: Monitoring framework dashboard with 3 levels of control and observation

10.1.4. Platform and Apps Evolution

This section describes the platform and applications developed for the two distinct DE TS use cases.

eRSU-assisted Platooning

Figure 72 presents the main components involved in the DE-TS platooning US, which are vehicles, eRSUs and MECs. These components can communicate with each other through the 5G network via Uu interface using 5G modems, or directly via C-V2X PC5 interface using their integrated Cohda Wireless MK6c EVK units. In the DE-TS, four Cohda MK6c units were deployed in total, two of which are used as part of OBUs in the DE TS vehicles, and the remaining two being co-located with an eRSU on two distinct locations along the DE trial site track.

Scenarios of eRSU-assisted Platooning

There are two main messages flows considered in this user story, the platooning flow and the EDM messages flow. These flows performance are evaluated using different scenarios proposed by the DE-TS which involve the messages exchange via the two interfaces available with different configurations. In these scenarios, communication can be full PC5, full Uu, or hybrid (Uu + PC5). In the hybrid scenario, the Uu interface is used for the transmission of EDM messages through the MEC broker and the PC5 interface is used for the exchange of platooning messages and the broadcast of EDM service announcement messages. Regarding roaming between two V2X domains, in this user story the dual-modem/dual-SIM solution is used. The OBUs have two 5G modems attached, each of them connecting to a different MNO using their SIM. The MNOs are Deutsche Telekom (DT), O2 and TUB 5G SA network.

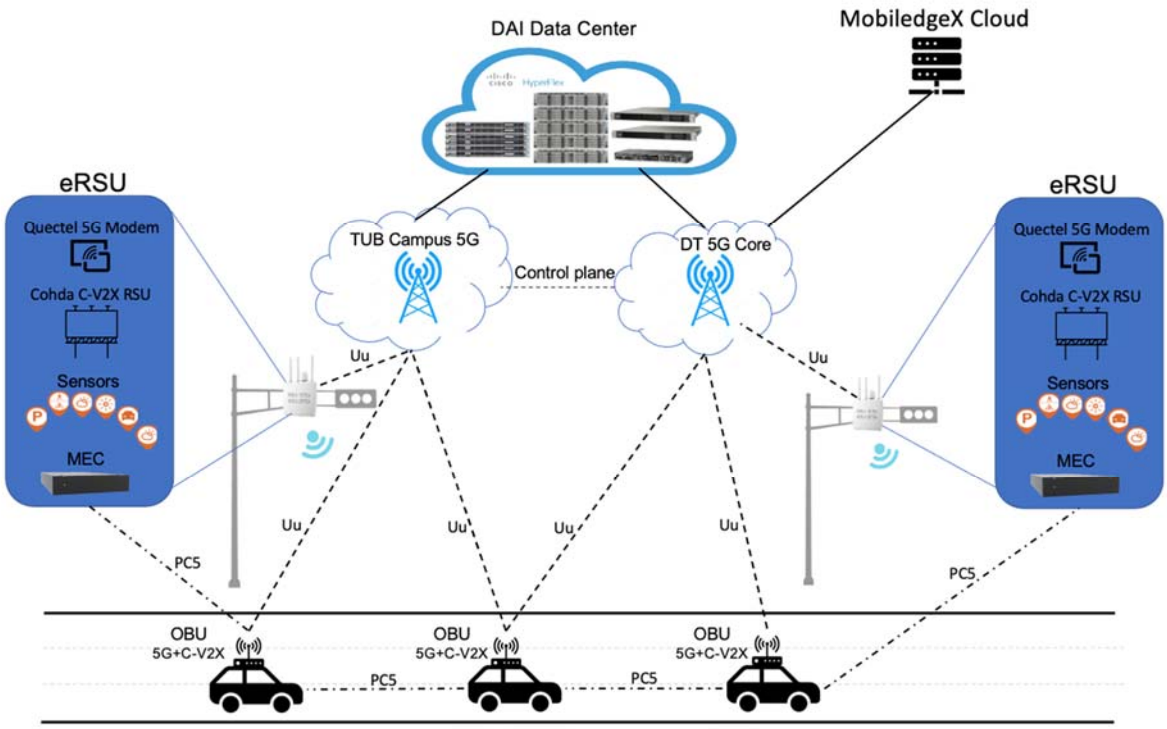


Figure 72: Overview on the roadside and cloud infrastructure of DE TS for the platooning US

In the platooning flow, with the goal of managing the platoon, custom platooning messages are exchanged between leader and follower vehicle via Uu or PC5, depending on the scenario. To exchange platooning messages via PC5, there is an application running in the on-board computers from each vehicle that forwards the right platooning messages to the Cohda MK6c unit attached. The Cohda unit broadcasts these messages so that vehicles close by can receive and process them.

In the case where platooning messages are sent via Uu, vehicles publish/subscribe to the MQTT brokers deployed in the two available MECs, which are MobiledgeX (DT) and DAI Data Center (O2 or TUB Standalone). In this case, the vehicle's on-board computer uses the 5G modem attached to it for communication. In Figure 73 the full workflow for the hybrid scenario is presented.

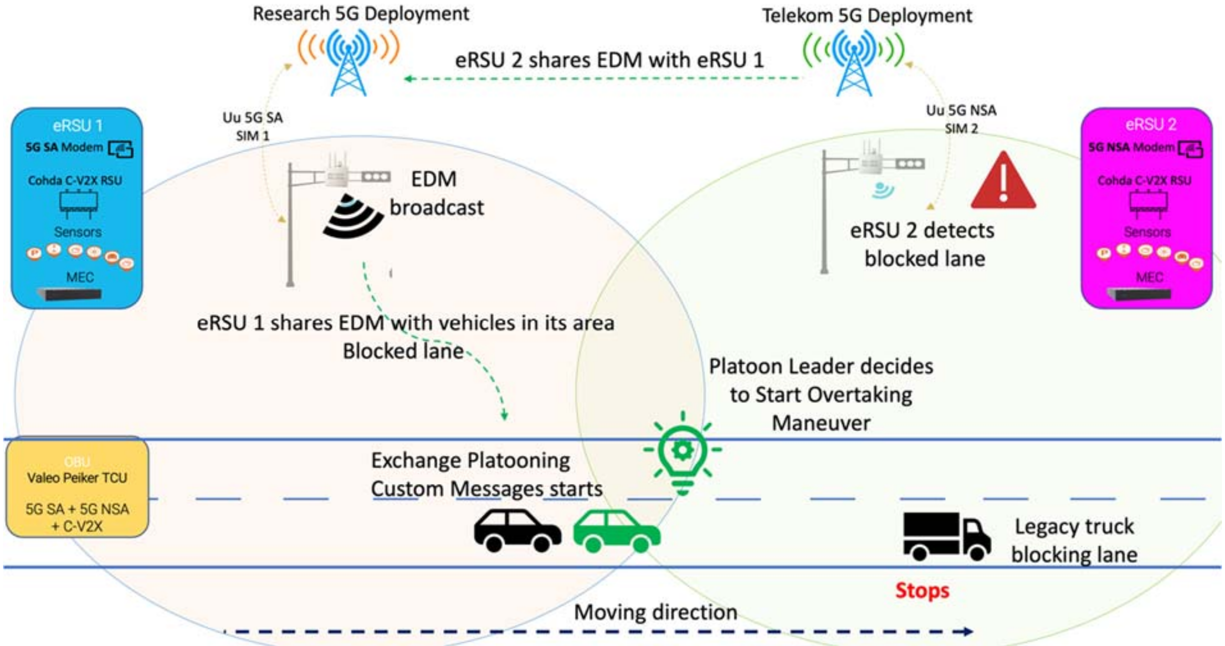


Figure 73: e-RSU assisted platooning messages flow and components

As described in previous deliverables, the eRSU-assisted Platooning consists of the EDM-Service assisting the platooning leader while it leads a platoon along the urban corridor of the DE TS. The EDM-Service is supported by traffic light sensors, vehicle traffic sensors, weather sensors, and the most relevant in the user story, camera-based sensors, which provide an object detection mechanism from the video streams. This object detection mechanism is hosted directly at the far edge MEC infrastructure co-located with the eRSU, which allows for very low latency in producing Cooperative Perception Messages (CPMs), which are then broadcasted to the CCAM-enabled vehicles. A screenshot showing the traffic situation captured from the camera sensor located at the TUB main building is depicted in Figure 74. In this screenshot one can see the outcome of the object detection module with different bounding boxes and characteristics:

- A yellow bounding box corresponds to a person.
- A green bounding box corresponds to a car or truck.
- Red numbers denote the tracking ID of the recognized object, which denotes that the object has been recognized to be the same object in several consecutive frames.
- Red lines show the tracking history of vehicles and indicate the starting point when a vehicle has been initially tracked.

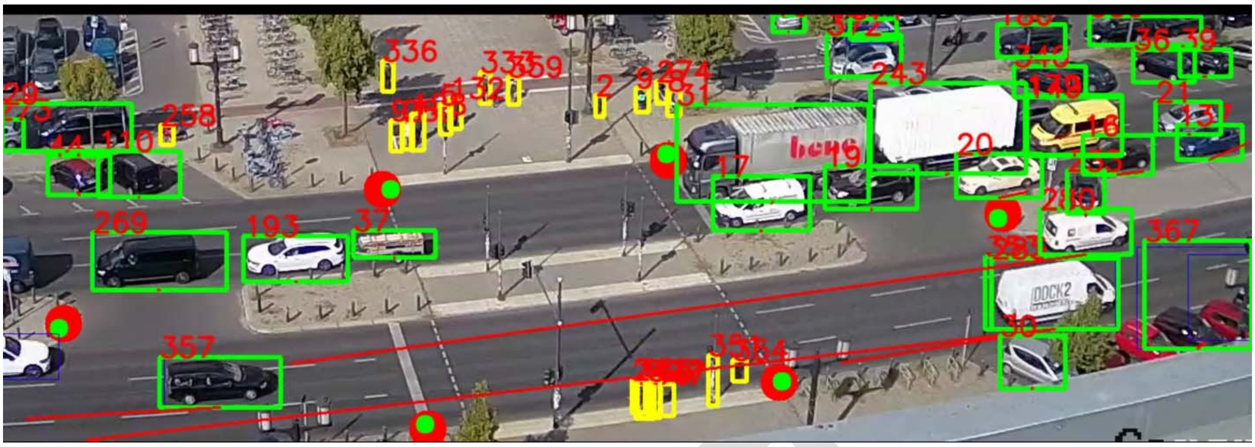


Figure 74: Screenshot from object detection on camera sensor frame captured at TU main building – EDM Service

Not shown in Figure 74 is the speed and direction of the detected objects, which is also available as an output, but is not included in the overlay, as the visualization would be too crowded to be useful when including these details. Furthermore, it should be noted that, in principle, the different classes of vehicles, like buses, trucks, or normal cars could be distinguished from one another, however sometimes the camera angle prevents the successful classification of the object and this in turn then confuses the tracking mechanism. For this reason, the tracking mechanism is configured to detecting cars only.

In this user story, a dedicated object detection application has been implemented to detect if a vehicle stopped in the middle of the road, blocking the lane where the platoon leader is driving. In this case, an event notification message is generated indicating that the lane is blocked by a vehicle including its coordinates, so that the platoon leader can assess the situation and plan an overtaking maneuver. The object detection application is visualized in Figure 75, showing the detected stationary vehicle on the lane marked with the blue spot.

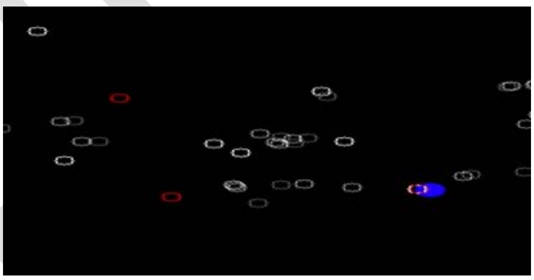


Figure 75: Stationary vehicle detection application

Summarizing what was explained above, in the EDM flow different types of ITS messages are involved and transmitted to the vehicles, among them are traffic light status (SPATEM), object detection (CPM) or event notification (DENM) messages. As for the platooning flow, these messages can be transmitted via Uu or via PC5 interface, depending on the tested scenario.

For EDM messages transmission via PC5 interface, the eRSU is in charge of broadcasting the messages and the platooning leader vehicle will receive and process them, each using their integrated Cohda MK6c unit. Regarding the eRSU, TUB has implemented a Cohda connector which receives the ITS messages from both Cohda boxes, MK6C (PC5), and also implements their own unified parsing of the standardized message formats. By doing so TUB's implementation is vendor-independent and the received messages can be handled in the available resources of the co-located MEC edge PC. The Cohda boxes are used also for measurements of C-V2X applications, for which an own parser of transmitted PC5 messages has been implemented.

MEC broker application for platooning

For the transmission of EDM messages via Uu interface, similar as in the platooning flow, the messages are exchanged in a publish/subscribe fashion using the MQTT brokers deployed in the two MECs. Before the platoon leader can receive any EDM message, it needs to know the address of the broker, which is broadcasted by the eRSU via PC5 in a service announcement message. Once the platoon leader receives this message for the first time, its MQTT client connects to the MQTT broker and it's then ready to receive EDM messages. In this scenario, the broker only forwards the messages that are relevant for the platoon leader regarding its actual position and heading parameters. The broker "knows" from which area the messages come as it knows the location of the sensors producing this information. To filter the messages that are not relevant for the platoon leader, the broker also needs to know its actual position and heading. The platoon leader shares its current position and heading by sending each second a custom "status" MQTT message to the broker. This is how the geobroker feature is implemented.

Overview on developed components for eRSU-assisted Platooning

Platooning application

In order to implement the communication between the platooning leader and platooning follower, and from vehicles to eRSUs either directly via PC5 or indirectly via 5G Uu, we have developed a flexible application implemented using Python and multi-threading. This application is deployed on the On-board PC of the vehicles and implements the custom Platooning Control Messages (PCMs) in alignment with the protocol messages defined within the scope of the Ensemble project[3]. The platooning leader sends out CAM

[3] https://platooningensemble.eu/

messages with isJoinable flag set. The platooning follower receives the message and requests to join the platoon. When the PCM handshake is complete, the platooning leader and follower continue as a platoon.

e-RSU assistance GUI

The in-vehicle visualization of EDM messages generated by the eRSUs has been implemented to display the dynamic situation perceived from the roadside infrastructure via ITS messages. As can be seen also in Figure 76 it displays a map for navigation/overview with the vehicles own position and driving direction on map and the V2X messages with location reference. The following message types are represented by the GUI:

- DENM – shown as an icon corresponding to the message on the map + "warning summary" area. (based on distance/driving direction).
- CAM – position(s) of the V2X participants, (vehicles and ITS stations).
- MAP and SPAT – understand traffic light data (lane assignment) and show current phase.
- CPM – show detected vehicles from V2X (mainly infrastructure).

The technologies used to implement the GUI include a python server for the backend, a webfrontend implemented in html/java-script, mapbox-js to render the map including all the available information that provides a position (lat/long). Furthermore, we used bottle, a minimalistic python webserver library to deliver html and java-script to the client (webbrowser), as well as a REST/JSON-API for the communication with the WebFrontend.

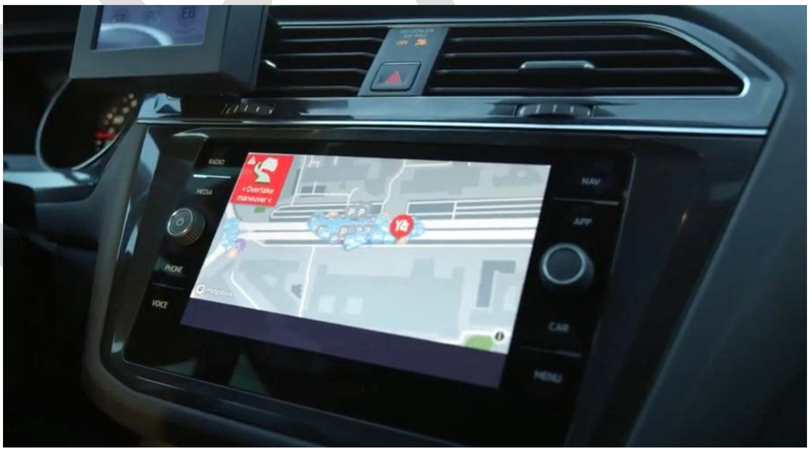
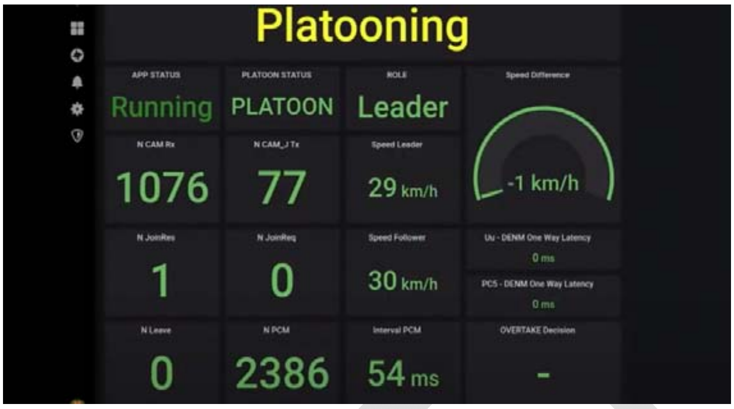


Figure 76: In-vehicle GUI showing the eRSU-assistance

Dashboards to display platoon and 5G network status

While the e-RSU assistance GUI shows the EDM messages, the platooning dashboard provides a visual representation of the platoon's status (i.e., the vehicles role, speed, CAM message reception and relevant statistics). The dashboard for platooning is shown in Figure 77 below and has been implemented utilizing Grafana.



Platooning

APP STATUS: Running
PLATOON STATUS: PLATOON
ROLE: Leader
Speed Difference: -1 km/h
N CAM Rx: 1076
N CAM_J Tx: 77
Speed Leader: 29 km/h
N JoinRes: 1
N JoinReq: 0
Speed Follower: 30 km/h
Uu - DENM One Way Latency: 0 ms
PC5 - DENM One Way Latency: 0 ms
N Leave: 0
N PCM: 2386
Interval PCM: 54 ms
OVERTAKE Decision: -

Figure 77: Platooning Dashboard

In addition to the platooning dashboard we have created a dashboard that visualizes the 5G network status (i.e., the current provider and MEC broker subscription for EDM messages) and provides an overview on the number of EDM messages received via the MEC and their latency.

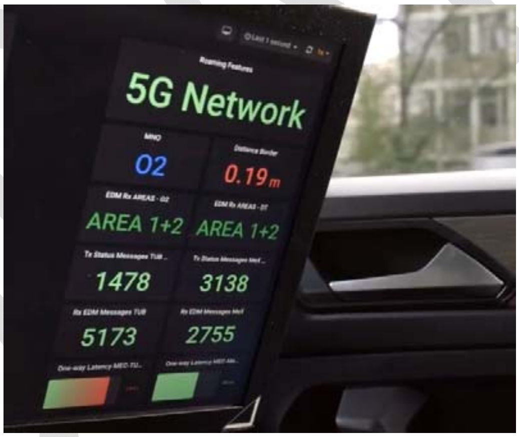


Figure 78: 5G network status dashboard

EDM-enabled extended sensors with surround view generation

This section describes the application developed for the EDM-enabled extended sensors use case with surround view generation. The main components of this use case are systems listed below and depicted in Figure 79.

- Systems on-board the Vehicles including sensors, OBU running CAM functions and two 5G modems to facilitate a quick mobile network *operator change for targeted conditions.*

- *RSUs with various environmental sensors and IP camera to provide a wider and trusted view.*

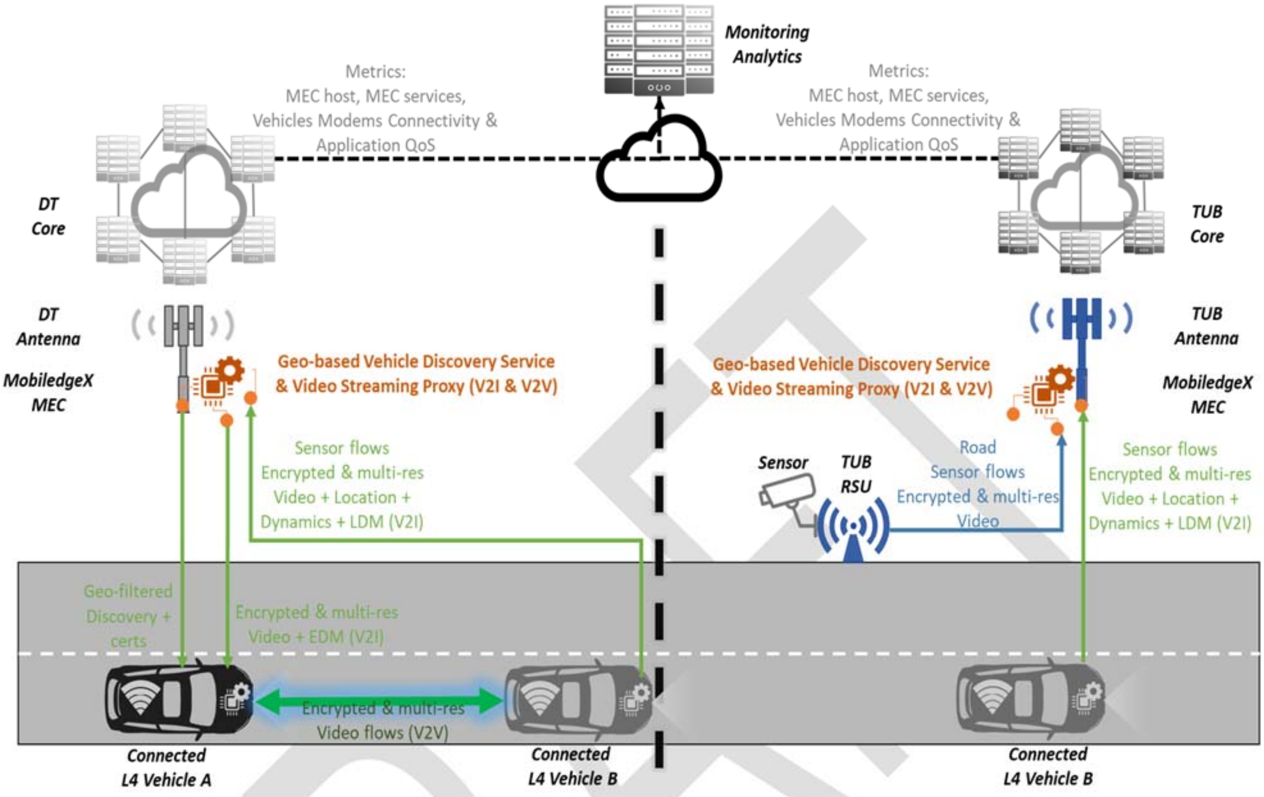


Figure 79: Overview on the Vehicle Discovery and signaling for V2I and V2V

- Base Stations coming from Deutsche Telekom, O2 and TUB.
- MEC infrastructure provided by MobiledgeX from Deutsche Telekom ready to be locally operated serving services such as:
  - the Vehicle Discovery Service and
  - the WebRTC Video Streaming proxy
- Cloud service for the Monitoring and Analytics system gathering all the captured metrics and logs and connected to the 5G-MOBIX testing and logs infrastructure.

All these systems, depicted in Figure 79, play an active role on the EDM-enabled extended sensors use case with surround view generation.

Introduction on Surroundview concept

Automotive surround view is a video-based driving assistance function, in which a model of the ego-car and the vehicle's surrounding are displayed on a monitor within the vehicle. The scene can either be displayed as a virtual bird-view, emulating the perspective from an elevated point a few meters above the vehicle, or as a freely adjustable viewing direction pointing outwards from the vehicle center. Thereby, the direction may either be chosen by the operator or may be selected by an underlying ADAS software.

The key component of the surround view camera system are the fisheye objectives which cover a field-of-view of up to 180 degree, in comparison, a conventional front camera covers 50-60 degree. The surround view system used in both the Valeo and Vicomtech test vehicles consists of four Valeo fisheye cameras (1280x800 30fps) that are mounted around the vehicle (one camera in front, one camera at the rear, and two cameras are mounted to the side rear-view mirrors). The camera images are forwarded to the onboard processing unit (including a GPU), which renders the surround view via coherent image stitching. Valeo developed a Surround View Application based in Computer Vision algorithms as shown in the RTMaps diagram from Figure 80.

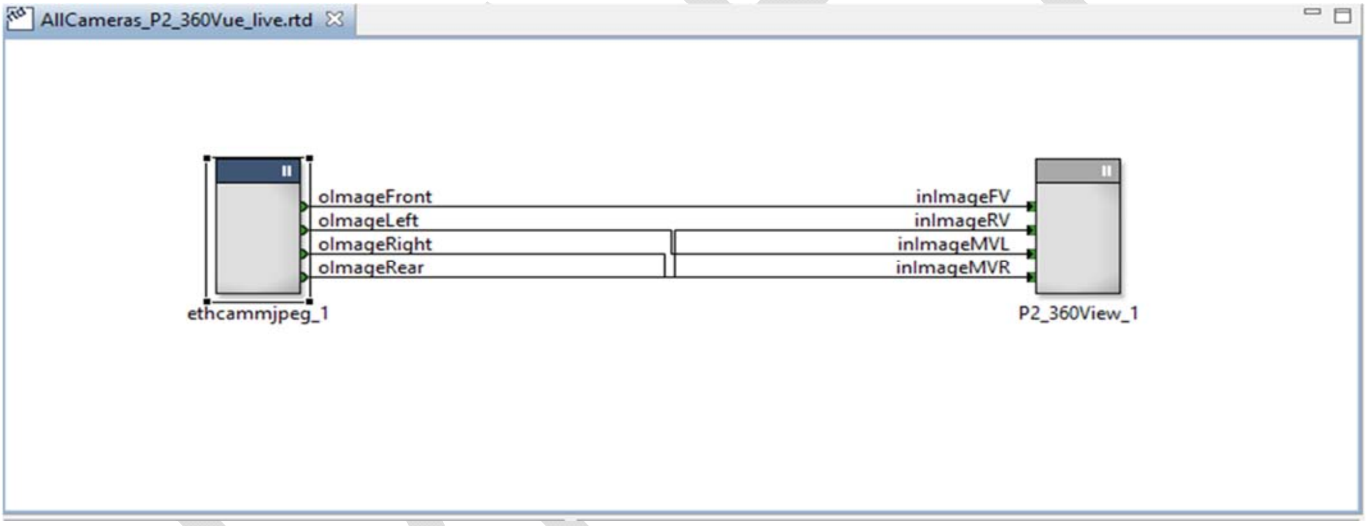


Figure 80: RTMaps diagram of the 360° SurroundView

One of the fundamental problems in the 360° surround view synthesis is the geometric alignment of the four fisheye images. To overcome this problem, we need to calibrate the cameras. Once the calibration is done, we can compute the perspective transformation from the four fisheye camera images to a virtual bird-view. The calibration is done off-line, saved into a configuration XML file, and fed as a property into the RTMaps component (P2_360View_1 in the Figure 56). Both Vicomtech and Valeo test vehicles are equipped with Valeo fisheye cameras, so the calibration procedure is the same.

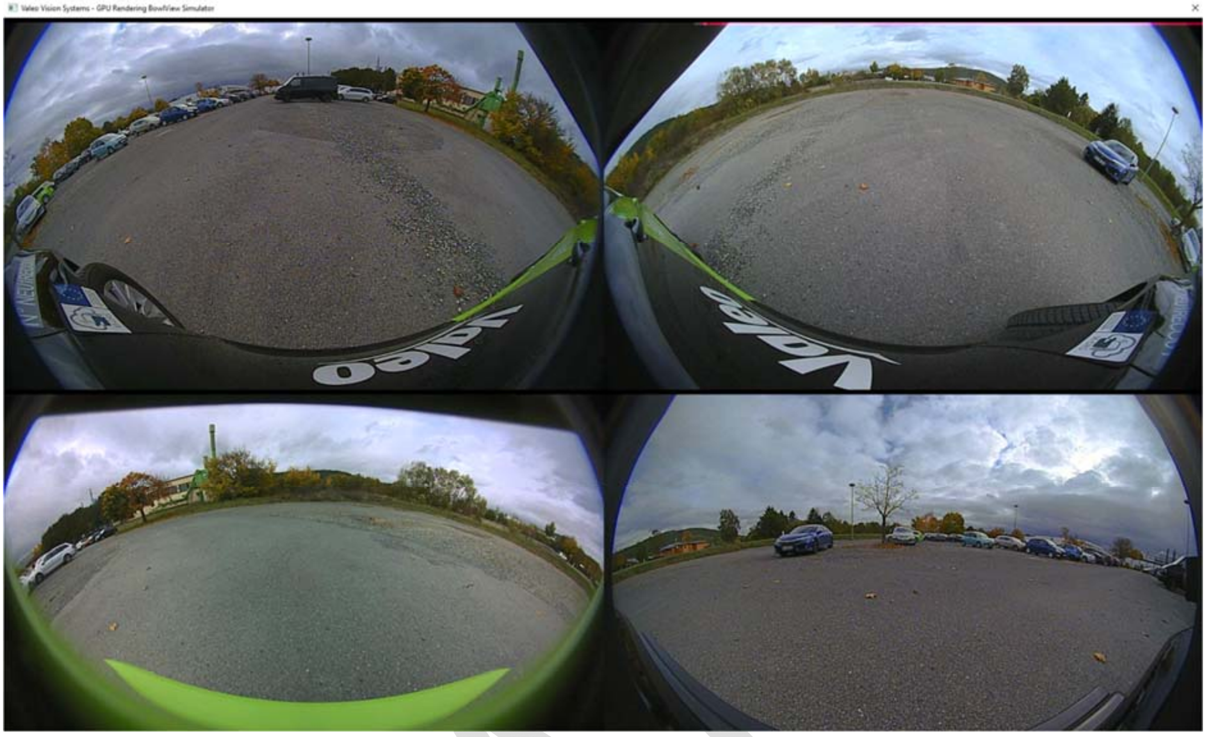


Figure 81: Four raw fisheye images of the Valeo vehicle

Assume that the surround view application is installed on the onboard computer of the Vicomtech vehicle, and the user wants to see the surround view image of the camera system in the Valeo vehicle. The four fisheye images from the Valeo vehicle are sent to the Vicomtech vehicle in real-time via the 5G network. Then, the user can see either raw images, as presented in Figure 81, or the processed bird-view image, showed in Figure 82, on the onboard display of the Vicomtech vehicle. The pose (position and orientation) as well as the zoom (field-of-view) of the virtual bird-view camera image can be selected freely.

Technical specs: The image resolution of the Valeo fisheye cameras is 1280 x 800 pixels. The images are produced with framerate about 30 frames per second. At 32 bit per pixel, this results in a data stream of almost 500 MB/s of raw images that are handled by the surround view application. The Valeo implementation supports real-time rendering on the on-board computer used in the Valeo vehicle.

Vehicle Discovery Service

The monitoring analytics collects metrics from all the actors involved in the user story in order to allow live monitoring of the status, forensics and support decision making. Here, the different corners included, the infrastructure metrics such as CPU, RAM usage, throughput, etc., the vehicles connectivity metrics, such as bandwidth or latency and the video streaming metrics such as latency, jitter and bitrate provide a holistic context of the ongoing communications. Then, the edge services to enable surround vehicle/RSU discovery and the video streaming proxy allow to automatically identify a relevant sensor from vehicle dynamics and network performance to establish a video streaming session in two communication modes, V2V and V2I.

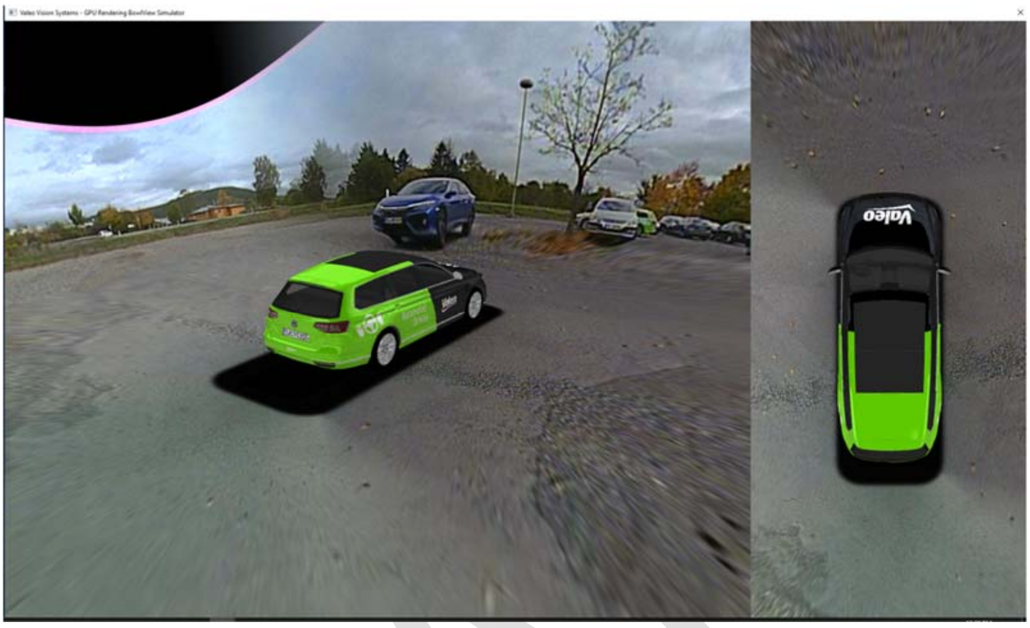


Figure 82: Resulting surround view image on the display of the Vicomtech vehicle

The V2V communication is established when direct communication between vehicles is possible, then the proxy just bridges the signalling messages and the handshake negotiation for encryption. While in the V2I option the proxy also delivers all the media streams which are pushed to it from origin and delivered from it to destination. As no re-encoding or media transformation is done the processing requirements of the WebRTC proxy are kept simple avoiding the quality loss. The RSU behaves as any other vehicle providing a high perspective for the environment, making it available to any system interested in a wider local view which includes the identification of environmental items, such as pedestrians, parked vehicles or fixed road obstacles. Finally, the vehicles systems collect geo-position from GNSS systems and provides presence and position to the edge Vehicle Discovery Service where the appropriate instance to subscribe in is tied to a broker. The vehicles have two communication options sending the data and video streams directly to the peer vehicle or through edge video stream proxy in the case the destination is not reachable due to networking restrictions.

The Vehicle Discovery Service and the WebRTC proxy has been dockerised and deployed at MobiledgeX at the Berlin Cloudlet.

The Vehicle Discovery Service includes:

- A MQTT server v 1.16 to allow the registering of the different participants, the continuous updates of geo-position based samples and the queries of candidates to provide relevant sensor flows

- An InfluxDB database v1.15 for time series recording the geo-position records for all the vehicles
- A Python v3 algorithm to suggest an appropriate candidate considering vehicles dynamics and network communication delays to provide an optimal candidate at the moment that information could be used by the vehicle performing the request

This Vehicle Discovery Service is using different topics in order to:

- create different communication channels for geo-position updates,
- for querying candidates and
- for notifying video stream session start.

The topics are defined in order to facilitate the service discovery and the migration of sessions across neighbour areas.

The WebRTC Proxy, developed with Python v3, includes SSL certificates to protect connection. It brings two alternatives depending on the setup to test and perform:

- V2V, where the proxy just mange signalling for establishing the communication between the peers
- V2I, where the proxy also serves as a bridge for video streams which are forwarded to the peer

Both systems are lightweight, standard and ready to scale up based on MQTT topics addressing.

The DE TS has included different deployment and enhancement activities required to connect all the involved systems and actors and to make services available and published ready to be discovered and consumed by CCAM applications embodied in the on-board systems of surrounding vehicles.

In order to get all the systems working in place the main activities have pivoted around:

- Video streaming stack, including the signalling to get peers connected and the WebRTC proxy to support V2V and V2I options. In more details this activity includes:
    - Use STUN/TURN and ICE protocols to ensure connectivity beyond NAT network functions and ensuring that IPs and ports behind private networks could be reached by peers.
    - Employ TLS/SSL to perform certificate handshake between peers and encrypting all the data flows; exploit NVIDIA acceleration to encode the video streams leveraging the GPU processing capacity to avoid overloading the onboard system at the vehicle and ensuring a smooth and timely generation of compressed video.

- Enable the WebRTC proxy to and evolve it to support V2I and V2V signalling and communications, requiring the tunnelling of local streams to forward them to the peers.
  - Ship WebRTC proxy in a Docker and deploy in the Edge infrastructure for Deutsche Telekom; embody the WebRTC peers, based on Gstreamer v1.18, in RTMAPS modules facilitating the building of automotive processing pipelines from the camera capture, to the format conversion and sampling.
  - Aggregate QoS logs, captured from RTCP messages, pushed to Prometheus system with the defined period to measure performance and take mobile network operator change decisions.
  - Setup an NTP/PTP synchronization mechanism to get local systems synchronised.
- Vehicle discovery service based on an MQTT IoT stack to facilitate the subscription to the appropriate service at the edge. In more details this activity includes:
  - Establish isolated communication channels protected with SSL certificates where different systems can be enrolled.
  - Empower Vehicle Discovery Service with algorithms to propose a relevant sensor to a vehicle trajectory including the compensation of the shifts produced by network delays and trajectories stored in InfluxDB for the time series with the geo-position of different vehicles.
  - Ship MQTT server in a Docker and deploy in the Edge infrastructure for Deutsche Telekom; embody the Discovery clients of the peers in RTMAPS v4 modules facilitating the building of automotive processing pipelines from the geo-position capture, to its format conversion to VCD or CPM messages and ASN.1 encoding.
  - Aggregate monitoring package to capture logs and performance metrics being pushed to Prometheus.
  - Setup an NTP/PTP synchronization mechanism to get local systems synchronised.
- Surround View driving function, empowered by Computer Vision techniques to produce a 360º view of the vehicle environment, ready to include external video sources. It comprises:
  - Connect Vehicle Discovery Client and WebRTC peer to the Surround View application to allow and automated execution of the extended sensors user story to extend surround view with see-through data.
  - Embody the surround view application in RTMAPS modules facilitating the building of automotive processing pipelines from a set of video flows, recorded, live or streamed.

 - o Aggregate trigger and start-up timestamps pushed to Prometheus system to measure network performance and back mobile network operator change decisions.

- Modem integration on OBU and RSU systems to get Quectel modem working in bands to be used at development, testing and TS areas. It includes:
  - o Installation of Quectel drivers from the source code repositories from Quectel.
  - o Installation of standard Quectel drivers using a Kernel version 5.4.20 from Linux which includes the drivers.
  - o Development of daemon to monitor modem status and activity.

Implementation of the Surround View Application

In order to capture geo-position-based data from sensor devices, an RTMAPS pipeline is employed feeding the Vehicle Discovery Service with 10Hz positions updates including not only geo-position, but also acceleration and heading, as shown in Figure 83.

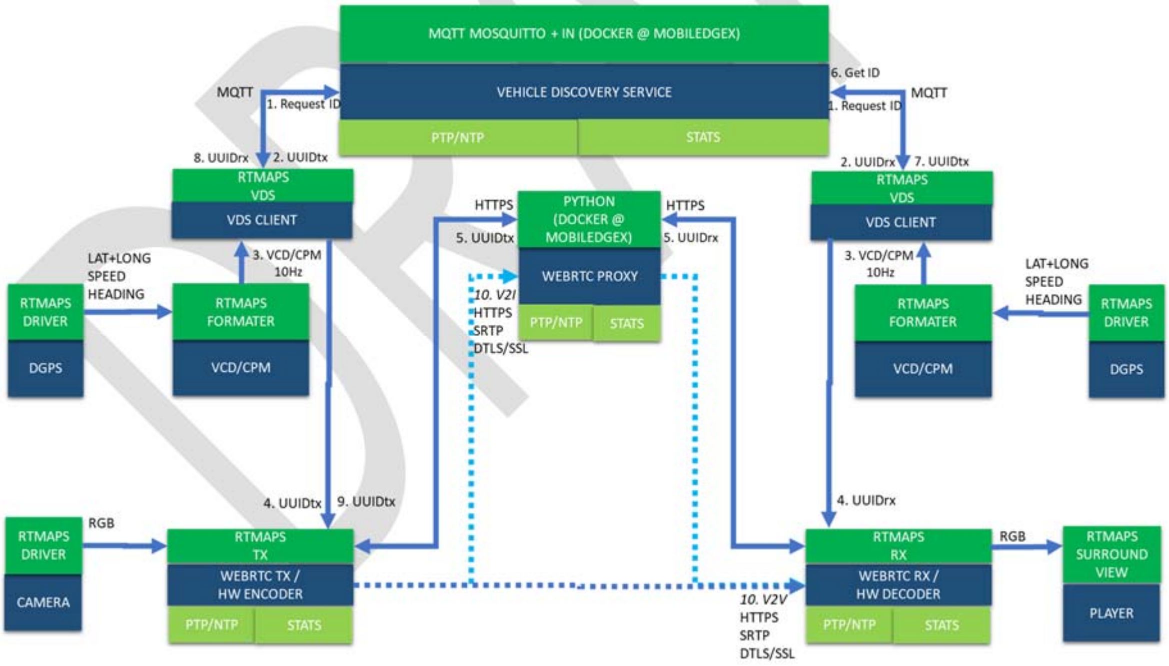


Figure 83: RTMAPS Pipeline in the cars (Transmitter and Receiver) connected to Systems deployed in the Edge

To allow the utilization of a video stream from an external vehicle the calibration is a capital aspect. Thus, the four cameras from VALEO that cover 360º around the vehicle are calibrated with a marker-based setup, as depicted in Figure 84. This process is essential to allow computer vision algorithms to make build a coherent image stitching.

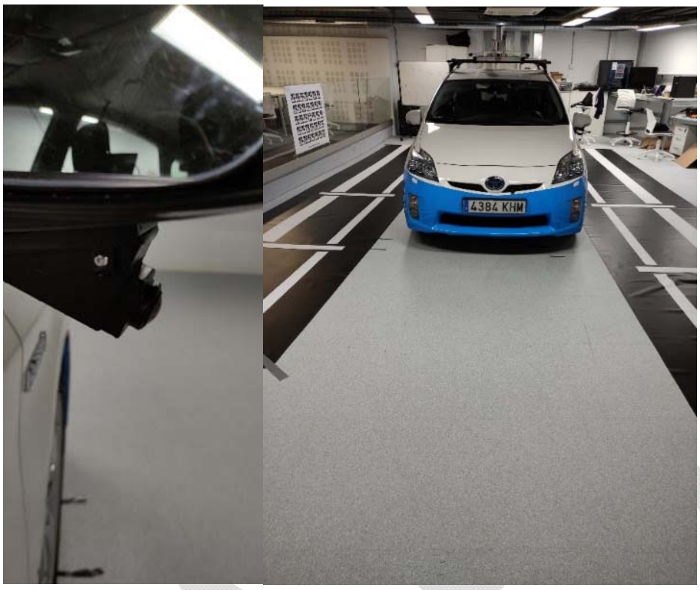


Figure 84: VALEO's Cameras and calibration setup

In terms of RSU cameras, a system has been developed to be able to capture RTSP video sources and turn it into a regular WebRTC flow that any system could consume. It is important to highlight that this transformation does not require transcoding meaning low processing load and a lossless conversion.

Furthermore, the RTMAPS pipeline pushes the geo-position through the Vehicle Discovery Client to keep presence map at the Edge service updated. Then the client gets the vehicle UUID when the vehicle is registered in the Vehicle Discovery Service. These IDs are used to suggest the appropriate external data source to be used by the surround view CCAM application.

Specifically, the Transmitter and Receiving pipelines in RTMAPS are depicted in Figure 85, Figure 86 and Figure 87.

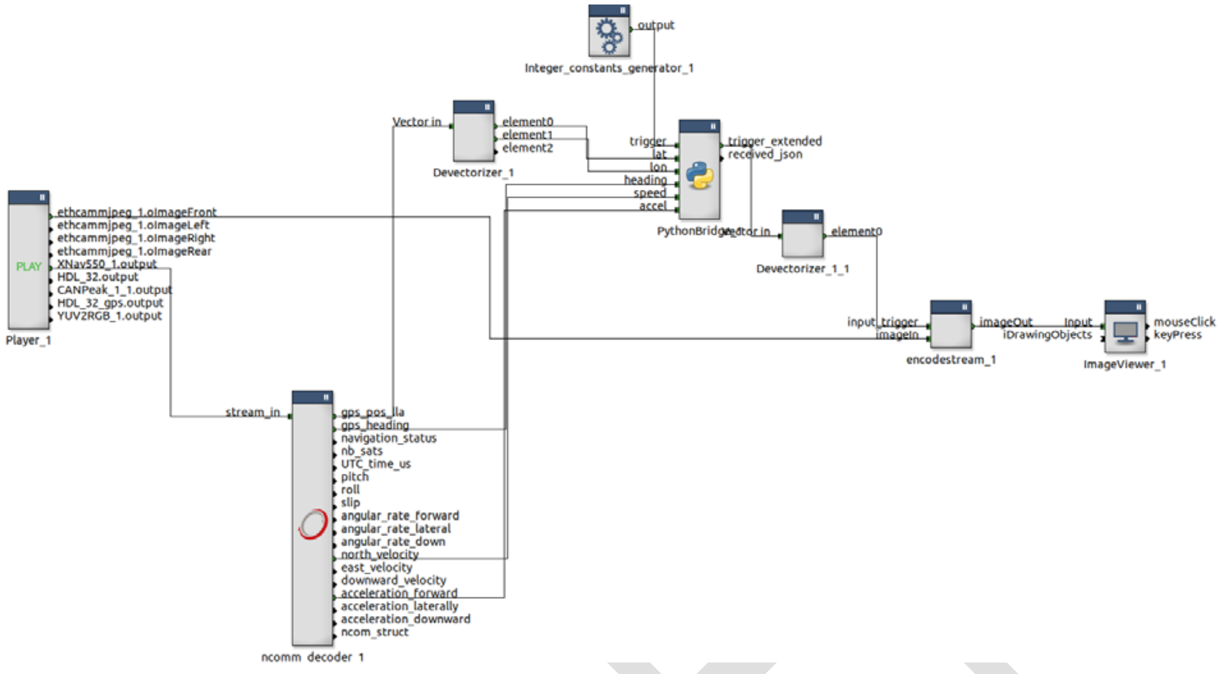


Figure 85: Transmitter pipeline in RTMAPS

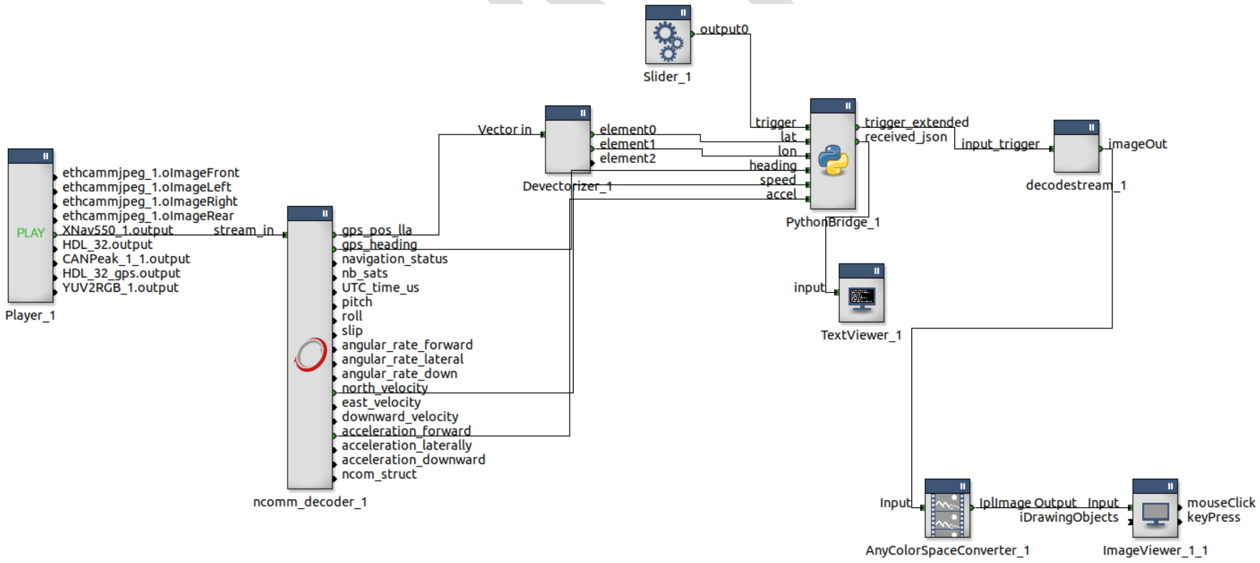


Figure 86: Receiver pipeline in RTMAPS

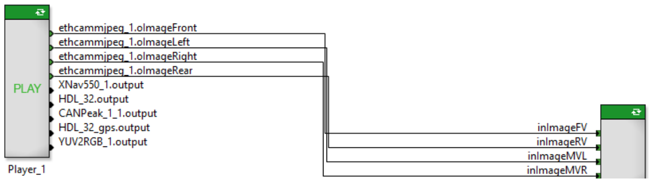


Figure 87: Surround View pipeline in RTMAPS

In order to launch the Discovery Service and generate the surround view with extended sensors, it is necessary to create a video stream session. For that reason, a manual HMI component has been added to start the session. Only then the driver can ask for a surround view covering the blocked field of view as showed in Figure 88.

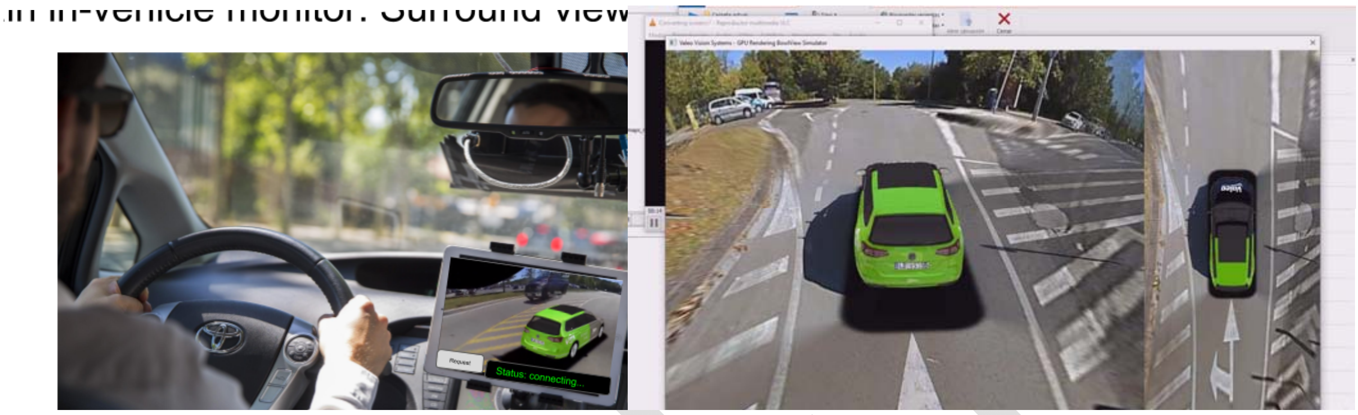


Figure 88: HMI Application for Surround View Generation

Infrastructure Integration and Testing Activities

Several parallel activities regarding the integration and testing of DE TS components have taken place within the scope of T3.4. First and foremost, the extension of the eRSU, amending the edge PC with the Quectel 5G modem, has been thoroughly tested for proper functionality in the Ubuntu Server 20.04 environment. After some necessary firmware update of the RG500Q-EA device, the integration testing in the lab environment at TUB has been concluded successfully. A picture of the integrated platform is provided in Figure 89, not including the outdoor case to protect the setup from weather conditions and rain.

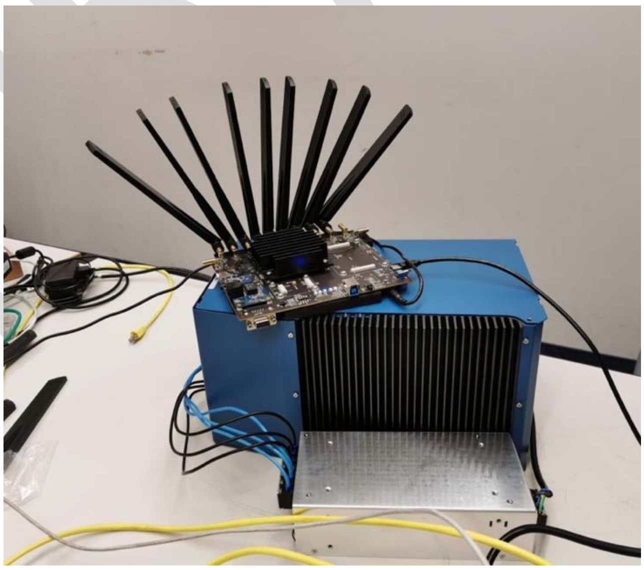


Figure 89: Integration of Edge Server with 5G modem

10.1.5. Future Work

With the application development and hardware deployment activities at the DE TS being completed, the focus of work lies on the full trialling of the use cases (within the scope of WP4) and data collection for WP5. That said, some maintenance tasks remain to keep the DE TS infrastructure operational during the phase 5 activities. Due to the regulatory environment in Germany, the installation of PC5 communication devices requires a license from Bundesnetzagentur when operating the devices on their deployment location at the DE TS, which needs renewal after 12 months, if the regulatory regime does not formally change in this regard.

10.2. Trial Site FI Roll-out

10.2.1. Introduction

The Finland trial site is located within the Otaniemi area of Aalto University campus. This location allows the 5G-MOBIX project leverages 4G/5G testbeds deployed in the Otaniemi area for testing and evaluation of 5G-MOBIX user stories under multi-PLMN deployments recreated to emulate cross-border environments.

10.2.2. FI TS Apps and Functionalities

In this subsection we describe the applications that have been developed to implement the two FI TS user stories, one each for extended sensors and remove driving use case categories.

Applications developed for FI TS Extended Sensors user story

The FI TS extended sensors user story constitutes two developments from AALTO, namely, a crowdsourced HD mapping application and a service discovery system that enables a UE (vehicle) to discover a MEC in a visited network. These developments are described in brief below.

Crowdsourced HD mapping application

The application is composed of two parts:

1. an edge computing system

2. an HD mapping application which runs on the edge computing system.

The architecture of the edge computing system is shown in Figure 90. The coordinator, which is responsible for resource allocation, task allocation, storage service, and service migration, was designed by AALTO. There is one coordinator in a local area, e.g., a city, all the vehicles and MECs inside the area are registered to the coordinator before using the system. During the registration process, a vehicle reports its sensors and the MEC reports its hardware specs to the coordinator. The coordinator relies on a heartbeat to monitor the location of the vehicles and the load of the MEC. The information will be further utilized for resource and task allocation. For example, in the see-through application, the coordinator can assign a vehicle's camera to another vehicle behind it. In the offloading case, the coordinator can assign a MEC of the lowest load to the vehicle for load balancing purposes.

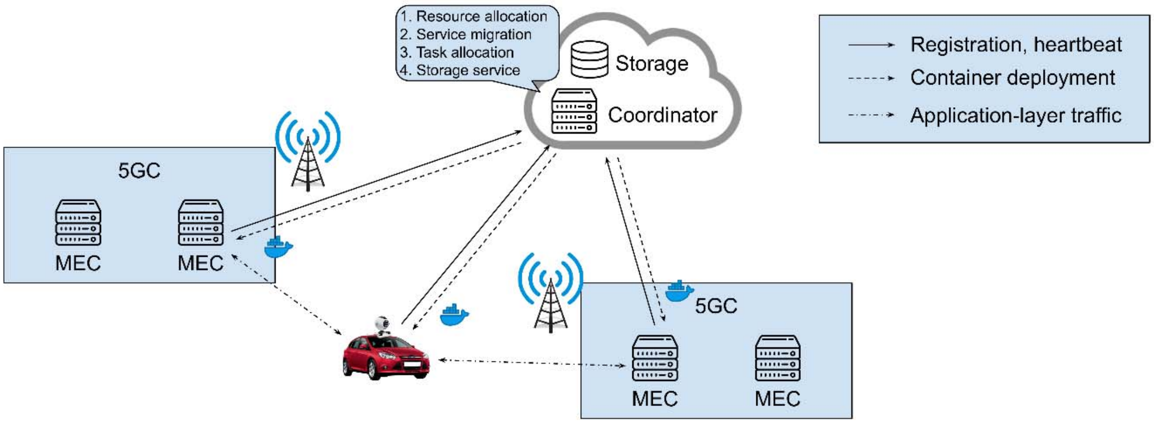


Figure 90: Edge Computing System

For the task execution part, the applications, including video streaming and HD mapping, run on the MEC and listen to vehicle connections. The vehicle is connected to the MEC deployed inside its PLMN. It streams video to the MEC and receives processing results. During the move of the vehicle, it may change PLMN according to the signal quality of available networks. The change of PLMN triggers migration, where the vehicle terminates applications on the old MEC and connects to the new MEC for data processing.

The implementation of the HD Mapping application is based on object detection and vehicle location. The MEC uses the received video streaming to detect the location and the size of traffic signs. Combining with the GPS data reported by the vehicle, the actual location of a detected traffic sign is estimated on the map.

Service Discovery System

The service discovery system is designed to help a UE find the IP address of a MEC in the local network. The system is built on top of DNS to ensure the compatibility with existing applications.

The assumption of the service discovery mechanism is as follows:

1. The UE connects to a MEC via its public (optional)/private IP address.
2. The UE can connect to the MEC via its private IP address if it is inside the same 5GC as the MEC or it is roaming to the MEC's 5GC.
3. Each 5GC has a LDNS which can be modified to support the TAC extension, which is proposed by us as an EDNS record.

The architecture of the service discovery system is shown in Figure 91. The MECs register themselves in a cloud-located coordinator, which instructs the DNS server on how to reply to different domain names. The LDNS deployed at the 5GC is a resolver for relaying and caching the DNS queries in the local network. Its job is to accelerate the DNS query process and the response is purely decided by the cloud-located DNS server.

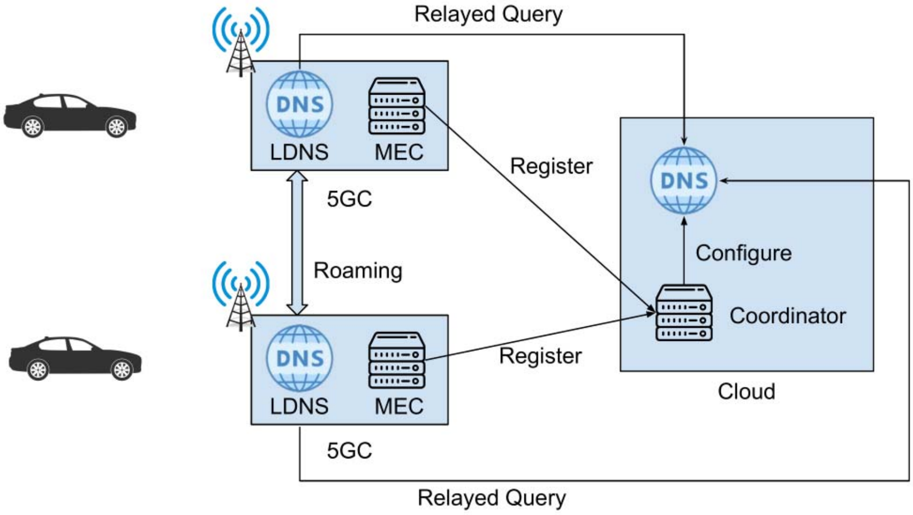


Figure 91: Service Discovery Architecture

The UE can get the IP address of the MEC inside the same 5GC by sending a DNS query to the LDNS, which is further relayed to the cloud-based DNS server and embeds the TAC into the DNS query. The DNS server matches the DNS query's embedded TAC code with the registration information of the MECs to find which MEC is inside the same 5GC as the UE. Then, the private IP address and the public IP address (if applicable) of the MEC are returned to the UE.

The service discovery mechanism supports service migration. The UE needs to use a DNS query to find the IP of the MEC whenever its connected gNB changed or its IP address is changed. If the newly found MEC is different from the previous one, the UE can migrate from the old MEC to the new one. During the stage of migration, the UE has access to both the old and the new MECs so that it can implement application-specific logic for data transfer, e.g., move its user data from the old MEC to the new MEC.

An extension of the service discovery mechanism is designed for the cross-border user cases, where deployment at the edge is not possible due to telecom constraints. In this case, the cloud DNS will use the UE's source IP address to identify its location because we are not allowed to modify LDNS and embed TAC in the DNS query. The localization accuracy of this extension is not as good as the original solution because matching an IP address with its geographical address is not an easy job. However, the accuracy is sufficient in CBC as we only need to know which country the IP is located at.

Applications developed for FI TS Remote Driving user story

The FI TS remote driving user story integrates two main applications. These are the primary remote driving applications that run between the vehicle and the Remote Operations Center and the LEVIS video streaming application. A brief description of these two developed applications and their interaction in the use story is provided below.

Remote driving application

The Remote Driving user story follows the general communications architecture of Sensible 4, where the vehicle connects to a fleet control server through a network connection as shown in Figure 92. This has been previously done with 4G, and now the 5G connection capabilities are tested. The vehicle gathers data, such as position, obstacles, operations, etc, that need to be presented to the remote assistance operators in a format they can quickly and safely make recommendations to the route in case anomalies arise and the vehicle cannot handle itself. From 5G-MOBIX perspective, the Remote Operations Center (ROC) and Fleet Control Server are physically located in the same computer, as within the project scope there is no need for separation of the two. The LEVIS server operates in parallel to the Sensible 4 architecture.

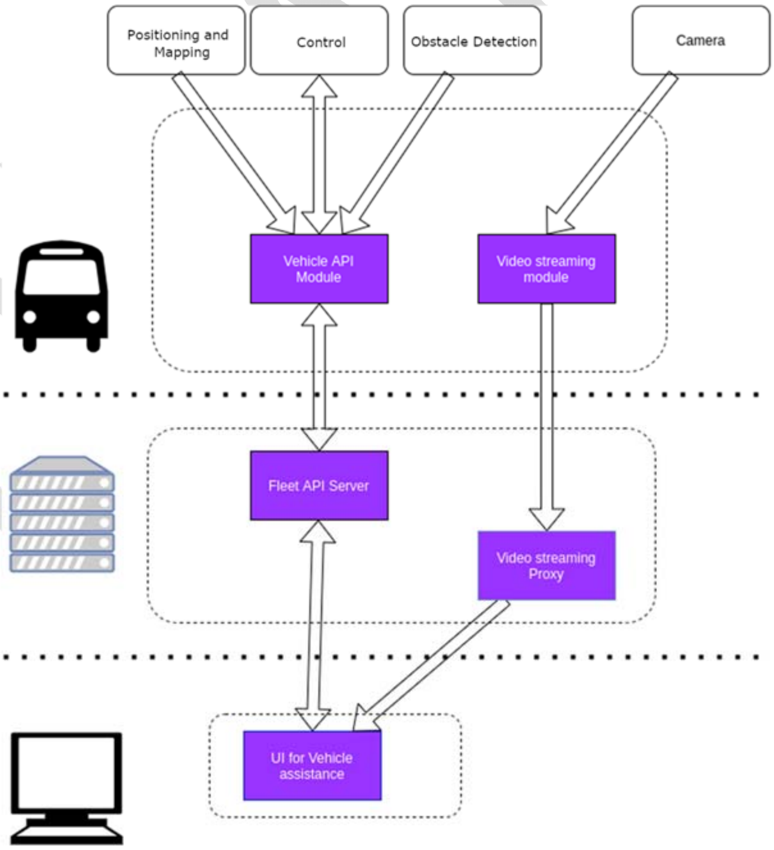


Figure 92: Remote Driving Architecture

All the individual pieces are in place and operational. The UI for vehicle assistance is at early development stages with a proof-of-concept UI completed (see Figure 93). Permission to act on a detected obstacle can be given, camera data can be viewed and Obstacle Detection data can be viewed. Localization on a map of the area can be observed through the UI.

The vehicle identifies that there is an obstacle in the predefined path and can request assistance from the operator. Through data sent by the vehicle to the ROC, the operator can then determine the position of the vehicle on the map of the area, see the camera streams, identify detected obstacles, observe the reference route and alternative routes, and Accept/Reject alternative route.

The system has been extensively tested in the CARLA simulator, and first vehicle tests were conducted in Sensible 4 test drive environment with a 4G connection.

For 5G-MOBIX needs, the UI needs to be capable to present also LiDAR data in a visual format understandable to the operator. The development of feature was completed and tested in time for the remote driving early trials with 5G connectivity between vehicle and ROC.

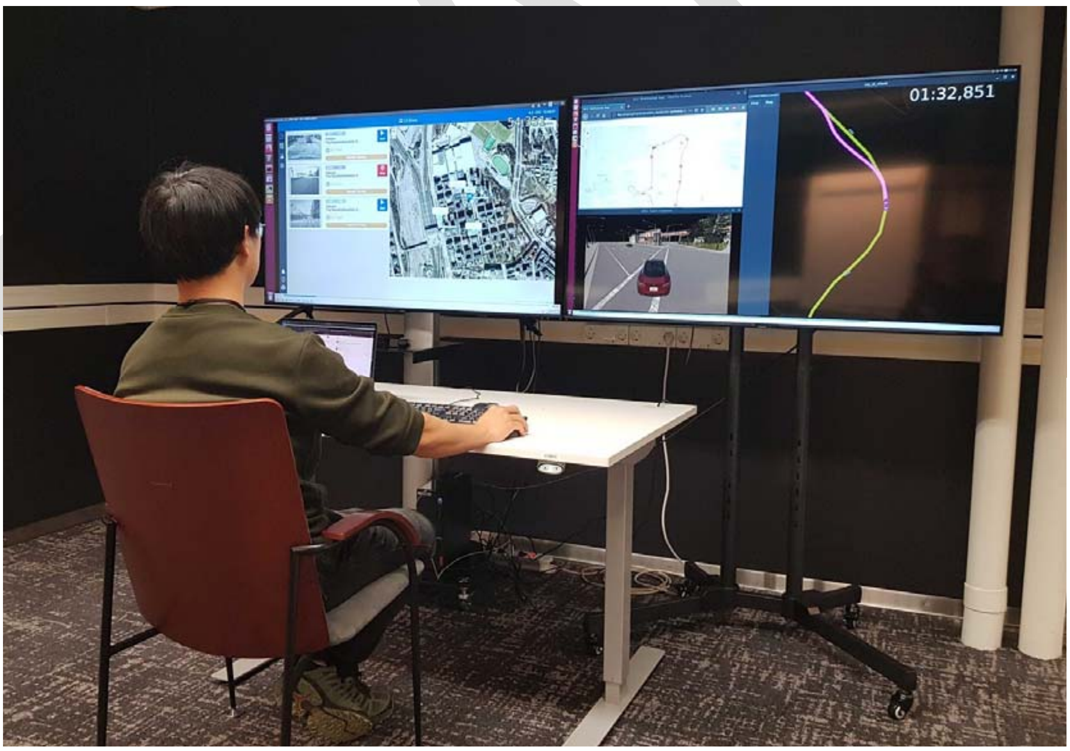


Figure 93: Remote Operator at work

LEVIS application

LEVIS application is a platform developed by AALTO that enables low latency cloud-based video streaming between two ends, namely sender and receiver. The design goals of the developed platform are two-folds: i) ensure a very low end-to-end (E2E) latency between the stream sender and receiver, ii) ensure a smooth and short streaming outage (i.e., close to the connectivity outage time) during handover operation. The deployment scenario for Finland's remote driving user story is depicted in Figure 94. From this figure, the following components can be identified:

- **Autonomous Vehicle (AV):** it represents the video stream sender. On the vehicle, there is a Single Onboard Computer (SOC), which consists of a Jetson Xavier, with an attached camera. Both the Jetson Xavier and the AV are connected to outside world through the same OBU.
- **Streaming Server:** it is located at the cloud (AALTO University data center). It receives the stream from the AV and makes it available to some consumers.
- **Remote Operation Centre (ROC):** described in previous Section.

Initially, the AV drives autonomously and a local video recording is started when the vehicle starts moving. When an obstacle is encountered, the vehicle stops and triggers a remote driving request to the ROC as well as a live streaming request directed to the Jetson Xavier. Upon request reception, the Xavier sends the last 15 seconds recorded locally, in parallel it starts sending the live stream to the LEVIS server. Once the LEVIS server starts receiving the live stream from the Xavier, it calls some APIs deployed at the ROC to promptly open two different players to the Remote Human Operator (RHO), where one displays the last 15 seconds recorded before the obstacle, and the other displays the live stream coming from the vehicle. Based on the video feedback, the RHO formulates some commands to be sent to vehicle. It should be noted here that a VPN connection is established between the vehicle and the ROC to secure the exchanged messages and commands.

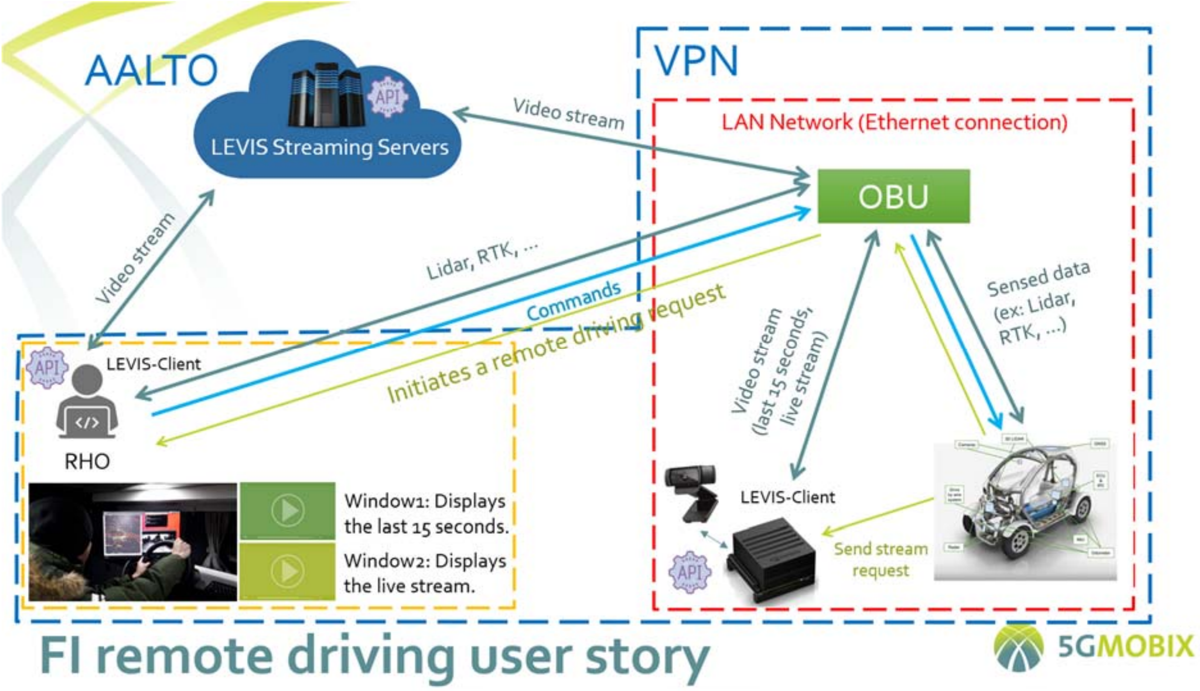

AALTO
API
LEVIS Streaming Servers
VPN
LAN Network (Ethernet connection)
Video stream
OBU
Video stream
Lidar, RTK, ...
Commands
Initiates a remote driving request
Video stream (last 15 seconds, live stream)
Sensed data (ex: Lidar, RTK, ...)
API
LEVIS-Client
RHO
Window1: Displays the last 15 seconds.
Window2: Displays the live stream.
LEVIS-Client
API
Send stream request
FI remote driving user story
5GMOBIX

Figure 94: FI TS remote driving user story deployment scenario

In order to fulfil the aforementioned design goals, a modular architecture, illustrated in Figure 95, has been proposed and implemented for the LEVIS system. In this figure, three separate entities can be identified, namely the sender, the receiver and the server. At each entity, there is a pipeline composed of a set of interconnected blocks, where each block handles a specific task, such as, capturing, encoding and pay-loading. Hardware acceleration is used to reduce the E2E latency, notably at encoding phase, using the GPU of the Jetson Xavier Single SOC. For each pipeline, a recovery module is deployed that keeps watching the status of the whole pipeline and reconstructs it in case of failure. It is worth noting that V2X communication is employed, so the video is sent and retrieved from the cloud, by the sender and the receiver, respectively. Upon stream reception at the cloud server, the server, in its turn, exposes the coming stream over RTSP server, and the receiver establishes RTSP session with the server to start receiving the stream. It should be noted here that the communication between the sender and the server is stateless. Such that, if a handover operation occurs, the stream resumes as soon as the connection is back.

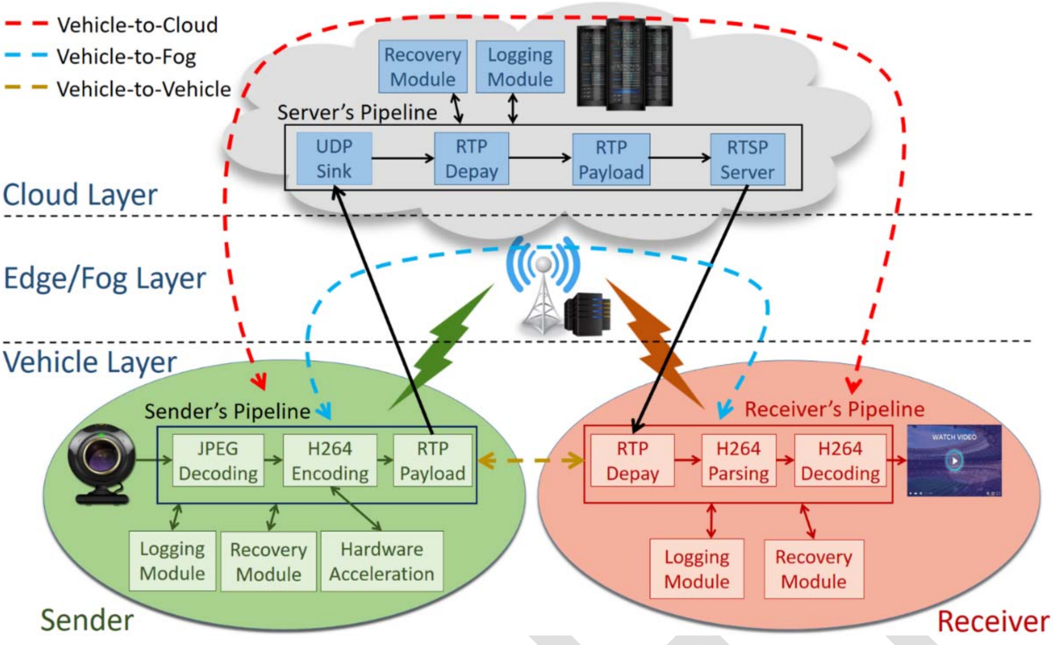


Figure 95: LEVIS streaming architecture[4]

10.2.3. Future Work

Infrastructure Support for Early Trials and Upgrades for Full Trials (Phase 4 and Phase 5)

The activities in the FI-TS are summarized as follows for each user story:

- *FI-TS extended sensors user story*: the development of the MEC discovery was completed and testing was conducted in the lab with local connectivity. The next tests were performed in the field, whereby, the MEC service discovery and migration was tested between MECs associated with two different 5G PLMNs. This final MEC implementation was also targeted to be customised, re-deployed and tested in the ES-PT CBC. Further enhancements the MEC service discovery were considered to take advantage of the features enabled by AALTO 5G network migration from NSA to SA mode.

- *FI-TS remote driving user story*: the development of the LEVIS streaming client and server is complete and current thereafter followed the integration of the LEVIS application to the S4 automated driving vehicle system and ROC. Initial testing of this integration work was conducted in lab environments and the integration work was completed in January 2021. At the same time the remote driving application was being developed to include LiDAR sensor data in the transmission from the vehicle to the ROC (completion by February 2021). Thereafter, the testing and validation of the complete remote driving user story was conducted on the road, including the testing of the

[4] O. El Marai and T. Taleb, "Smooth and Low Latency Video Streaming for Autonomous Cars during Handover," IEEE Network Magazine, p. 8, 2020

continuity of the remote driving session (including video streaming), when vehicle 5G connection is handed over between two unique PLMN. To that end, all the early trials implemented the inter-PLMN handover using multi-SIM OBUs.

At the current time of reporting, all the application development work for the two aforementioned user stories had been completed, with only pre-trial testing being done in time for the final FI-TS trial and demos tentatively scheduled for April 2022.

10.3. Trial Site FR Roll-out

10.3.1. Introduction

The FR TS is implementing one user story called "infrastructure-assisted advanced driving", where the main intelligence for vehicles' advanced maneuvers will be located in the MEC/cloud side. It deals with safe lane change maneuver assisted by Cloud/MEC: Automated vehicles need to change the lane for various reasons, where in some cases the local perception of the perception of the CAV built by the vehicle embedded sensors are not enough to conduct a safe lane change maneuver. In the second phase of the FR TS, cloud system is used to inform the AV about a non-allowed autonomous driving zone, and then the vehicle needs either to change its lane or change its driving mode.

10.3.2. FR Infrastructure Deployment and Enhancement Activities

RSU / Edge Deployment

FR TS is deploying MEC at the edge of network for data processing and maneuver intelligence required for autonomous driving maneuver coordination. For that purpose, AKKA, partner of FR TS, is deploying an ETSI based MEC platform to host VEDECOM applications.

Numerous open-source solutions can provide the necessary environment to run MEC latency sensitive applications. However, only a few support mobility and are ETSI compliant. In previous studies, AKKA has explored the most recognized solutions by the community (LLMEC, CORD, StarlingX, Akraino Edge Stack) while emphasizing how strongly they comply with the ETSI reference architecture and how well they fit into the CCAM context. This was summed up in a paper[5] released by AKKA.

OpenShift, one of the studied open-source solutions, has satisfied multiple requirements of the FR TS such as, user friendliness, abundant documentation. Additionally, it is closely compliant to the ETSI standard. Consequently, this solution was chosen as the main component of the AKKA's MEC platform.

ETSI MEC architecture includes different components that have explicit functionalities. Each used technology or tool should be capable of reproducing these functionalities, or at least partially.

The MEC platform prototype architecture that AKKA has designed is based on the ETSI reference architecture. In order to highlight the latter, ETSI blocks are represented in blue rectangular-shaped boxes whereas the green boxes represent the tools we rely on technically (see Figure 96).

[5] On the implementation of an ETSI MEC using open-source solutions – Virtual ITS European congress 2020

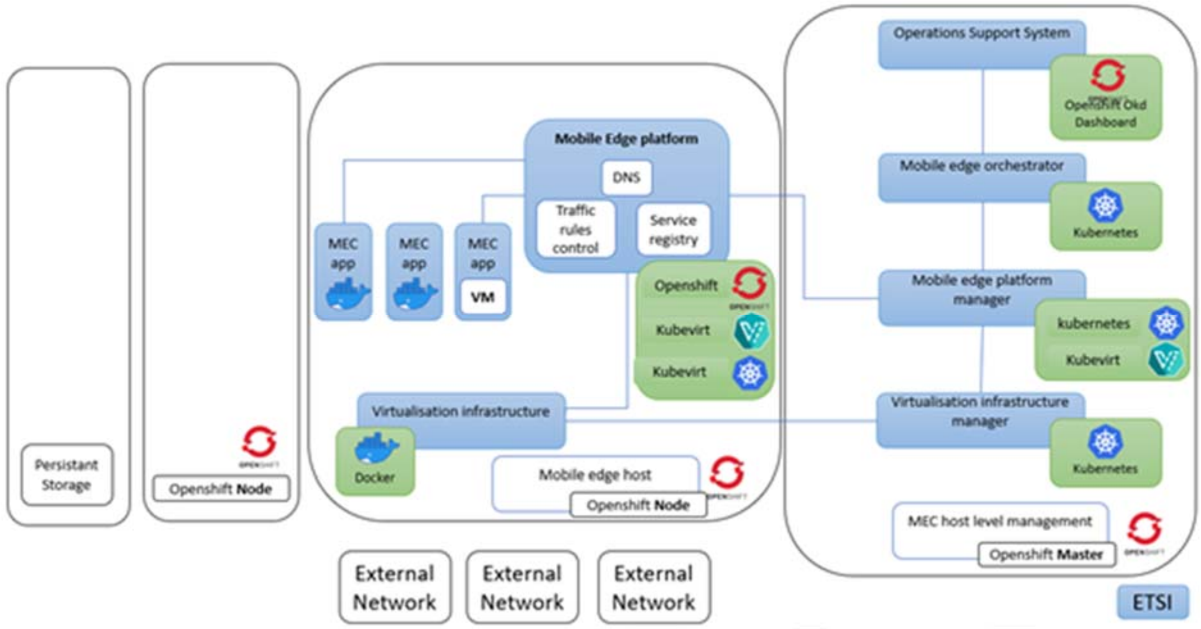


Figure 96: MEC prototype architecture

OpenShift origin, lately renamed as OKD, is the main technical open-source solution chosen by AKKA and provides valuable tools within its infrastructure. OpenShift leverages Kubernetes underneath as the primary infrastructure component. On Kubernetes we can deploy applications in the form of containers such as Docker. In addition to the local developed images, OpenShift provides a wide container registry that comes built in with OKD. OpenShift also comes with a built-in integration to source code management systems through which users can import their application code. Finally, OpenShift uses an ETCD key-value store that hosts information about various components. Generally, OpenShift is configured with multiple nodes where the actual Docker images are hosted. These nodes are managed by one or more master nodes.

To manage the cluster that will host FR TS's MEC applications, OKD comes with a built-in web console that developers can access to browse and manage their applications in addition to a command line interface that allows you to manipulate all these tools (Kubernetes, Docker) as well as other services that you may add in the cluster if needed.

AKKA's team unravelled OKD and studied its components to build an ETSI-compliant architecture. Considering OpenShift configuration, the Master node will be dedicated for management at the MEC host level and orchestration at the MEC system level. The MEC hosts will be represented by OpenShift regular nodes.

Considering that VEDECOM will be needing two types of application isolation, namely, containers and Virtual machines, AKKA has chosen KubeVirt, a tool that can be added to the OpenShift infrastructure allowing VMs to be deployed and managed by OpenShift in the same way containers are. Hence, OpenShift can handle an application deployed on VMs just like any other containerized application within Docker. It allows the French trial site to use applications that cannot be containerized but must be isolated within a VM instead.

Sensor Development

To carry out the infrastructure assisted advanced driving, several sensors will be deployed at both Satory (see Figure 97) and TEQMO tracks.

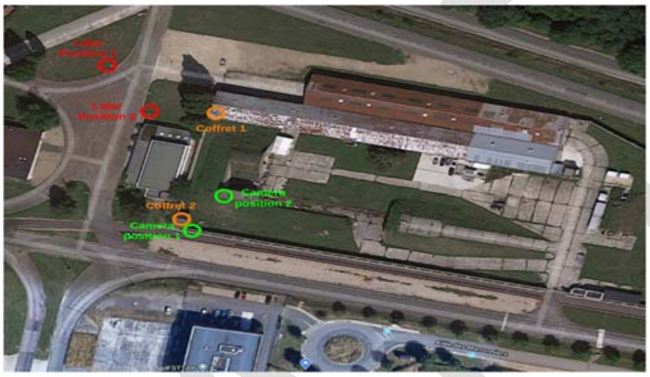


Figure 97: Sensors deployment plan at FR TS Satory Site

Hence, the following equipment will be installed:

- 1 LiDAR with Qortex UT
- 1 PTZ camera Axis M5525E
- 1 Fisheye camera Axis
- 2 Roadside coffrets

Roadside sensors data are connected through optical fiber to the Infrastructure server where their data are made available to other servers and for Supervision Operator. Video flows are available directly from the Axis device itself using IP address of the device, the RTSP service and the API from the constructor.

The supervision operator is the only one that can use PTZ capabilities (zoom, move, …).

Data from the LiDAR are the direct output of the Qortex (Treatment Unit from Quanergy). A bridge will be developed for the data to be made available both for the MEC and for the PPU to perform calibration and upload corrected data for augmented perception feature at the Supervision Operator.

FR TS Cloud Deployment

FR TS user story deals with safe lane change maneuver assisted by Cloud/MEC. For that purpose, CATAPULT, FR TS partner, is developing a cloud system that will host VEDECOM applications based on a Google cloud. This latter is used to host non delay sensitive applications.

10.3.3. FR Platform and Apps Evolution

In the "infrastructure-assisted advanced driving" US from the FR TS, the main intelligence for vehicles advanced maneuvers is located in the MEC/cloud side. Hence, the functionalities like data fusion, risk analysis, trajectory guidance modules are residing in the MEC, as opposed to other approaches where the intelligence is located in the vehicle.

To do so, VEDECOM has designed a modular approach to be developed in the MEC where the modules are categorized in 3 different classes: (1) C-ITS modules (2) Perception modules and (3) Maneuver coordination modules, as depicted in Figure 98. In the following, the different developed functionalities will be detailed:

- **C-ITS modules**

This module includes the functionalities related to the ITS stack and the sending/reception of ITS messages to vehicles/ from vehicles. The CAM client module is responsible for receiving all the CAM messages generated by connected and autonomous vehicles. This data is then stored in a local database at the MEC.

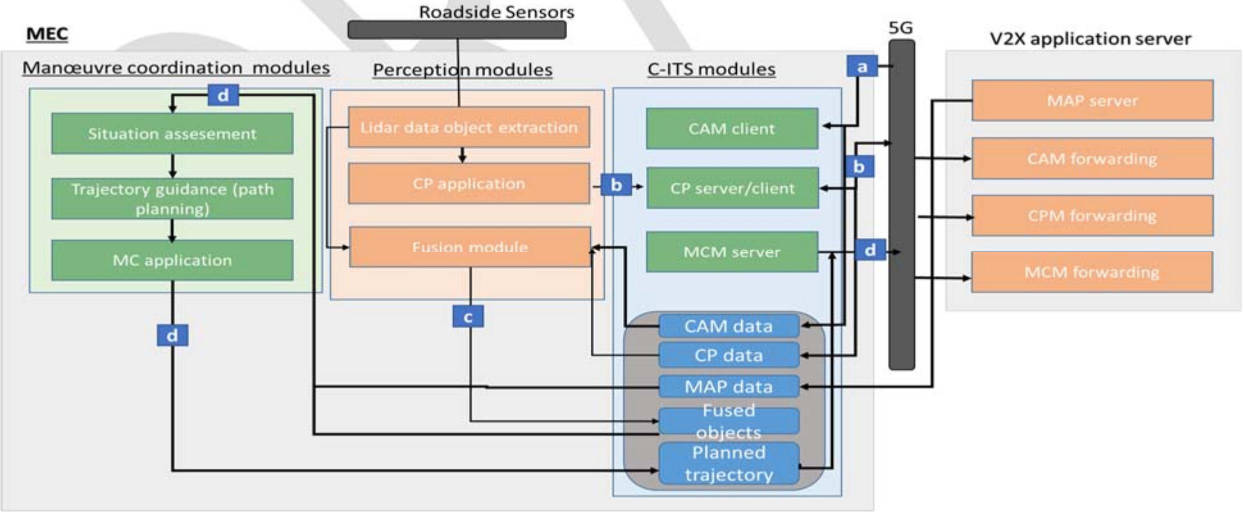


Figure 98: FR TS MEC developed functionalities

The cooperative perception server module is allowing the sending of CPM messages, generated from roadside sensors data, to autonomous and connected vehicles located in the testing zone to enhance their perception of the environment. On the other hand, the CP client is collecting the CPM messages from vehicles and storing them in the local database at the MEC entity. Furthermore, the MCM server module sends the MCM messages, constructed at the MEC level, to the concerned autonomous vehicles in the testing track to trigger the lane change maneuver.

- **Perception modules**

The main goal of this framework is to construct the CPM message from the collected roadside sensors data. To do so, several steps are defined spanning from LiDAR data object extraction to CP application where the actual message is built. The data fusion module is used for the augmented perception where the perceived objects obtained from both the perception and the communication are fused. The output of this module is used as input for the situation assessment module to evaluate the collision risk.

- **Maneuver coordination module**

Three main sub-modules are present in this module. The situation assessment entity is evaluating the risk of collision based on input from the perception and communication data. Then, the output of this module is directed to the trajectory guidance module where the new path that has to be followed by the vehicle is carried out. Based on this, the Maneuver Coordination (MC) application will construct the MC message and fill it in with the required information.

- **Data fusion application**

Through the data fusion application, the MEC database is maintained up to date and has all the vehicles detected by the roadside sensors and/or CA (CAM) and CPMs received from the other vehicles within reach.

The MCS is typically used for guiding autonomous vehicles with the objective of avoiding collision with basic vehicles. This service is provided by MEC which has full perception and I2V communication capabilities regarding the vehicles being considered. For this purpose, the MEC will be using the MCM which is under standardization at ETSI TC ITS WG1 level.

An MQTT broker is installed in the MEC. A second broker may be installed with security features if needed. The CA/CP/MC/MAP services will publish their messages on specific topics. Vehicles and other services will subscribe to the required topics as illustrated by the next figure.

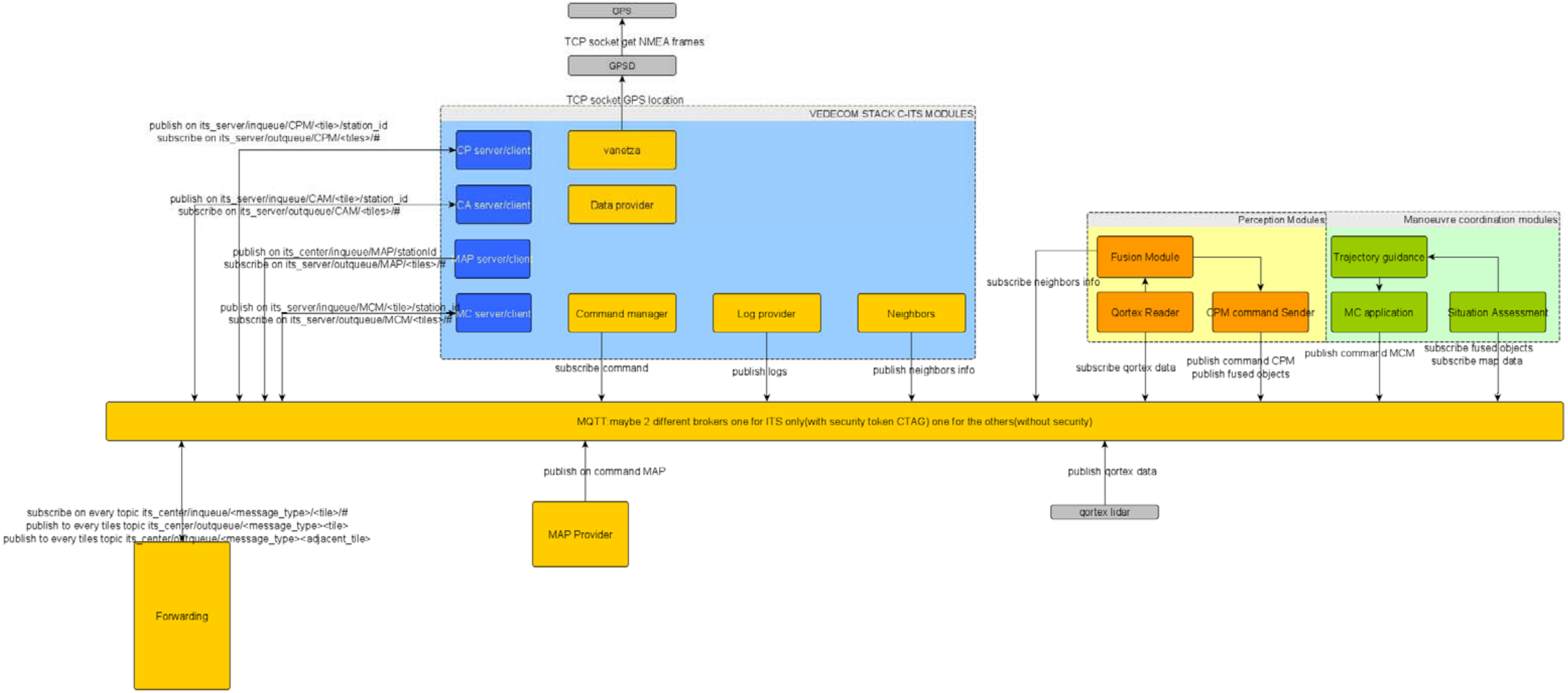
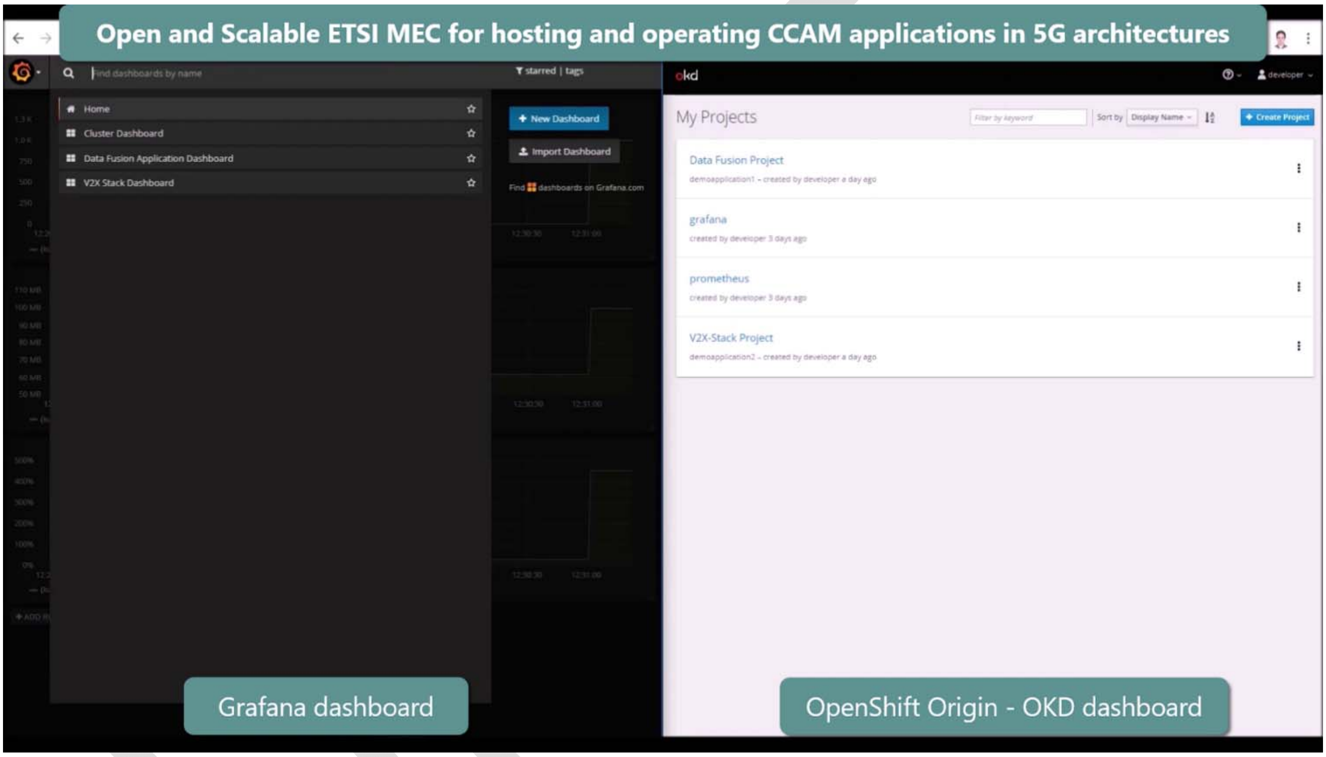

GPS
TCP socket get NMEA frames
GPSD
TCP socket GPS location
VEDECOM STACK C-ITS MODULES
publish on its_server/inqueue/CPM/<tile>/station_id
subscribe on its_server/outqueue/CPM/<tiles>/#
CP server/client
vanetza
publish on its_server/inqueue/CAM/<tile>/station_id
subscribe on its_server/outqueue/CAM/<tiles>/#
CA server/client
Data provider
publish on its_center/inqueue/MAP/stationId
subscribe on its_server/outqueue/MAP/<tiles>/#
MAP server/client
publish on its_server/inqueue/MCM/<tile>/station_id
subscribe on its_server/outqueue/MCM/<tiles>/#
MC server/client
Command manager
Log provider
Neighbors
subscribe command
publish logs
publish neighbors info
Perception Modules
Fusion Module
subscribe neighbors info
Qortex Reader
CPM command Sender
subscribe qortex data
publish command CPM
publish fused objects
Manoeuvre coordination modules
Trajectory guidance
MC application
Situation Assessment
publish command MCM
subscribe fused objects
subscribe map data
MQTT:maybe 2 different brokers one for ITS only(with security token CTAG) one for the others(without security)
subscribe on every topic its_center/inqueue/<message_type>/<tile>/#
publish to every tiles topic its_center/outqueue/<message_type><tile>
publish to every tiles topic its_center/outqueue/<message_type><adjacent_tile>
Forwarding
publish on command MAP
MAP Provider
publish qortex data
qortex lidar

Figure 99: MQTT implementation at the FR TS MEC

10.3.4. FR Infrastructure Integration Activities

FR MEC Integration

FR TS had carried out a first integration session during September 2020 to integrate the AKKA MEC platform in the VEDECOM server (see Figure 100). The OpenShift components have been installed on the MEC hardware. The necessary tests have been conducted to ensure that the cluster is operating correctly. Then a first release of the CCAM application has been successfully on-boarded.



Open and Scalable ETSI MEC for hosting and operating CCAM applications in 5G architectures

Data Fusion Project
demoapplication1 – created by developer a day ago

grafana
created by developer 3 days ago

prometheus
created by developer 3 days ago

V2X-Stack Project
demoapplication2 – created by developer a day ago

Grafana dashboard

OpenShift Origin - OKD dashboard

Figure 100: AKKA MEC Integration in VEDECOM server

To carry out this deployment, VEDECOM is using a DELL tower Dell Precision 5820. As it is needed to virtualize full running virtual machines and containers, the server has 32 GB of RAM DDR4. It has a 512Gb of disk space SSD to run the operating system which is Ubuntu 18.06 and 2 HDDS each of TB. The first HDD is used for the platform to be installed. All the VMs and containers and the orchestrator are installed in the first HDD of 2TB. The second HDD is used as a backup space where the orchestrator will save snapshots of VMs and containers.

The server uses a Xeon Intel processor clocked at 3.6Ghz and each VM uses 4 cores in parallel. While the containers are less demanding, the CPU speed is still needed for the heavy computational power of the VMs.

As the server will be connected to many networks it has 4 network cards.

FR Cloud Integration

The cloud server hosts the Vedecom CCAM application and requires a MQTT broker that communicates with both the OBU and the application on a Kubernetes cluster engine configured on a R440 DELL server. To enable communication with the Kubernetes API and provide enhanced security and efficiency, a containerized version of the CCAM application and MQTT broker has been installed. Both the MQTT broker and the Vedecom CCAM application are TLS encrypted.

The CCAM application and the MQTT broker that communicates with both the OBU and the application, are configured in the form of Docker containers (Kubernetes pods), to enable communication with the Kubernetes API. In addition, intra-cluster and external communication are enabled by exposing the cluster-IP and NodePort Kubernetes services manifests, respectively. It is worth mentioning that the WAN IP addresses shown in Figure 101 are exemplary; however, it is crucial that a static WAN IP address will be configured on the terminating device to create the VPN tunnel.

The end-to-end hybrid 5G-satellite architecture to be deployed, in the FR-TS is illustrated in Figure 101.

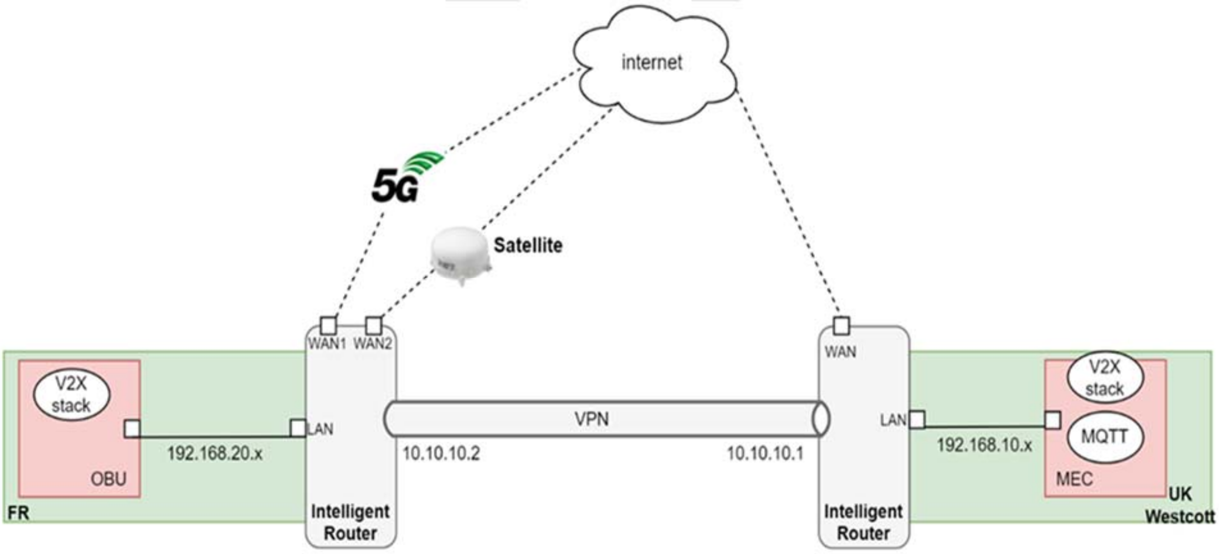


Figure 101: FR TS Cloud server integration

10.3.5. FR Infrastructure Testing Activities

FR TS has finished the implementation of MEC applications. The status of each module is given below:

- V2X applications server: ready
- Data fusion: ready
- Cooperative perception: ready

- Trajectory guidance: ready
- Predictive QoS: ready
- Risk assessment: ready

From the vehicle side, the AV and the OBU are also ready.

A testing phase had started to test the functioning of these modules. Two validation phases including laboratory validation and closed-circuit validation have been performed.

Laboratory validation

The objective of this step is to approve the output of the system before testing it on a closed circuit or on an open road. It consists in carrying out, in a first phase, unit tests to validate the various components of the system. Then, in a second phase, to make a complete chain test. In Table 10, the different tests made to validate the C-ITS modules in the MEC are presented.

Table 10: Validation tests for C-ITS modules in the MEC

Environment: VEDECOM Laboratory		
Component : MEC C-ITS modules		
TEST ID	**TEST PURPOSE**	**RESULT**
MEC -CAM01	Test that the **MEC CA Client** is able to receive CAM from connected vehicle.	**Validated**
MEC-MCM01	Test that the **MEC MC Server** is able to broadcast MCM message without trajectory guidance.	**Validated**
MEC-MCM02	Test that the **MEC MC Server** is able to broadcast MCM message, with trajectory guidance container.	**Validated**
MEC-CPM01	Test that the **MEC CP Server** is able to broadcast CPM message.	**Validated**
MEC-MAP01	Test that the **MEC MAP Server** is able to broadcast MAP message.	**Validated**

Also, tests related to the perception modules located in the MEC, as indicated in Table 11, have been conducted:

Table 11: Perception modules tests in the MEC

Environment: VEDECOM Laboratory		
Components: MEC perception modules		
TEST ID	TEST PURPOSE	RESULT
MEC-PER-L01	Test that the **MEC Local perception module** is able to detect a vehicle driving in a targeted lane.	**Validated**
MEC-PER-L02	Test that the **MEC Local perception module** is able to detect two vehicles driving in adjacent lanes.	**Validated**
MEC-PER-L03	Test that the **MEC Local perception module** is able to detect two vehicle driving in the same direction with an inter-distance of 5m.	**Validated**
MEC-PER-A01	Test that the **MEC Fusion module** is able to provide a list of objects when no V2X communication exists.	**Validated**
MEC-PER-A02	Test that the **MEC Fusion module** is able to provide a list of objects when no local perception exists.	**Validated**
MEC-PER-A03	Test that the **MEC Fusion module** is able to add an object detected by the local perception system to its list of objects when no V2X communication data exists.	**Validated**
MEC-PER-A04	Test that the **MEC Fusion module** is able to add an object to its list of objects based on CAM data when no local perception information is available.	**Validated**
MEC-PER-A05	Test that the **MEC Fusion module** is able to associate a CAM Data with an existing object that has been detected by the local perception.	**Validated**
MEC-PER-A07	Test that the **MEC Fusion module** is able to associate an object detected by the local perception with an existing object extracted from V2X communication data	**Validated**

In addition to that, tests related to MEC MQTT broker have been made. The results are shown in Table 12.

Table 12: MEC MQTT Broker tests

Environment: VEDECOM Laboratory		
Components: MEC MQTT broker		
TEST ID	TEST PURPOSE	RESULT
MEC-BR-MQTT02	Test that the **MEC MQTT broker** is able to forward CAMs messages from MEC to vehicles and vice-versa	**Validated**
MEC-BR-MQTT03	Test that the **MEC MQTT broker** is able to forward MCM messages from MEC to vehicles and vice-versa	**Validated**

MEC-BR-MQTT04	Test that the **MEC MQTT broker** is able to forward CPM messages from MEC to vehicles and vice-versa	**Validated**
MEC-BR-MQTT05	Test that the **MEC MQTT broker** is able to forward MAP messages from MEC to vehicles and vice-versa	**Validated**
MEC-BR-MQTT06	Test that the **MEC MQTT broker** is able to forward video from MEC to vehicles and vice -versa	**Validated**

Test-field validation

In order to validate the different MEC modules, we have carried out test in Satory site (closed one) as illustrated in Figure 102. The baseline scenario tests the insertion of a legacy vehicle in the highway circuit where an autonomous vehicle is heading. Infrastructure sensors detect the coming legacy vehicle and send their data to the MEC. This latter carries out a data fusion operation and assesses the collision risk between the two vehicles. If the risk is present, the MEC sends an MCM message to the AV to change its lane.

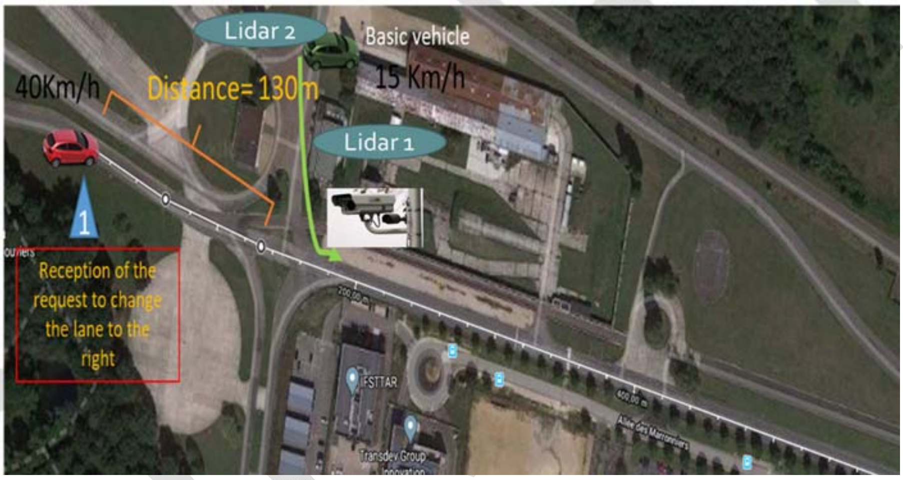
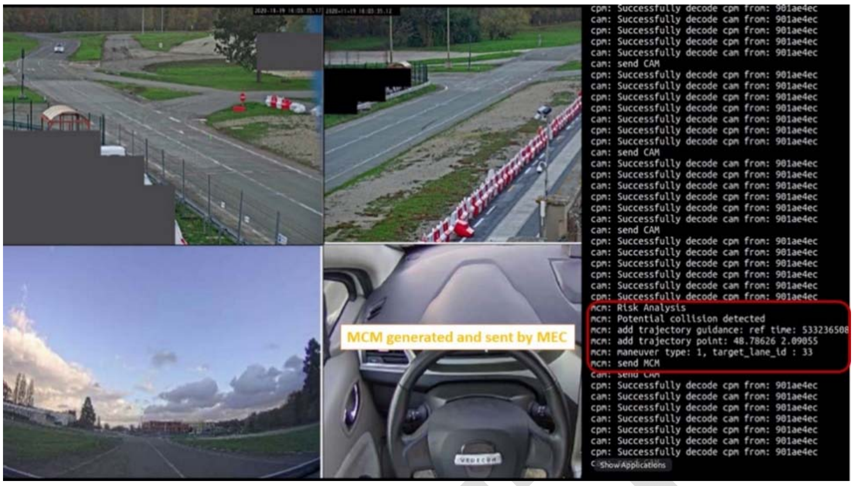


Figure 102: FR TS testing baseline scenario

In Figure 103, a capture of the MEC sending MCM activity is depicted when a collision risk is detected.



```
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cam: send CAM
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: send CAM
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cam: send CAM
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
mcm: Risk Analysis
mcm: Potential collision detected
mcm: add trajectory guidance: ref time: 533236508
mcm: add trajectory point: 48.78626 2.09055
mcm: maneuver type: 1, target_lane_id : 33
mcm: send MCM
cam: send CAM
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
cam: Successfully decode cam from: 901ae4ec
cpm: Successfully decode cpm from: 901ae4ec
```

Figure 103: MCM Activity in the MEC

10.3.6. Future Work

Infrastructure Support for Full Trials (Phase 5)

At the current time of reporting in March 2022, several activities are carried out in order to facilitate full trials. In this phase, infrastructure support and maintenance activities will be carried out in order to keep the FR TS operational for the remaining duration of the project.

10.4. Trial Site NL Roll-out

10.4.1. Introduction

The NL test site manages three different user stories: Cooperative collision avoidance, Extended sensors with CPM messages, and Remote driving using 5G Positioning, defined in Sections D2.1 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019). This section describes the development stage and roll-out details of the NL CCAM infrastructure depicted in Section 3 of D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), in order to carry out the defined user stories.

10.4.2. Overview of Infrastructure Components

An overview of the deployment in the NL test site is depicted in Figure 104. The colours refer to the partners responsible for the deployment for the specific components.

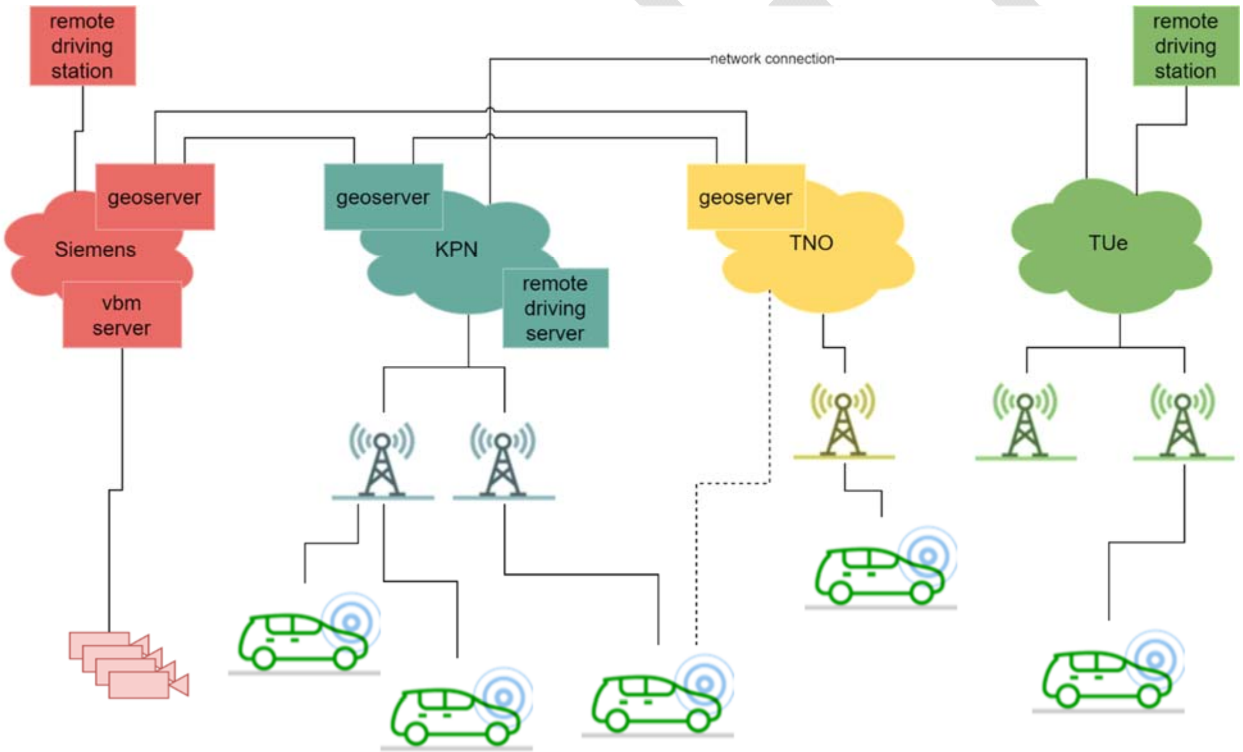


Figure 104: Overview of the roadside deployment in the NL trial site

On the trial site, three different 5G networks are deployed (KPN, TNO, Tue), such that a virtual cross border situation can be created. The test vehicles for the different use cases can connect to the different networks. At three locations, a geoserver is deployed that handles the delivery of ITS messages to vehicles and roadside systems. The geoservers are deployed in the MEC of the networks (KPN, TNO), but can also be deployed in the cloud (Siemens). A geoserver in the MEC is not required for Tue, as it is possible for vehicles to directly connect to a geoserver deployed elsewhere, if network connectivity is realized.

For the remote driving user story, two remote driving stations are deployed, at Siemens and at Tue. These connect to the vehicles connected to the different networks. A remote driving server is deployed in the KPN MEC, to facilitate the realization of the connection between remote driving stations and remotely driven vehicles.

For the extended sensors use case with CPM messages, the Video-Based Monitoring system (VBM) of Siemens is used. The cameras are connected via a fiber network to the back office of Siemens, where all image processing takes place.

For the CoCA application, the vehicles will be connected to different networks, one vehicle will be connected to the KPN network, the other vehicle to the TNO network

10.4.3. Infrastructure Deployment and Enhancement Activities

Sensor Development and Deployment

The video-based monitoring (VBM) system is used in the extended sensor user story. Cameras are deployed along the A270/N270 highway in The Netherlands, as depicted in Figure 105. Cameras are spaced every 100m, covering in total about 5 km of highway and interurban roads.

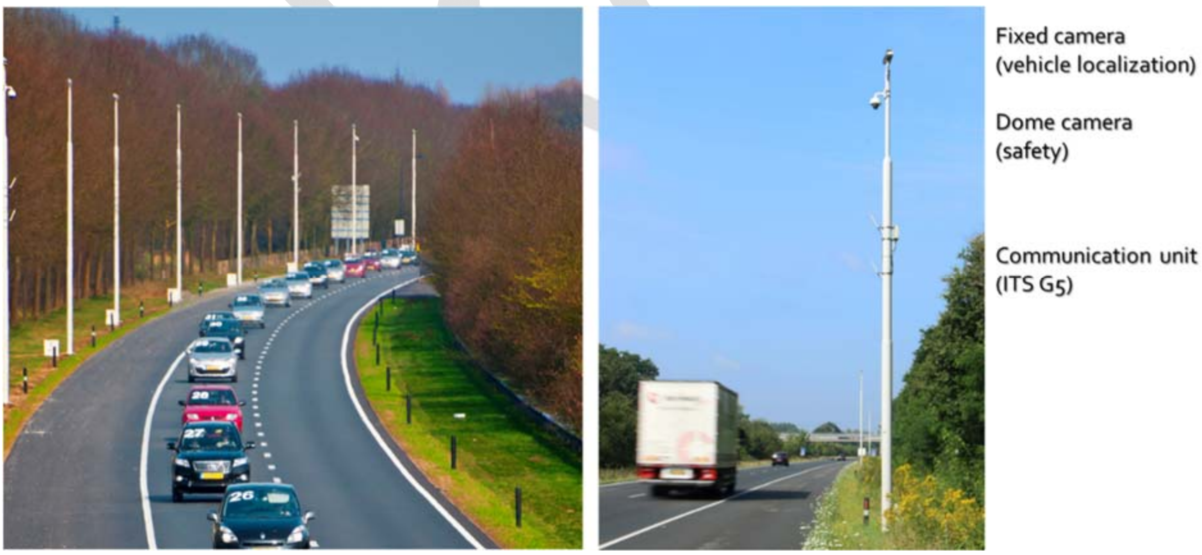


Figure 105: Overview of the deployed VBM cameras

The VBM system consists of 50 detection cameras, 20 detector servers, and 1 VBM tracker. The cameras are connected via an optical fiber network to the Siemens back-office. In the back-office, the detector servers and tracker computer are installed. The detection system is described in more detail in (VAN HUIS, BAAN, & VAN LOON, 2009). An overview of this deployment is shown in Figure 106. The cameras are the online outdoor equipment, all other equipment is deployed in the Siemens back-office.

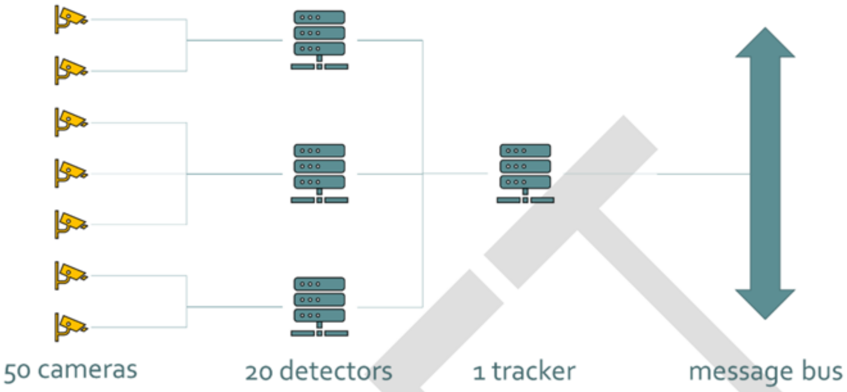


Figure 106: Schematic overview of the architecture of the VBM system

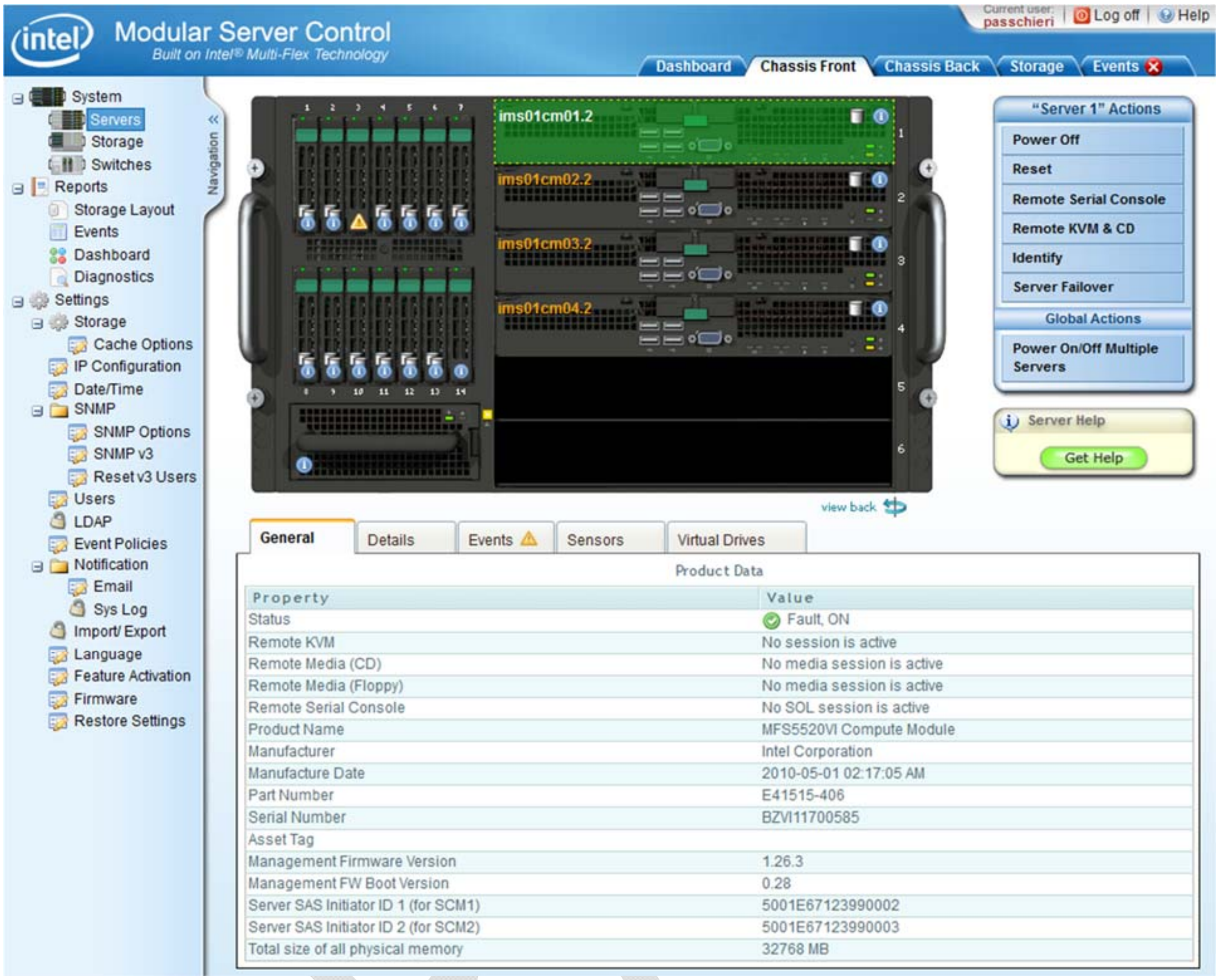
The message bus is implemented in a RabbitMQ broker version 3.6.9[6] on a virtual machine deployed in a private cloud operated by Siemens. The private cloud is deployed in the same back-office environment where also the detector servers and the tracker PC are installed. This guarantees minimum latency between vehicle detections and applications utilizing those detectors. The CPM message creator application is connected to the same message bus, as detailed next.

RSU / Edge Development and Deployment

No dedicated RSU are deployed in the NL trial site. Currently, 2 MEC server environments have been deployed, for the networks of KPN and TNO, respectively. A third MEC deployment is foreseen in a later stage in the Tue network. Note, that for MEC hand-over related activities, 2 MECs are sufficient.

The MEC in the KPN network is an integral part of the mobile network deployment itself and is described in D3.3 (5G-MOBIX, Deliverable D3.3: Report on the 5G technologies integration and roll-out , 2020). The eNB in the TNO 5G network are connected to the TNO core via the physical network infrastructure of Siemens, utilizing the same optical fiber network that is used for the VBM camera system. From the Siemens network infrastructure, a connection is realized to the TNO core network. The MEC that is part of the TNO 5G network has also been deployed in the Siemens network, enabling local breakout between the eNB and the core network. The MEC is deployed in a set of virtual machines in an Intel Modular Server, version 10.4.100. This system consists of 32 CPU cores, 256 GB RAM, and 16 TB storage. A Citrix XenServer 6.2.0 is running as virtualization layer on top of the Intel Modular Server (see Figure 107).

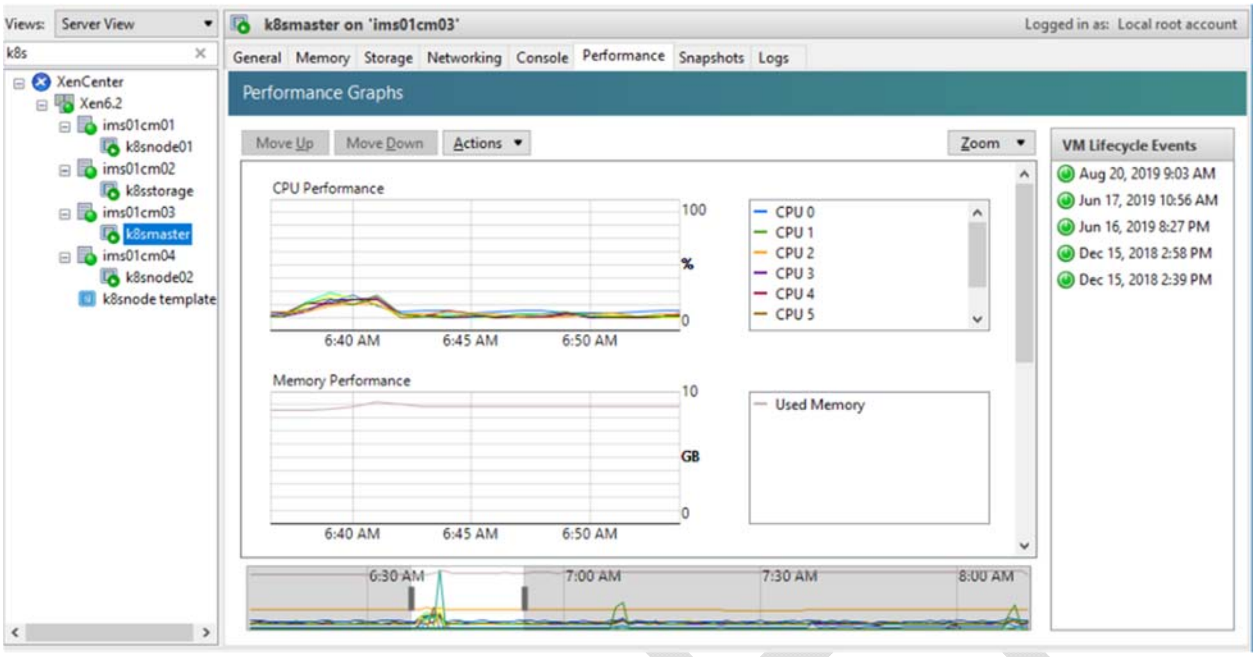
[6] http://rabbitmq.com



Property	Value
Status	Fault, ON
Remote KVM	No session is active
Remote Media (CD)	No media session is active
Remote Media (Floppy)	No media session is active
Remote Serial Console	No SOL session is active
Product Name	MFS5520VI Compute Module
Manufacturer	Intel Corporation
Manufacture Date	2010-05-01 02:17:05 AM
Part Number	E41515-406
Serial Number	BZVI11700585
Asset Tag	
Management Firmware Version	1.26.3
Management FW Boot Version	0.28
Server SAS Initiator ID 1 (for SCM1)	5001E67123990002
Server SAS Initiator ID 2 (for SCM2)	5001E67123990003
Total size of all physical memory	32768 MB

Figure 107: Overview of the management interface of the Intel Modular Server

The MEC computational environment is realized as a Kubernetes cluster, version 1.13.5, deployed in a set of virtual machines in the XenServer (see Figure 108). A minimum of 3 nodes are deployed, 1 master node and 2 worker nodes. Additional worker nodes are created on demand in the XenServer virtualization layer.



Aug 20, 2019 9:03 AM
Jun 17, 2019 10:56 AM
Jun 16, 2019 8:27 PM
Dec 15, 2018 2:58 PM
Dec 15, 2018 2:39 PM

Figure 108: Overview of the Kubernetes cluster nodes, deployed in the XenServer environment

The Kubernetes cluster uses docker version 19.03.6 as runtime environment and is managed with Rancher version 2.3.6 (see Figure 109).

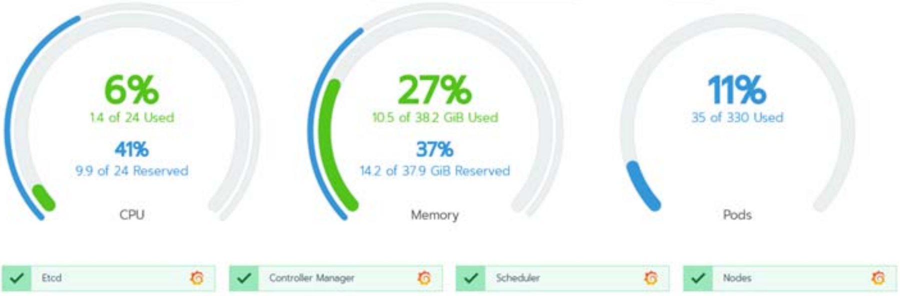


Figure 109: Management console overview of the TNO MEC

Cloud Deployment

All cloud functions specific for 5G-MOBIX NL Trial site are deployed in the same XenServer environment managed by Siemens where the TNO MEC is also deployed. The details of this local cloud solution are described in the previous section.

10.4.4. Platform and Apps Evolution

The three user stories tested at the NL Trial site require dedicated functions in the MEC or cloud. These functions are described in more detail in the sections below.

Remote driving

A central FleetManager function is required for the remote driving use story. The task of the FleetManager is to enable remote driving stations and remote vehicles to be connected, i.e., it manages the link between specific remote driving control stations and specific vehicles. For single operator deployment scenarios, the FleetManager is deployed in the KPN MEC. To support multi-operator solutions as well, a second instance is of the FleetManager is deployed in the Siemens local cloud environment.

The FleetManager is only involved in setting up the connections between vehicles and driving stations. The actual remote driving is based on direct communication between the vehicle and the driving station, not passing through the FleetManager. Every vehicle and driving station is always associated with a single FleetManager, so before testing activities, they have to be associated with either one of the 2 deployed FleetManagers. This is a simple configuration activity that needs to be performed but ensures that a single instance can manage the connections, and thus guarantee safe operation.

This application will also be used to validate and demonstrate the flexibility in deployment models that has been realized by selecting identical runtime environments (Kubernetes) for both cloud and MEC deployments.

Extended Sensors with CPM messages

Two network functions are required for this user story. First, an application AppVbmCpmSender is required to use the vehicle detections from VBM and convert them in CPM messages. Secondly, a GeoServer application that is responsible to distribute geographically scoped messages to and from vehicles, roadside systems, MEC systems, and cloud systems need to be developed (Figure 110).

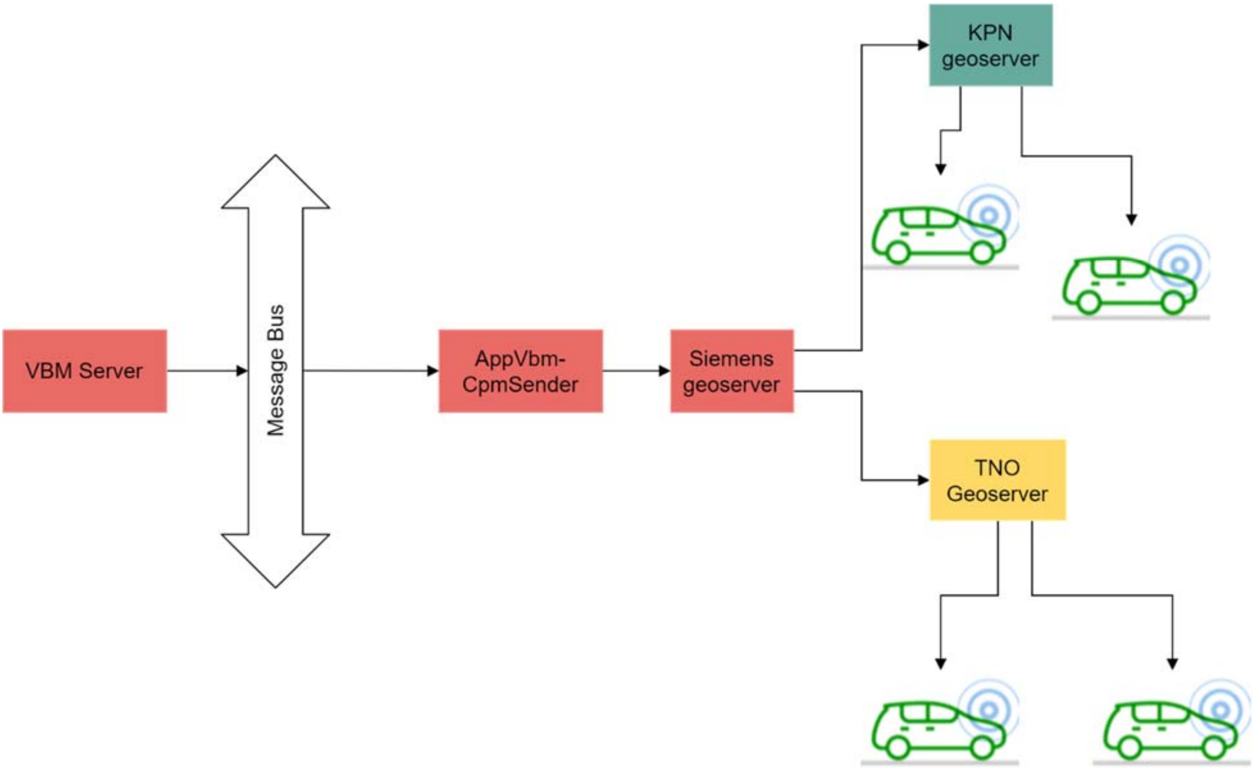


Figure 110: Deployment architecture of the functions required for the user story extend sensors with CPM messages

AppVbmCpmSender

The AppVbmCpmSenser application consists of two main parts. The first part is a generic ASN.1 library that is capable of generating standard ITS messages based on formal ASN.1 definitions. These definitions are standardized by, e.g., ETSI ITS and/or SAE. All published standards are supported by this library. This generic library is used also for other projects and activities. All ITS messages generated or received on the NL Trial site by Siemens are processed by this library, ensuring conformance to the standards. A similar approach is by TNO, which ensures consistent interoperability at the message definition level.

The CPM message is being standardized by ETSI in (ETSI TR 103 562, 2019), but this standard is not yet finished. Therefore, only draft versions of the standard can be used. The library support drafts version 0.19 and 0.15. At this moment, the vehicles used for this user story have implemented version 0.15. Therefore, also the AppVbmCpmSender uses the version 0.15 of the standard.

The second part of the application is responsible for interpreting the VBM data and filling the data fields in the CPM messages. A CPM message consists of two main parts: a description of the sensors that have generated the data in the message (position, orientation, coverage, etc.) and the actual object detections (position, velocity, type of object, accuracies, etc.). A choice needs to be made on how to combine the detections from VBM in one or more CPM messages. One extreme case would be to combine all detections from all cameras in a single message. This would minimize the number of messages per second and would also minimize the message overhead. However, the messages would become very large, possibly causing fragmentation in the message transmission, resulting in a reduced performance.

The second extreme would be to use a single message per camera, leading to small messages, relatively large overhead, and large amounts of messages: if all 50 cameras would be used, that would lead to 500 CPM messages/second.

The CPM message uses an internal coordinate system relative to the sensor location, and not a global coordinate system (e.g., GNSS latitude and longitude positions). The output of VBM is already in global coordinates, so it first needs to be transferred back to a local coordinate system[7]. There is actually no need to use the real local coordinate system of the sensors, any local coordinate system can be chosen for this. In the vehicle, the coordinates will be transferred back to a global coordinate system (before transferring to a local vehicle coordinate system) so the easier the transformation is made on the roadside, the easier it will be for the vehicle as well. Therefore, it was decided to only use a local reference system per message that is based on the first camera (in the driving direction) that is included in a message. Further, to find a balance between message size and number of messages, up to 10 cameras are included in a single message, i.e., a stretch or road of about 1 km per message. Apart from finding a balance between message size and the number of messages, there is also a technical limitation in the standard that limits the extension that can be included in a single message: any detection include in a message needs to be within 1.5 km from the origin of the chosen coordinate system, which puts a theoretical upper limit of 3 km coverage per CPM message. As it was decided to use the first camera per message as the reference position, which makes it much easier for a human interpretation of the actual coordinates, and thus allow for faster and more reliable development, there will always be positive coordinates in the longitudinal direction, and thus are limited to 1.5 km at most.

In addition to these functional components in the AppVbmCpmSender, also the interfaces to the message bus and GeoServer are implemented, to allow to connect the application to its inputs and output (Figure 111).

[7] It will be proposed to the ETSI task force working on this standard to (also) allow for global coordinates to be used, to improve performance in similar use cases as developed here.

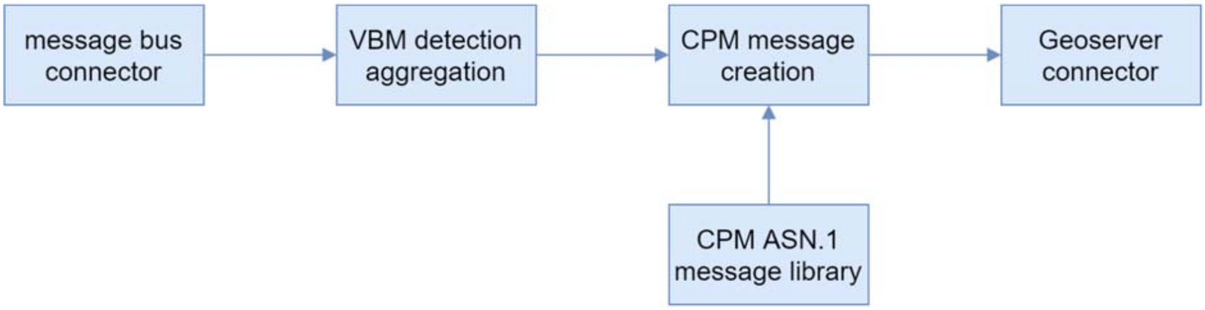


Figure 111: Diagram describing the functions of the AppVbmCpmSender application

Geoserver

The geoserver is responsible for the location-based distribution of ITS messages to vehicles and other ITS systems. Typically, a single geoserver is deployed in the MEC of the 5G networks, to facilitate low latency communication to and between vehicles. The geoserver is implemented based on a client-broker architecture, where the geoserver is the broker. The communication between client and geoserver is based on the MQTT protocol. In MQTT, every message is tagged with a topic when it is sent. For this purpose, a topic definition that encodes the message type and version, relevant geographical location, and sender station, was adopted. The geoserver is implemented as a docker container and deployed in the MEC of the TNO and KPN networks, and in the Siemens cloud. A single container definition is used for all locations, ensuring consistent interoperability, where only the authentication information is configured differently per deployment (to reflect the different responsibilities for the three deployments). The architecture of the geoserver is agnostic of geographical location or scale, which allows the same solution to be deployed in the cloud or MEC. This also allows for both horizontal scaling, by reducing the geographical area covered by a single instance, or vertical scaling, by realizing an aggregation tree based on geographical areas. It is similarly possible to scale based on message type, as both message type and geographical area are encoded in the topic. Generic load balancers exist for the MQTT protocol that can be added to the Kubernetes clusters to inspect the Topics and distribute the messages to the appropriate broker. This extension has not yet been deployed, as the scale of the 5G-MOBIX trials do not require this function.

This solution has been initially developed in the European Concorda project, and further developed in 5G-MOBIX. The most important extension is the definition of an interface to interconnect multiple instances of geoservers that need to serve the same or adjacent geographical areas. This function is required when multiple network operators deploy geoservers in their own MEC environments, but still requires messages to be exchanged across these servers (as the vehicles are driving on the same physical road, so need to receive the same information). A second use case where interconnectivity is required, is when geoserver deployments for adjacent geographical areas need to have part of each other's data to be able to provide this data to vehicles *before* they enter the area of the other geoserver. This will occur within the network of a single operator with separate MEC instances, but also in cross border situation. The latter is the focus of the NL Trial site, where an artificial cross border situation is created between the KPN and TNO 5G networks.

The challenge of the interconnection is to ensure efficient exchange of all relevant messages, and prevent messages ending in a loop going back and forth between the geoservers. An additional constraint is to prevent stations from sending messages via the network, get their own messages returned: this has negative effects on the network performance and would need to be taken into account carefully in applications in end-stations. Standardisation on this topic is in progress, both in ISO (ISO TS 21184) and in ETSI (ETSI GS CIM 009, V1.2.1. 2019-10).

The AMQP protocol was selected to realize the interconnection between the geoservers. The advantage of AMQP above MQTT is that it has functions to add more metadata to every message than only a single topic, whereas its architecture is very close to MQTT. This means that the MQTT topics can be transparently distributed via the AMQP message exchange, and that any filtering that is required, can be done both at the MQTT and at the AMQP interfaces. The overall architecture is depicted in Figure 112 .

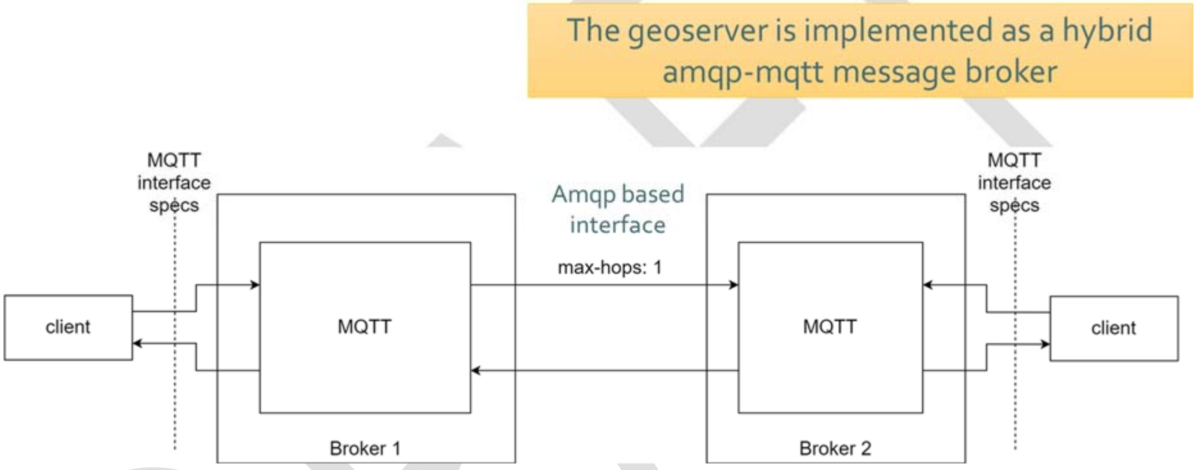

The geoserver is implemented as a hybrid amqp-mqtt message broker

Figure 112: Deployment architecture of multiple geoservers

The AMQP interfaces can be configured statically by subscribing to the area of interest (as defined in the MQTT topics) and message types. By setting the max-hops parameters of these connections, it is ensured that messages will not go back and forth between the servers, and also ensure that vehicles will never get their own messages back. It is also possible to dynamically create these interconnections based on subscriptions from clients on the MQTT side. Depending on the number of clients, this can be more efficient, as only information is exchanged that is actually required by the clients. If multiple geoservers need to be interconnected, a complete mesh can be created, as shown in Figure 113.

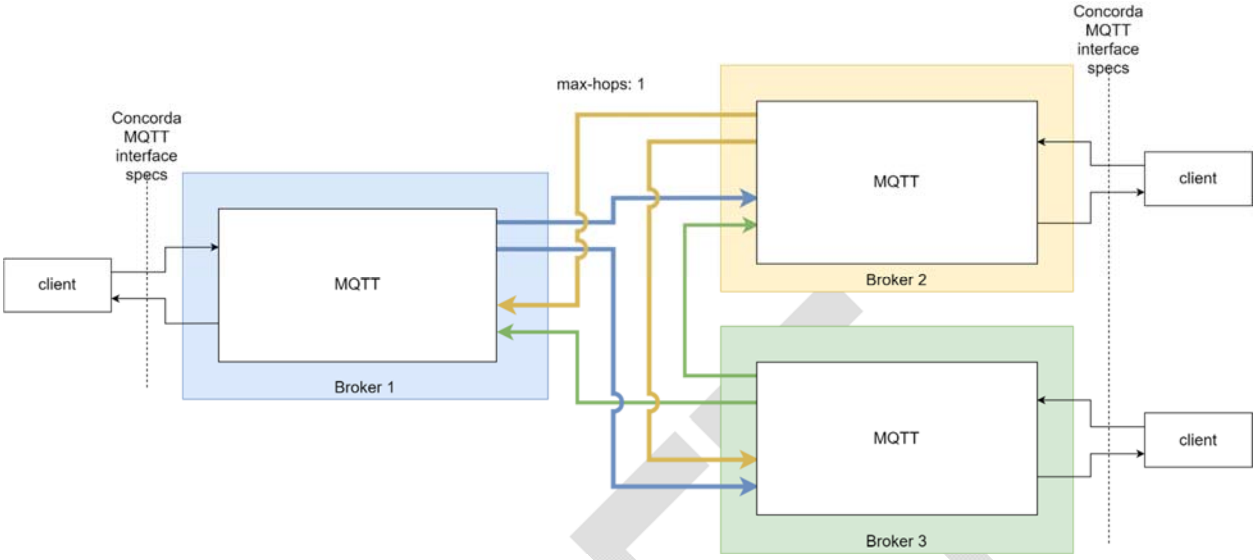


Figure 113: Interconnectivity of multiple geoservers

This solution has been implemented in geoservers deployed in the Siemens, KPN, and TNO MEC and cloud environments. At the moment it is not clear yet whether this direct interconnectivity between multiple geoservers is sufficiently scalable, or if another solution needs to be worked out. It is possible to create a hierarchical architecture with multiple layers of geoservers with only minor modifications. The most important is to configure the max-hops parameter such that messages can go through multiple layers of geoservers. The max-hop can be set at the originating geoserver, but it could also be modified at higher layer geoservers. For 5G-MOBIX these higher layer geoservers are not required, and the fully meshed direct connectivity will be sufficient and will result in the best performance.

The intermec AMQP protocol can be implemented in the Rabbitmq brokers used in the NL Trial site for the MQTT brokers based on a (proprietary) function called Federation[8]. The communication between the brokers is fully compliant with the AMQP 0.9.2 standard, and follows our AMQP interface definitions, but the federation function implements a dynamic subscription based on the subscriptions of the clients.

Also, for this function, the advantage of using identical, containerized functions prevent interoperability issues during deployment in the three different locations.

Cooperative Collision Avoidance

The CoCA user story is implemented in the vehicles, and does not need Roadside, or cloud application functions. It uses the MCM for communication between the vehicles and the MEC. Two different scenarios will be demonstrated: with an application on the MEC, which provides advices to vehicles, such as speed

[8] https://www.rabbitmq.com/federation.html

advice and lane change in order to avoid collisions, and without an application on the MEC, in which vehicles negotiate the maneuver between each other. The CoCA use case does depend on the message distribution service provided by the geoserver. This function has been described already in detail in the previous section. For the CoCA user story, however, the communication is between vehicles, instead of between vehicles and roadside sensors. This means that, although the same deployment is used, the communication is confined in the geoserver deployments in the MEC of KPN and/or TNO (depending on whether cross operator tests are performed, or single operator tests).

10.4.5. Infrastructure Integration Activities

The integration has been performed in a layered approach. First, the network connectivity is realized between the 5G networks of KPN, TNO, and Tue, and the cloud environment of Siemens. On top of that, the execution environments are integrated, which involves the geoservers. Lastly, the integration of application functions is realized.

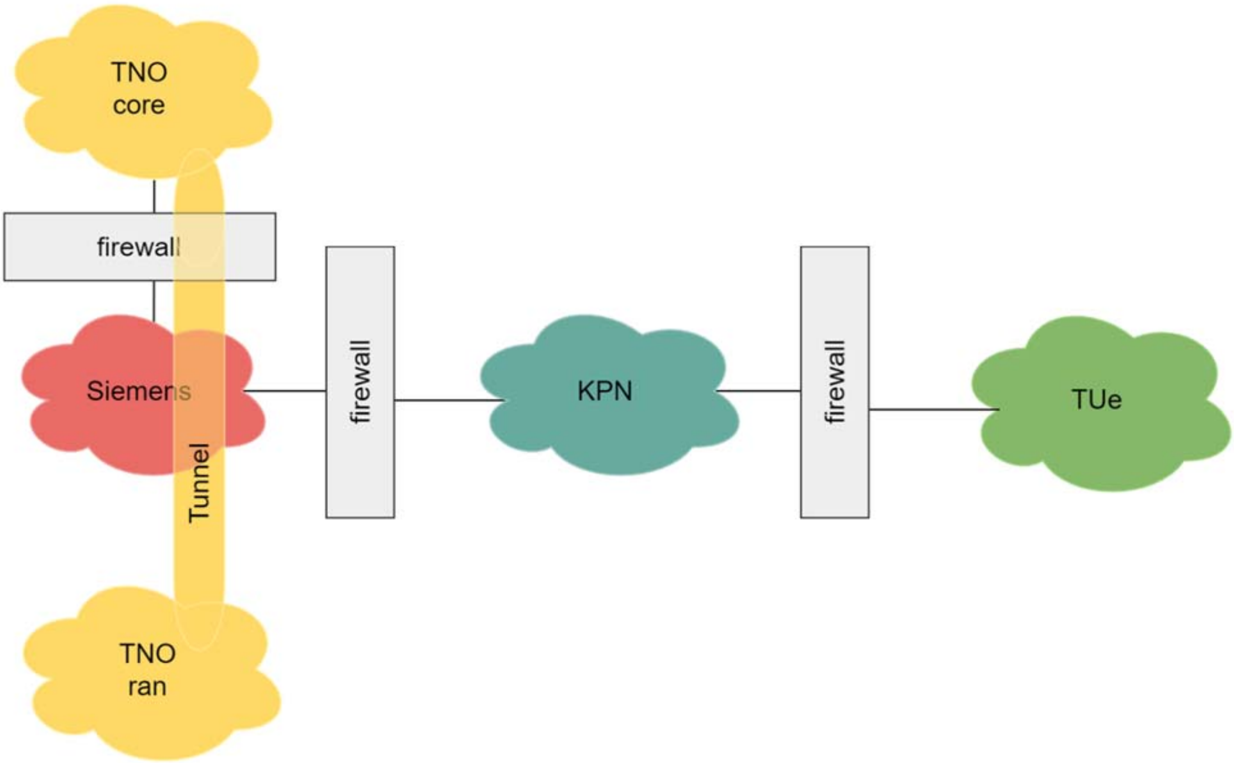


Figure 114: Network layer integration

As illustrated in Figure 114, the different networks of the different partners have been connected with each other, with proper firewalls in between the networks. A transparent tunnel between the TNO RAN and Core networks is realized via the Siemens network. All other networks are fully independent from each other. The firewalls all operate at OSI layer 4, which means that interconnectivity is arranged at application level.

At the level of the execution environments, three geoservers are deployed, exactly in line with Figure 113: one geoserver in the Siemens cloud, one in the KPN network, and one in the TNO network.

At the application level, all functions for the Extended Sensor and Remote driving user stories have been deployed in the various networks and execution environments, see Figure 115. The bottom left part of the diagram depicts the functions for the remote driving user story, the top and right part the functions for the Extended Sensor user story, and the right part of the CoCA user story. The colours refer to the different responsible partners. The vehicles are only schematically indicated.

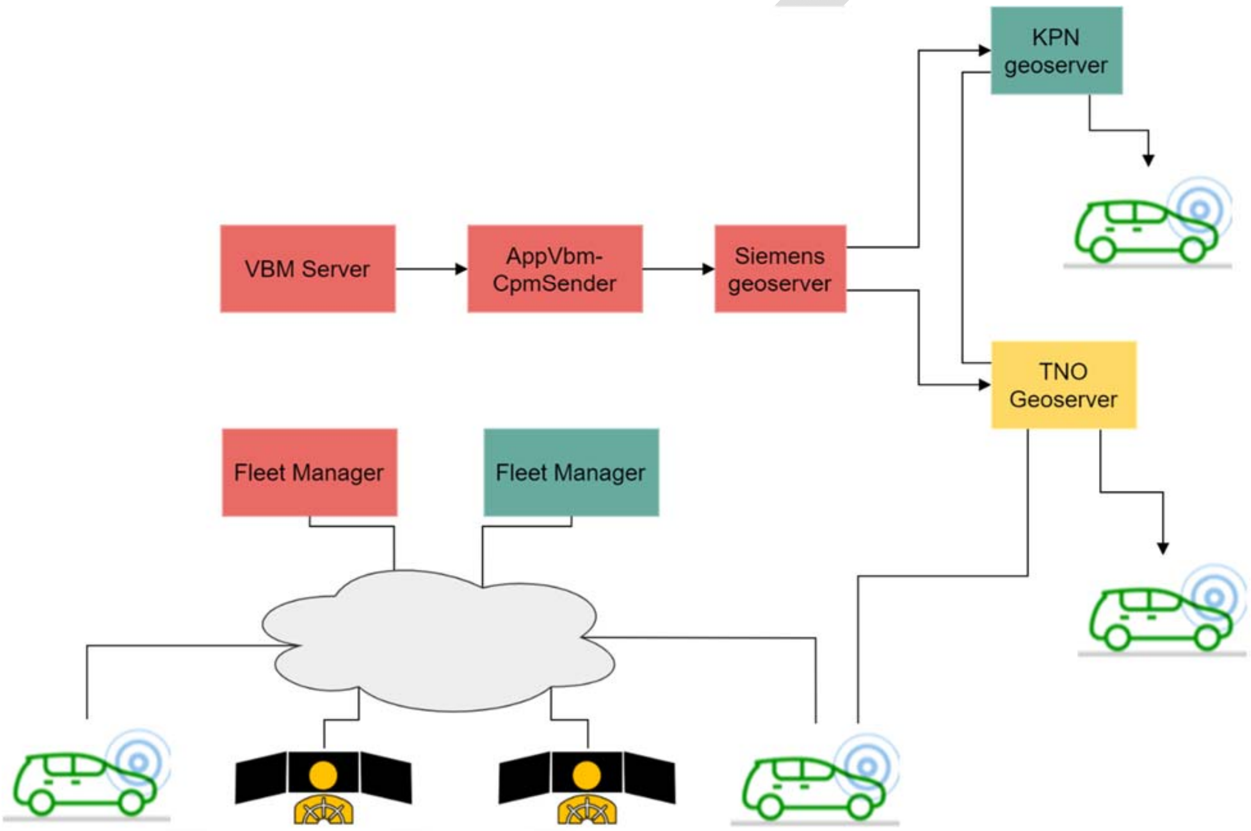


Figure 115: Overview of the integration of the different components for the three user stories

10.4.6. Infrastructure Testing Activities

The testing activities have focussed on the connectivity at the application level for the different user stories. For the Remote driving user story, the interconnectivity between the remote driving stations, the Fleet Managers, and the vehicles has been verified.

The results of testing the VBM sensor and CPM message generation are shown in Figure 116, where the coloured dots indicate the detected vehicles, where the colours indicate the velocity. On the right, 2 statistical plots are shown, used to monitor the overall performance of the VBM system and the individual cameras, and in Figure 117, where the blue squares indicate the object locations included in the CPM messages, while the green circles indicate the (virtual) sensor locations. Note, that the blue squares fully overlap the underlined coloured circles of the VBM detections.

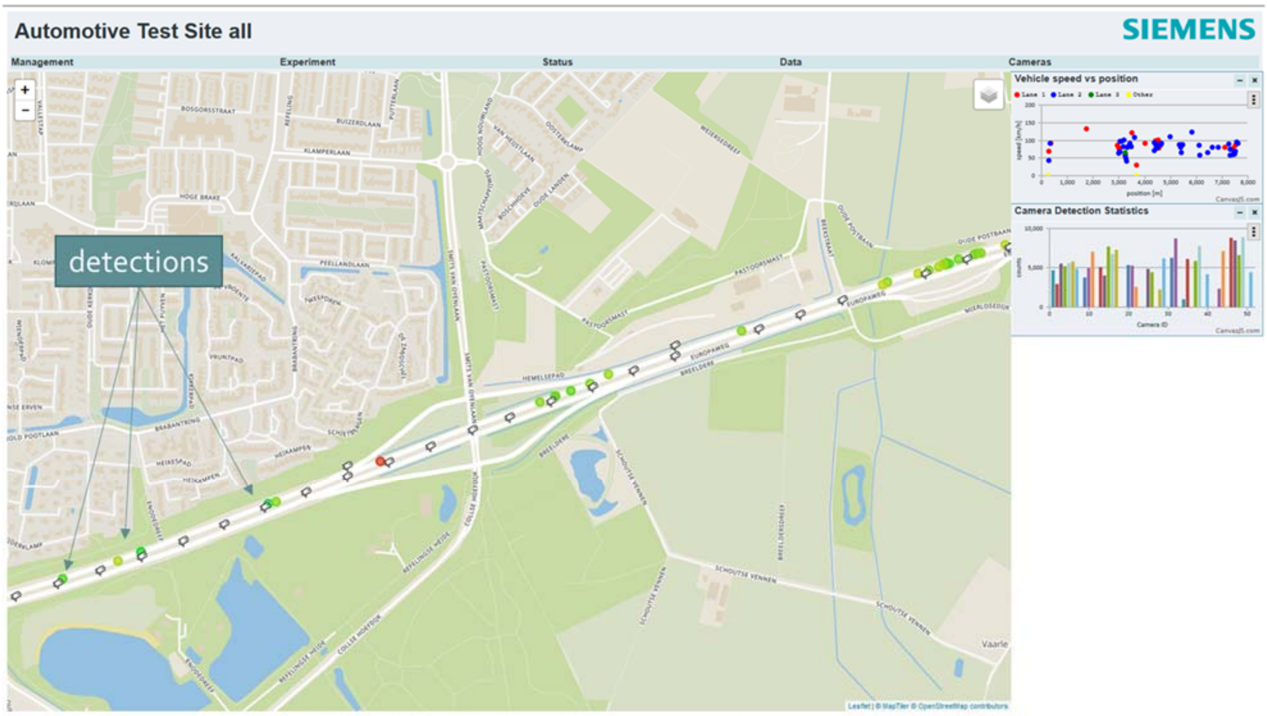


Figure 116: Representation of the VBM detection, projected on a map of the NL Trial site

Figure 116 and Figure 117 show snapshots of the tests performed for the AppVbmCpmSender application. As this is a real-time interface, it has been visually confirmed that all VBM messages are included in the CPM messages.

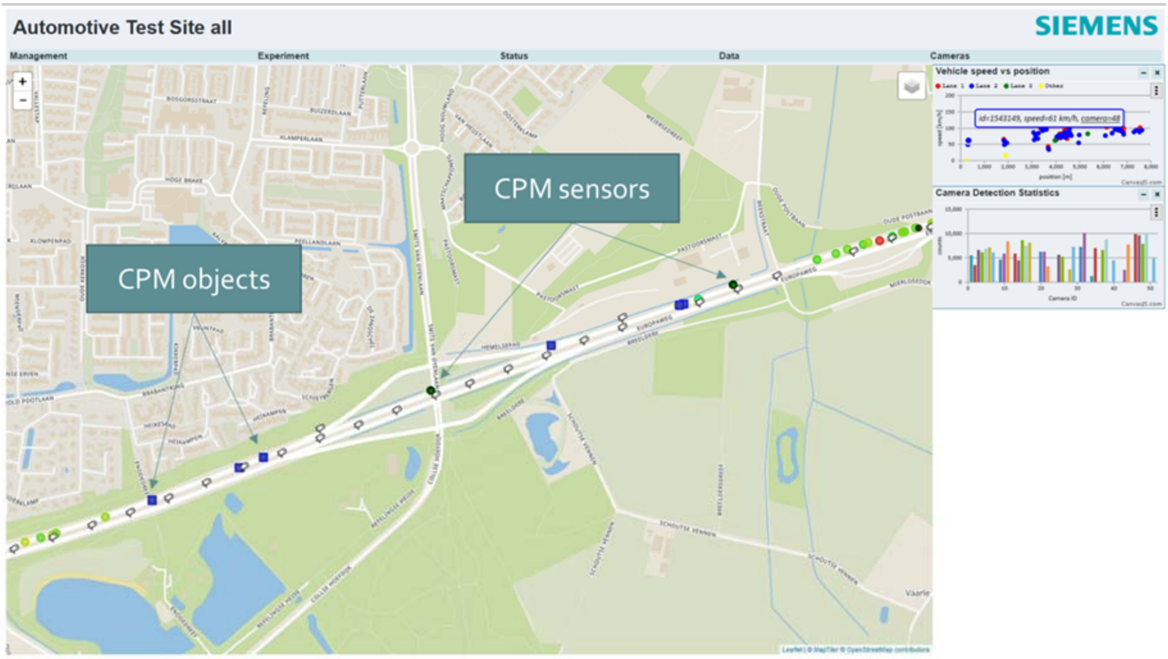


Figure 117: Overlay of the generated and transmitted CPM messages

Also, some preliminary tests have been performed where the CPM messages are received by test vehicles, and offline compared with the VBM detections. It was observed that a small fraction of the VBM detections is not included in the CPM messages. This is due to the fact that a stringent maximum latency between detection and message timestamp is allowed in the CPM V0.15 standard. It is still under consideration if and how to resolve this issue. It can be easily resolved by adjusting the (draft) standards to also allow larger latencies, but application-level testing will need to confirm first if that is useful at application level to do.

10.4.7. Future Work

Infrastructure Support for Full Trials (Phase 5)

Operational support will be provided for the phase 5 trials. No updates or upgrades to the infrastructure are expected in this stage of the project.

10.5. Trial Site CN Roll-out

10.5.1. Introduction

The CN site is currently constructing a test site for intelligent networked highways (2020). The digital, networked, and intelligent transformation of 26 kilometers of roads and the construction of some simulated urban scenes has been completed. The experimental field has been divided into a comprehensive test area, an autonomous driving and vehicle-road collaborative test area, a smart bridge and tunnel test area, and an extreme scene test area.

CN site has completed 2km highway in the north of Miao mountain, with three full-width gantries with a spacing of 500 meters and an eagle-eye camera. Also, mechanical and electrical equipment such as roadside LiDARs and V2X roadside units are installed with 5G, LTE-V, DSRC, EUHT, Wi-Fi, 10G network, and other heterogeneous networks for 3D high-precision map and other vehicle-road collaborative applications

10.5.2. Overview of Infrastructure Components

The overall architecture and infrastructure components of the CN TS are depicted in Figure 118.

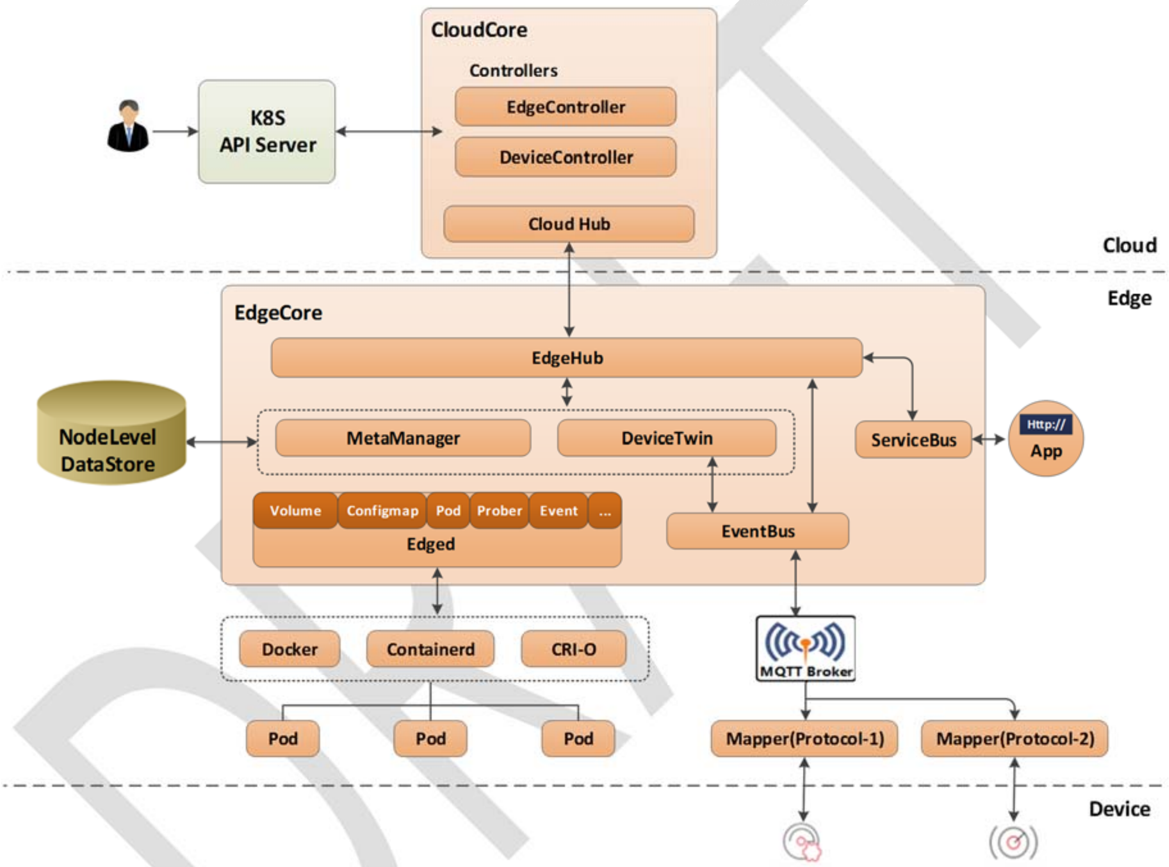


Figure 118: Overview on the architecture of the CN TS

- **Edged:** an agent that runs on edge nodes and manages containerized applications.

- **EdgeHub:** a web socket client responsible for interacting with Cloud Service for edge computing (like Edge Controller as in the KubeEdge Architecture). This includes syncing cloud-side resource updates to the edge and reporting edge-side host and device status changes to the cloud.

- **CloudHub:** a web socket server responsible for watching changes at the cloud side, caching and sending messages to the EdgeHub.

- **EdgeController:** an extended Kubernetes controller which manages edge nodes and pods metadata so that the data can be targeted to a specific edge node.

- **EventBus:** an MQTT client to interact with MQTT servers (Mosquitto), offering publish and subscribe capabilities to other components.

- **DeviceTwin:** responsible for storing device status and syncing device status to the cloud. It also provides query interfaces for applications.

- **MetaManager:** the message processor between edged and EdgeHub. It is also responsible for storing/retrieving metadata to/from a lightweight database (SQLite).

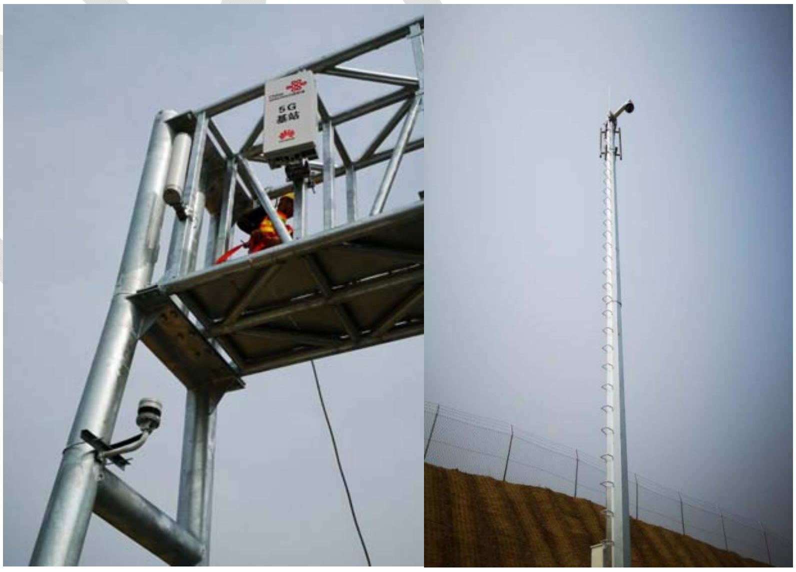
- **Hardware:** we have assembled V2X modules (including RSU and OBU) on the test site. The current delay in the delivery of 5G modules has caused developmental delays.

10.5.3. Description of the implemented infrastructure

The site in China mainly includes SDHS and SDIA, and our main facilities are located in the SDHS site. The SDHS part of CN site includes four test areas as following.

1. Comprehensive test areas (see Figure 119):

This area covers about 5.5 kilometers, mainly including simulated urban scenes and some temporary test scene constructions.



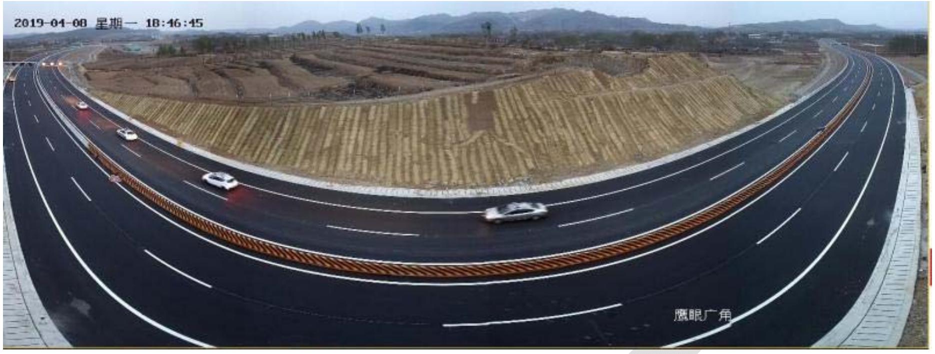


Figure 119: Comprehensive test areas

2. Autonomous driving and vehicle-road collaborative test areas (see Figure 120)

This area covers 10 km, with multiple bends and up-down slopes, mainly for automatic highway driving and continuous functional verification and testing of the CCAM (vehicle-road coordination) system. The CCAM applications mainly involve automatic driving (Autopilot), formation driving (platooning), C-V2X and hazardous event handling functions.

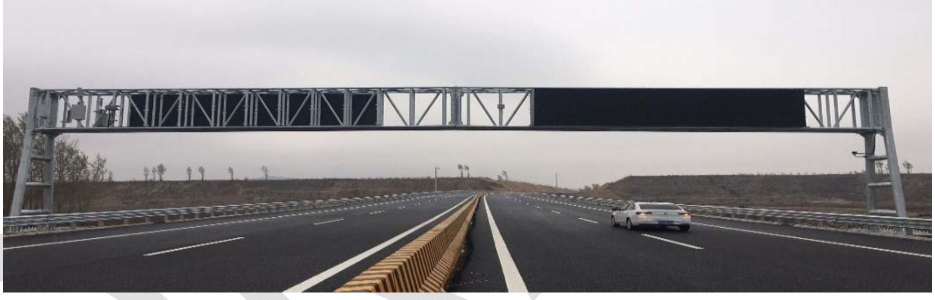


Figure 120: Autonomous driving and vehicle-road collaborative test areas

3. Smart bridge and tunnel test areas

High-precision positioning, wireless communication, and traffic condition sensing equipment has been deployed in these areas to carry out tests and verifications of the high-speed passage of typical autonomous driving tunnels, handling dangerous traffic incidents in tunnels and bridges, and collaborative applications of tunnels, bridges and vehicles.

4. Extreme scene test areas

This area is about 5 km, which mainly tests smart cars' autonomous driving performance in extreme weather and road conditions.

10.5.4. Enhance the road infrastructure to meet the requirements of 5G-MOBIX use cases

In SDHS part of CN site, we have built a multi-mode heterogeneous vehicle networking system covering 26 km of roads, toll stand-offs, and chain tunnels. This system can meet the high speed between V2V/V2I/V2P Low-latency data communication and high-definition video transmission. In SDIA part of CN site, we deployed the 5G module device.

Integrate the road infrastructure with 5G network

To conduct the intelligent roadside perception system, in SDHS part of CN site, we deployed intelligent RSU in the vehicle-road collaboration and autonomous driving test area, including LiDAR, millimeter-wave radar, panoramic monitoring, edge server, V2X equipment, variable speed limit signs and other equipment to achieve accurate real-time perception, calculation and release of the road traffic environment.

In SDHS part of CN site, we made 3D digital modelling, established high-precision road and scene models, and combine real-time sensing and fusion data to provide dynamic high-precision map navigation and traffic target information for autonomous vehicles.

Evolution of platforms with extended toolboxes, applications, and interfaces

In SDHS part of CN site, we deployed the smart road network management and cloud control platform in the original bridge and tunnel management office. The platform's monitoring scope covers the test area and office and living areas, forming a complete test area supervision, scheduling, data storage and central computing system.

The monitoring center can prompt test vehicles and personnel about the use of test sections, release information about major emergencies, strengthen the monitoring and management of test vehicles, and prompt test users to conduct tests in a standardized manner, thus reducing the probability of dangerous accidents, and improving safety. The monitoring center can also perform timely supervision and early warning of various emergencies in the test base and do a good job in related dispatching and command work to effectively guarantee the test order of the test base. The monitoring center will realize the real-time upload, analysis and processing of the running data of the tested intelligent networked vehicles through the communication technology of the vehicle network, the remote monitoring technology, the vehicle bus technology, and the vehicle Ethernet technology.

Specification of stakeholders, roles and resources/efforts

- DUT makes the coordination and organization of the entire project, responsible for 5G technology research and application, coordinating related test cases of various road sections, integrating and integrating other parties' autonomous driving and V2X technologies, and developing corresponding autonomous driving applications.

- SDIA provides self-driving road sections, building corresponding 5G application environments, providing self-driving vehicles, testing related use cases and formulating related standards.
- CNHTC provides self-driving commercial vehicles.
- SDHS provides 5G road sections for autonomous vehicles, including 5G base stations and road-side equipment.
- DATANG provides C-V2X and 5G module device.

10.5.5. Infrastructure Integration Activities

We have completed the construction of the test environment and most of the related test equipment (5G base station, remote control center, server, RSU, OBU, etc.). We have carried out related simulation experiments (modelling, testing) under Veins, and are preparing to transplant the test program to actual equipment, MEC server, vehicle-mounted industrial computer, etc.

10.5.6. Infrastructure Testing Activities

We have carried out a test of remote driving and can carry out video returns. We tested the following KPIs: User experienced data rate, End to End Latency, Control plane Latency, User plane Latency, Position Accuracy. After some simple verification tests, we found that the SDIA part of CN site is competent for this project's test work.

Extended Testing & Impacts on trial plan

The research and testing of unmanned buses expands the original test plan's scenarios and further deepens the application scope of 5G-V2X.

10.5.7. Future Work

Open issues and Extended Implementation

The unmanned bus based on 5G vehicle-road collaboration is what we want to promote next.

The driving routes and areas buses are relatively fixed, with a simple environment and low speed. Therefore, it is easier to take the lead in promoting driverless technology in buses.

10.6. Trial Site KR Roll-out

10.6.1. Introduction

As described in D2.1 (5G-MOBIX, Deliverable D2.1: 5G-enabled CCAM use cases specifications, October 2019), KR trial site is targeting to verify the following use cases:

- (UC category 4) Remote driving: This use case category has a potential impact to the cross-border corridor especially related to the V2X continuity (AC1). KR trial site focuses on remote driving using an mmWave 5G NR communication system with e network slicing technology.
- (UC category 5) Vehicle quality of service support: This use case category has potential impact to the cross-border corridor especially related to the issues of low coverage handover (TH2) and V2X continuity (AC1). KR trial site focuses on Tethering via vehicle using an mmWave 5G NR communication, which is required to provide Internet connectivity to a vehicle relay through an mmWave-based wireless backhaul. Types of base stations (BSs) include macrocell BS, microcell BS, and BS-type roadside unit (RSU). A UE-type RSU can also be a part of the infrastructure for wireless backhaul connectivity.

In order to verify the applicability and efficiency of the above use cases, KR trial site has implemented and integrated the road and cloud infrastructure, as described in the following sections in detail. Cross-border aspects to guarantee user QoS in a vehicle are also discussed.

10.6.2. Overview of Infrastructure Components

The main components of the KR-TS are mmWave OBUs, 5G network mainly composed of gNBs and 5G core network, and a V2X application server located at the remote driving center for remote driving support, as illustrated in Figure 121. A gNB consists of one or multiple centralized units (CUs) and one or multiple distributed units (DUs) deployed along the roadside, and MAC-PHY functional split is adopted so that CUs and DUs are responsible for the processing of higher-layer protocols and the processing of physical layer and RF, respectively. Optical fiber is used for the fronthaul connectivity between CUs and DUs, and each CU is further connected to the 5G core network that is also interconnected with the public Internet network and the V2X application server located at the remote driving center.

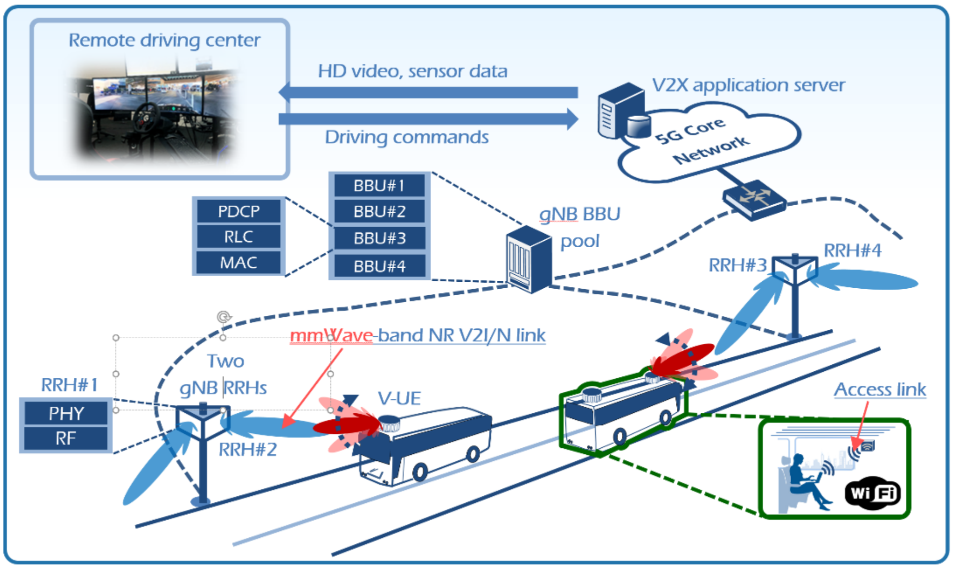


Figure 121: Overview of infrastructure for KR TS

10.6.3. Infrastructure Deployment and Enhancement Activities

Enhance the road infrastructure to meet the requirements of 5G-MOBIX use cases

For the Tethering, via Vehicle and remote driving user story, several requirements were identified in D2.2 (5G-MOBIX, Deliverable D2.2: 5G architecture and technologies for CCAM specifications, 2019) to provide high data throughput backhaul service up to several Gbps to a vehicle: network slicing, seamless handover, and beam switching. In order to meet these requirements, road infrastructure defined in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019) has been implemented and tested. More specifically testbeds for gNBs have been developed and the DUs of gNBs, which are in charge of physical layer and RF processing, have been installed along the test road. The higher-layer functionalities are provided in the CU which is connected to the DU through an optical fiber. The DU is further connected to the 5G Core for Internet access, and the 5G Core is also interconnected with the remote driving V2X server.

An enhancement of the road infrastructure is the employment of a synchronization unit (shown in Figure 122), which provides a reference clock and pulse per second (PPS) to each RSU so that the RSUs are all synchronized. By doing so, minimizing handover interruption time is possible using, e.g., RACH-less handover, which allows vehicle UEs to perform Neighboring cell search while communicating with the serving cell.

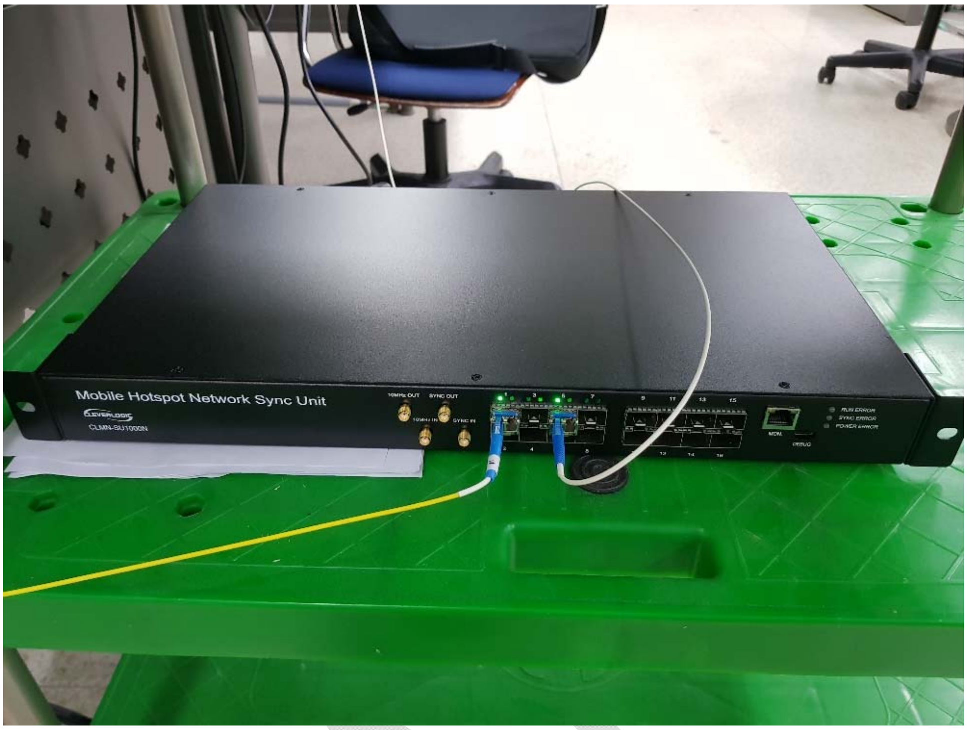


Figure 122: Synchronization unit

Integrate the road infrastructure with 5G network

Based on the roadside and cloud infrastructure of the KR trial site as described in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), the roadside infrastructure for Tethering via Vehicle and Remote Driving user story is being integrated with 5G network. The testbed implementations for the integrated road infrastructure are illustrated in Figure 123 and Figure 124. Both control plane and user plane protocols have been integrated across L1/L2/L3 functions. The integration of the 5G radio access network with the 5G core network has also been provided to provide diverse services to the users in a vehicle.

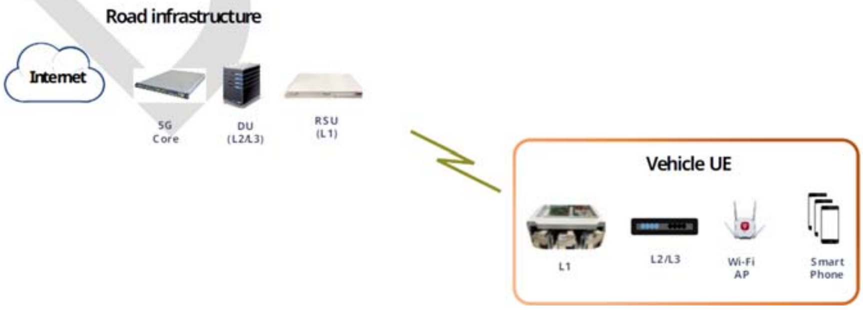


Figure 123: Testbed implementation of tethering via vehicle user story for integrated road infrastructure

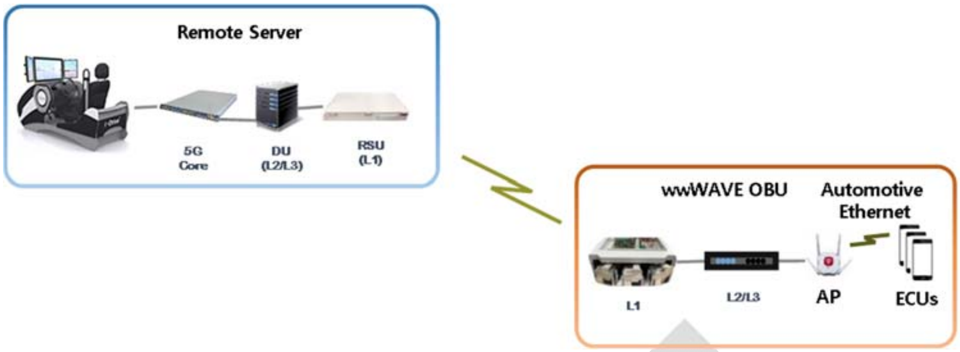


Figure 124: Testbed implementation of remote driving user story for integrated road infrastructure

10.6.4. Platform and Apps Evolution

Application for tethering via vehicle

For the KR TS, Benchbee is chosen to produce the metrics needed to calculate the network capabilities and the KPIs. Benchbee is a tool designed to test mobile network speed. Network performance such as ping, downlink data rate and uplink data rate can be measured by Benchbee, as seen in Figure 118. In the KR TS, especially for the tethering via a vehicle use case, typical Wi-Fi devices are smartphones, and these are used to test the network performance. For the performance measurement, the Benchbee App is installed on a smartphone and a Benchbee server is connected to the core network side. Therefore, the measurements cover all the network components such as the core network, gNB, vehicle UE, Wi-Fi AP and end-user devices.

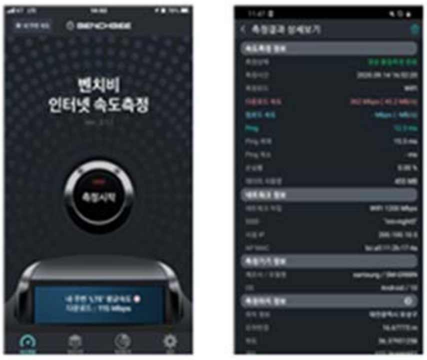


Figure 125: Benchbee network performance testing app

Application for remote driving

Around View Monitoring (AVM) is a system that provides the images of the situation outside the car in real time through the monitor inside with a total of four cameras mounted underneath the outside mirrors, in the front and the back of the test vehicle. It provides the highest safety and convenience to drivers by supporting 3D views of areas around vehicles as if seen from the sky using four HD cameras installed in the front, rear, right and left rear-view mirrors, as depicted in Figure 119.

In this system, a total of four cameras, including one on the front and back of the vehicle as well as one on each outside mirror on the side of the vehicle are attached to provide a 360-degree view to the driver using a composite video with a real time distortion correction. Through the 4 cameras installed on the vehicle, it can view front, back, left and right side. When the vehicle is started, if the gear selector is put into the R position, it automatically starts the AVM. If the gear is changed from reverse to parking position, AVM operation stops. This system is a supplemental system only. It is the responsibility of the driver to always check the area around the vehicle before and while moving.

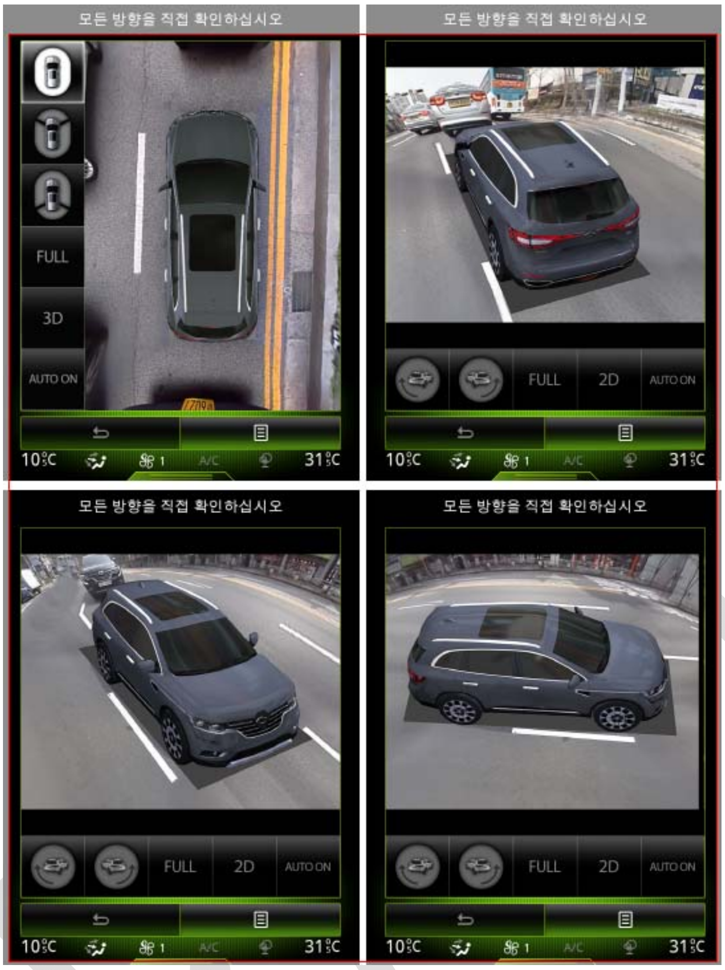


Figure 119: AVM system for remote driving application

10.6.5. Infrastructure Integration Activities

The architecture for the implemented infrastructure of Tethering via Vehicle user story is provided in Figure 120, where a synchronization unit is added to provide network synchronization among the RSUs. In the original architecture in D2.3 (5G-MOBIX, Deliverable D2.3: Specification of roadside and cloud infrastructure and applications to support CCAM, 2019), the synchronization unit was not included. The synchronized operation of RSUs can provide more robust handover and lower handover interruption time by enabling neighbour cell measurement while transmitting/receiving data.

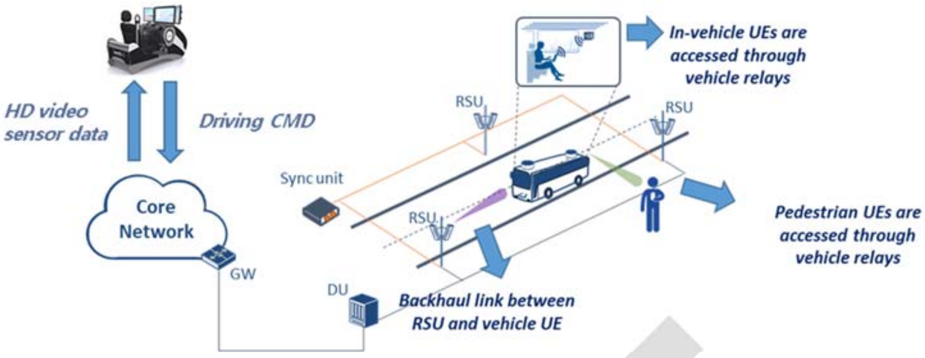


Figure 120: Infrastructure Integration Activities

10.6.6. Infrastructure Testing Activities

Preliminary indoor test was conducted to verify basic RSU functionalities including cell search, random access, channel sounding, control/data modulation/demodulation. Figure 121 shows the RSU-L1 setup in an indoor test environment. Through indoor PHY-to-PHY test, the maximum downlink throughput of 3 Gbps and the maximum uplink throughput of 200 Mbps was verified.

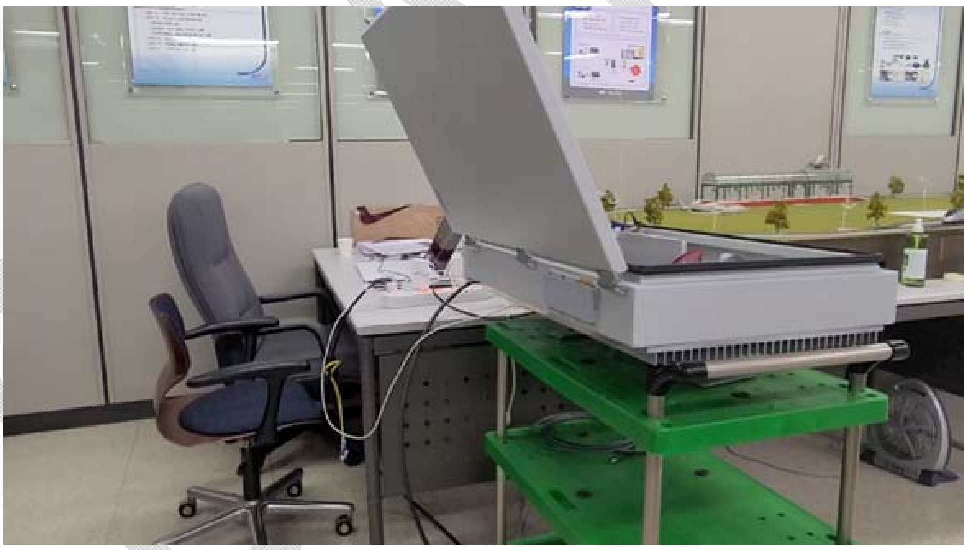


Figure 126: RSU-L1 setup in indoor test

The next step is to verify the integrated operation of L1/L2/L3 and 5G core. The physical layer processing and higher-layer functionalities were integrated with the 5G core (see Figure 127). Main 5G core functions such as registration/deregistration were verified. In addition, actual user services such as 4k YouTube video streaming (see Figure 128) and smartphone speed test using BenchBee were tested. It was shown that stable and non-interrupted video streaming was possible. In addition, around 400 Mbps downlink internet connection speed was recorded.

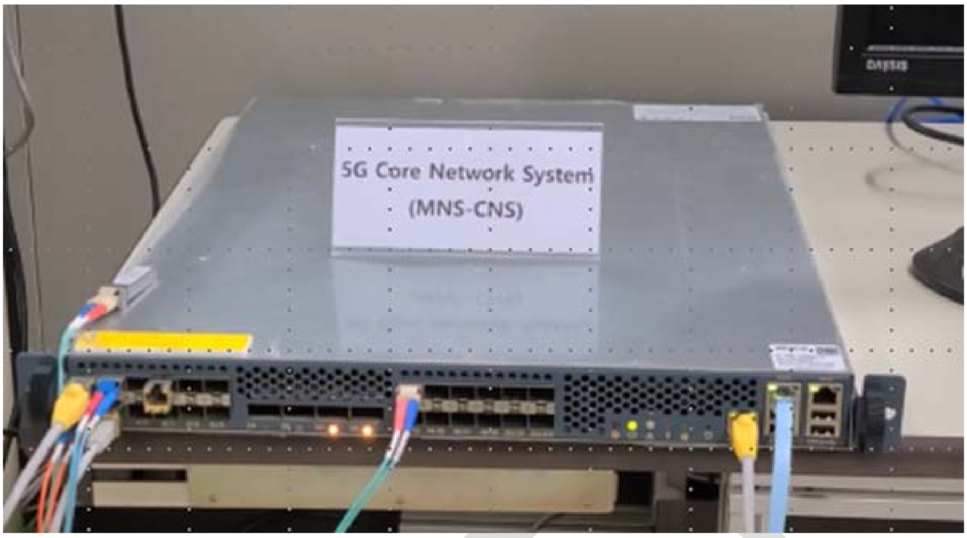


Figure 127: 5G core

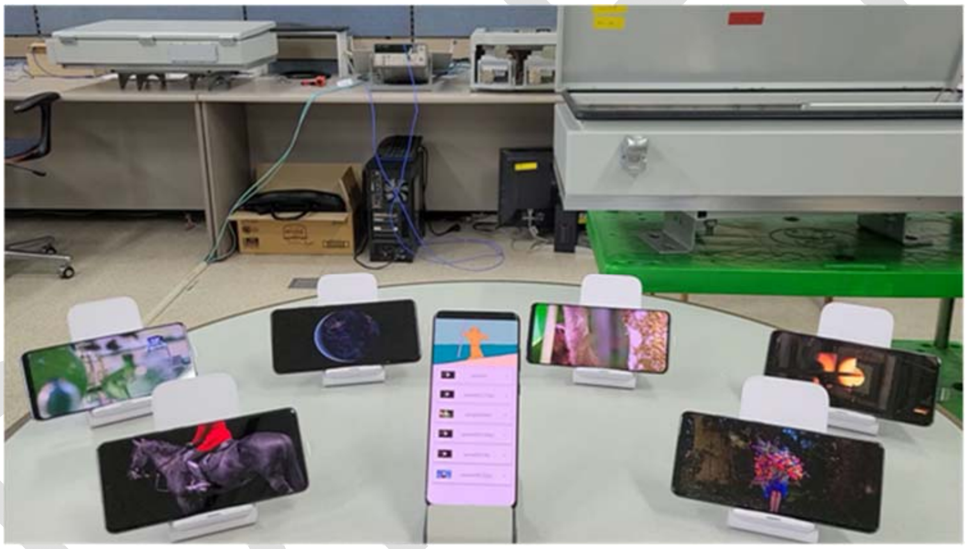


Figure 128: Video streaming test

Further RSU functionalities were tested as follows:

- CQI/RSRP reporting: downlink channel state was measured through channel state information reference signal (CSI-RS) and synchronization signal block (SSB). The measurement results were reported using uplink control channel (i.e., PUCCH). The measurement was performed with a periodicity of ten milliseconds.
- Link adaptation: the modulation coding scheme (MCS) was controlled based on the CQI report provided by the UE. Using this CQI report, the RSU can adaptively determine the MCS values in the L2 scheduler. It was confirmed that the MCS values are properly determined based on the measured CQI values.

- Timing adjustment: the uplink timing advancement (TA) can be done during the random-access procedure and during the transmission using uplink sounding reference signal (SRS). It was verified that the TA values are properly calculated and updated.

Outdoor tests for Tethering via Vehicle were conducted on Highway 45 of Korea, between Yeoju JCT and Gamgok IC. Test track is 2.7 km long and is covered by 5 RSUs (numbered from 1 to 5). Each RSU is installed 15 above the ground. The beam from each RSU is steered towards the highway. The RSU deployment on a highway test track is depicted in Figure 129.

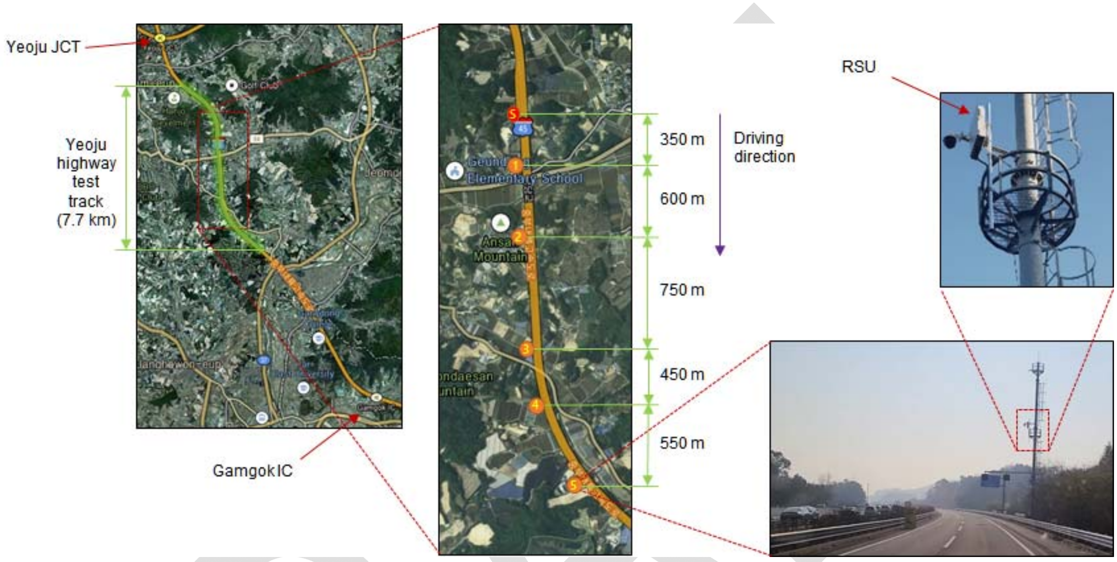


Figure 129: RSU deployment on a highway test track for Tethering via Vehicle scenario

The UE is installed on a test vehicle moving on a highway test track. The UE has a beam switching capability which can select the best transmit/receive beam at the UE. This beam switching can be beneficial since the signal quality can be maintained even when test vehicle changes the driving direction. The UE deployment on a test vehicle is depicted in Figure 130.

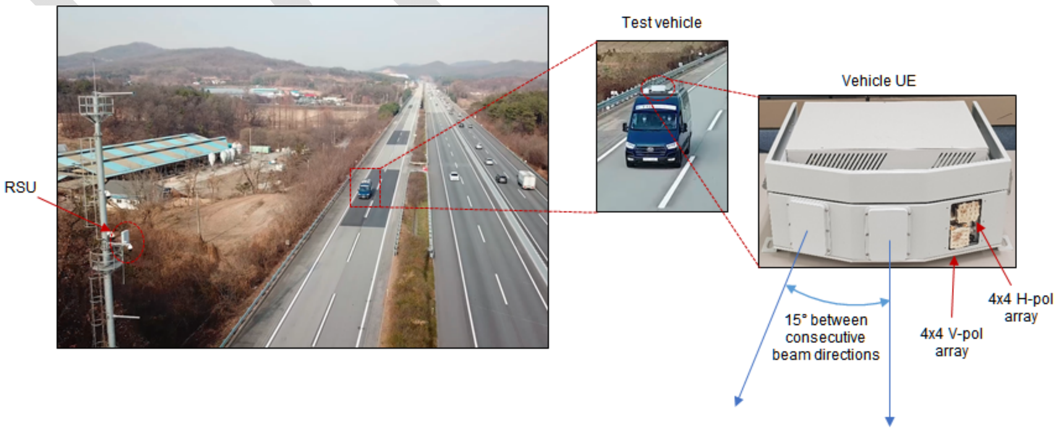


Figure 130: UE deployment on a test vehicle

It was shown through the highway test that mean SNR of 12.47 dB and mean data rate of 1.581 Gbps can be achieved. The handover operation involving the 5 RSUs are shown to be successful.

10.6.7. Future Work

Further functional enhancement and system optimization will be conducted for the trial of Tethering via Vehicle. More robustness is expected by further implementing and testing the features like hybrid ARQ (HARQ), adaptive modulation and coding (AMC) scheme, etc.

10.7. Installation guides and technical details

10.7.1. ES-PT CBC installation guide and technical details

COCKPIT HARDWARE REQUIREMENTS AND COMPONENTS

The following equipment is required to setup the cockpit infrastructure into the remote driver premises:

a. **Microsoft Windows 10 based PC**

- ✓ Intel i7 or better
- ✓ NVIDIA GTX1070 GPU (or better) graphic card
- ✓ 32 GB RAM
- ✓ 500GB or better SSD DISK
- ✓ 4K Display 32" or better

b. **Oculus Rift Headset (see Figure 126, https://www.oculus.com/rift-s/?locale=en_US)**

Figure 126: Oculus Rift Headset

c. **Logitech G29 Steering Wheel and Pedals (see Figure 127,** https://www.logitechg.com/en-roeu/products/driving/driving-force-racing-wheel.html)

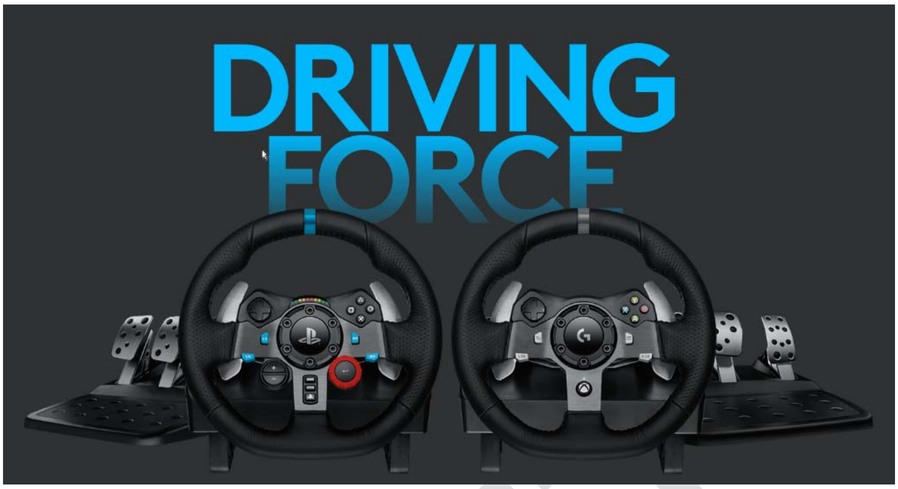


Figure 127: Logitech G29 Steering Wheel and Pedals

d. **Logitech Driving Force Shifter (See Figure 128,** https://www.logitechg.com/en-roeu/products/driving/driving-force-shifter.html).

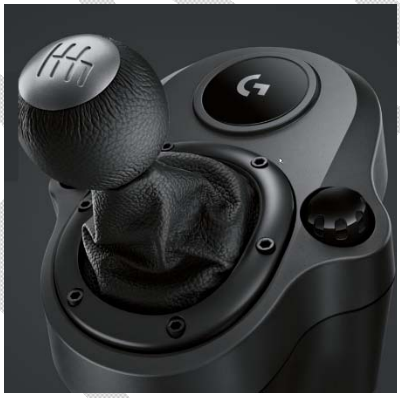


Figure 128: Logitech Driving Force Shifter

COCKPIT SOFTWARE REQUIREMENTS

a. **Recent NodeJS installation**

v11 or newer is recommended. Please check https://nodejs.org/en/download/

b. **Oculus driver version**

To set up the Oculus Rift S or Rift, one needs to download the Oculus app to you're the target PC. Please check https://www.oculus.com/setup

c. NVIDIA drivers, please check:

http://www.nvidia.com/Download/Scan.aspx?lang=en-us
https://www.geforce.com/geforce-experience/download

COCKPIT connectivity requirements

The cockpit will require having direct connectivity with the MEC which is in the same country as the cockpit physical location.

Somehow, it should also require access to a piece of software which needs to be accessible from MECs in both countries and takes care of vehicles and remote drivers registration. We will refer to this piece of software as RDRS (Remote Driving Registry Server).

Logitech connection

- Logitech Shifter should be connected to the wheel using the cable which is devoted to such purpose.
- It is important to connect the wheel to the external power supply, otherwise it will be detected but non-functional.
- The wheel should be configured with the switch on its top part set to PS3.
- Please be sure that it is configured in such way, otherwise, the events will not be correctly processed by the PC.

LOCALHOST EXECUTION FOR SANITY CHECK OF THE ENVIRONMENT

In order to check that the whole setup is working fine a localhost execution in simulation mode can be performed.

1. Go to config directory and edit the following files:

-config/MECs.json

```
                    [
                 {“MEC_NAME”  :  “MEC_TEF_001”  ,  “MEC_IP”  :  “127.0.0.1”,  “MEC_PORT”  :  “5001”,
“STANDBY_MEC_IP” : “127.0.0.1”, “STANDBY_MEC_PORT” : “5031”, “COUNTRY” : “SPA”},
                 {“MEC_NAME”  :  “MEC_NOS_001”  ,  “MEC_IP”  :  “127.0.0.1”,  “MEC_PORT”  :  “5031”,
“STANDBY_MEC_IP” : “127.0.0.1”, “STANDBY_MEC_PORT” : “5001”, “COUNTRY” : “POR”}
                    ]
```

-config/cellAndMECs.json

```
              [
              {“CellId”  :  “7738834”  ,  “Operator”  :  “movistar”,  “MEC_IP”  :  “127.0.0.1:5001”,
“STANDBY_MEC” : “127.0.0.1:5031” },
                       {“CellId” : “7759912” , “Operator” : “movistar”, “MEC_IP” : “127.0.0.1:5001”,
“STANDBY_MEC” : “127.0.0.1:5031”  },
                       {“CellId”  :  “2985544”  ,  “Operator”  :  “NOS”,  “MEC_IP”  :  “127.0.0.1:5031”,
“STANDBY_MEC” : “127.0.0.1:5001”},
                       {“CellId”  :  “2985548”  ,  “Operator”  :  “NOS”,  “MEC_IP”  :  “127.0.0.1:5031”,
“STANDBY_MEC” : “127.0.0.1:5001”},
                    {“CellId”   :   “9999999”   ,   “Operator”   :   “movistar”,   “MEC_IP”   :
“192.168.1.143:5011”, “STANDBY_MEC” : “127.0.0.1:5001”}
```

2. Open a PowerShell window in REF_IMPLEMENTATION directory for
   Starting RDRS server. Execute scripts/is.bat
   The server will start listening on port **4001** by default

3. Open a PowerShell window in REF_IMPLEMENTATION directory for starting
   first MEC instance of server.
   Execute:
   node bin\MEC_COMMAND_SERVER_v9k.js 127.0.0.1 4001 MEC_TEF_001

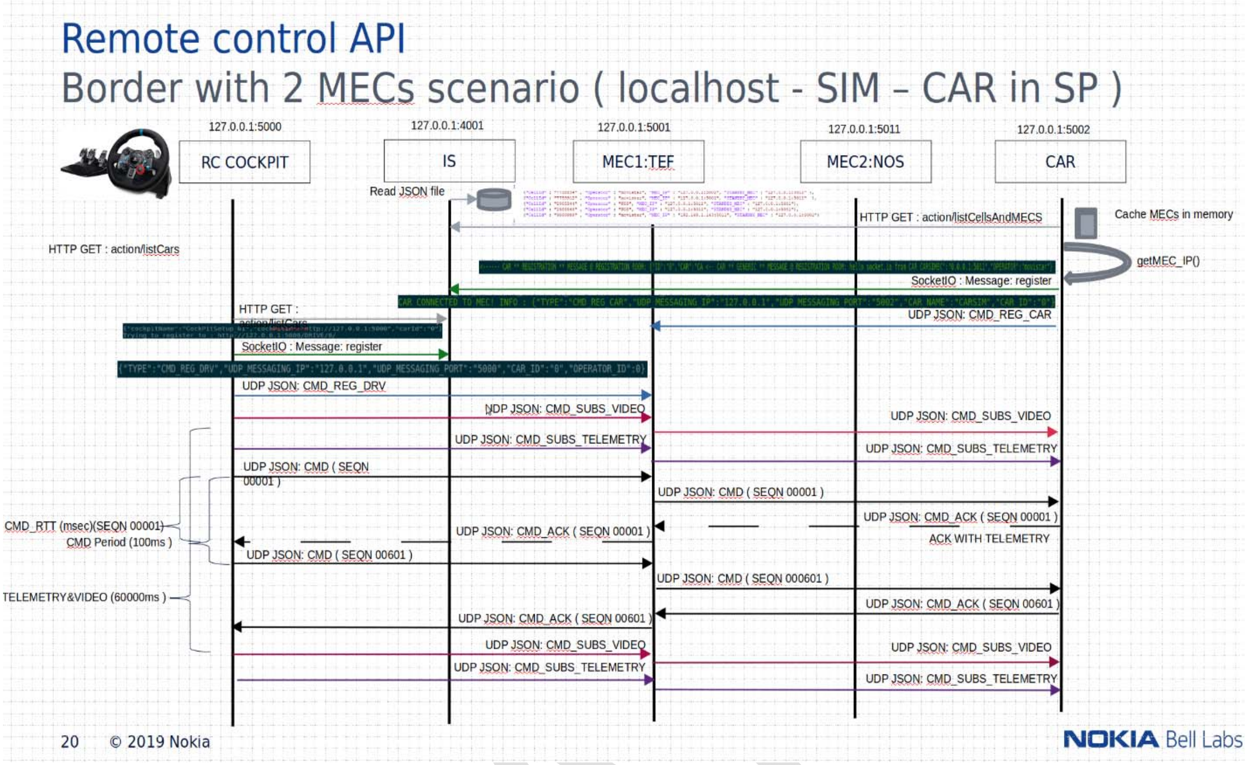
4. Open a PowerShell window in REF_IMPLEMENTATION directory for starting second MEC instance of server.
   Execute:
   node bin\MEC_COMMAND_SERVER_v9k.js 127.0.0.1 4001 MEC_NOS_001

5. Open a PowerShell window in REF_IMPLEMENTATION directory for starting a simulated car.
   Node bin/CAR_CLIENT_v9k.js 127.0.0.1 5002 127.0.0.1 5002 127.0.0.1 4001 0 CARSIM

6. Finally open a PowerShell window in REF_IMPLEMENTATION directory for starting the cockpit control application.
   Node bin\COCKPIT_v9k.js 127.0.0.1 5000 127.0.0.1 5001 127.0.0.1 4001 0 0 24

7. Check cockpit traces (a.7). As you move the gear, push pedals or change gears the CMD json contents should change reflecting the changes.

The protocol flow in localhost is presented in Figure 129.



Remote control API

Border with 2 MECs scenario (localhost - SIM – CAR in SP)

127.0.0.1:5000
RC COCKPIT
127.0.0.1:4001
IS
127.0.0.1:5001
MEC1:TEF
127.0.0.1:5011
MEC2:NOS
127.0.0.1:5002
CAR
Read JSON file
HTTP GET : action/listCellsAndMECS
Cache MECs in memory
HTTP GET : action/listCars
getMEC_IP()
SocketIO : Message: register
CAR CONNECTED TO MEC1 INFO : {"TYPE":"CMD_REG_CAR","UDP_MESSAGING_IP":"127.0.0.1","UDP_MESSAGING_PORT":"5002","CAR_NAME":"CARSIM","CAR_ID":"0"}
UDP JSON: CMD_REG_CAR
HTTP GET : action/listCars
SocketIO : Message: register
{"TYPE":"CMD_REG_DRV","UDP_MESSAGING_IP":"127.0.0.1","UDP_MESSAGING_PORT":"5000","CAR_ID":"0","OPERATOR_ID":0}
UDP JSON: CMD_REG_DRV
NDP JSON: CMD_SUBS_VIDEO
UDP JSON: CMD_SUBS_VIDEO
UDP JSON: CMD_SUBS_TELEMETRY
UDP JSON: CMD_SUBS_TELEMETRY
UDP JSON: CMD (SEQN 00001)
UDP JSON: CMD (SEQN 00001)
UDP JSON: CMD_ACK (SEQN 00001)
ACK WITH TELEMETRY
CMD_RTT (msec)(SEQN 00001)
CMD Period (100ms)
UDP JSON: CMD_ACK (SEQN 00001)
UDP JSON: CMD (SEQN 00601)
UDP JSON: CMD (SEQN 000601)
TELEMETRY&VIDEO (60000ms)
UDP JSON: CMD_ACK (SEQN 00601)
UDP JSON: CMD_ACK (SEQN 00601)
UDP JSON: CMD_SUBS_VIDEO
UDP JSON: CMD_SUBS_VIDEO
UDP JSON: CMD_SUBS_TELEMETRY
UDP JSON: CMD_SUBS_TELEMETRY
20 © 2019 Nokia
NOKIA Bell Labs

Figure 129: Protocol flow in local host

Cockpit Installation, Video Client and HMD setup with Oculus

Introduction

The final video flow setup is being evaluated at the moment of writing this document. There are several approaches regarding the video streams being received:

a. VR360 video blended with a low latency video in the front view. An HMD is required for this setup. Oculus is preferred option for HMD.

b. Three/Four low latency video streams blended into a VR360 video into a single HMD device. Oculus is preferred option for HMD.

c. Three low latency video streams displayed in three monitors as per traditional driving simulators setups.

For a and b, HMD is required as well as Nvidia 1070 graphic card. For c just 3 monitors are required. A graphic card capable of 3 video output will be required.

Video client setup

Gstreamer platform will be used to decode and perform all the steps prior to render the video.
This requires installing a full Gstreamer environment in the PC. Please follow the official installation steps as described in: https://gstreamer.freedesktop.org/documentation/installing/on-windows.html?gi-language=c

The decoding pipeline will be based on Gstreamer UDP multicast video reception with low latency configuration and H.264 video decoding.

Unity setup

In the case of using an HMD VR 360 application, it might require having a Unity installation in the PC. Moreover, a specific C# based plugin will be provided for integrating the video reception into the Unity rendering system. It is still under development such plugin integration which will be probably based on Gstreamer libraries.

In order to setup Unity, please follow the official documentation at: https://docs.unity3d.com/Manual/GettingStartedInstallingHub.html

Running the oculus application

There might be two flavours of the application. One will be an oculus application which will be either executed from unity or just using an executable generated by Unity. The other option could be an OpenGL based application which can be used as player. In the case of three different non 360 streams three players need to be executed pointed to the UDP video stream.