



5GMOBIX

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

D3.5

Report on the evaluation data management methodology and tools

Dissemination level	Public (PU)
Work package	WP3: Development, integration and roll-out
Deliverable number	D3.5
Version	V1.0
Submission date	31/01/2021
Re-submission date	N/A
Due date	31/12/2020



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

www.5g-mobix.com

Authors

Authors in alphabetical order		
Partner	Person	Contributed to sections
AALTO	Edward Mutafulungwa Norshahida Saba Oussama El Marai Xuebing Liu	Contributor to section 2.2.4 and 4.4
AKKA	Benoît Dalet Sadeq Zougari (Editor)	T3.5 task lead, definition of ToC and main document editor for D3.5. Section 1, 5 and 6 editor. Contributions to section 2, 2.1 and 3.4. Annex A and B editor
CCG	João Ferreira Joel Puga	Contributor to 2.2 and 4.1.2.2
CTAG	Marta Miranda	Contributor to 3.1, 3.2, 3.3 and 4.1
DEKRA	Oscar Castañeda	Contributor to 4.1.2.1
DUT	Hao Yu Zhiheng Yuan	Contributor to 2.2.8 and 4.8
ERICSSON	Ioannis Masmanidis Serhat Col	Contributor to 4.2.1 - 4.2.2.1 - 4.2.3 (Ericsson part) - 4.2.4
ICCS	Konstantinos Katsaros	Contributor to 2.2
IMEC	Dries Naudts Vasilis Maglogiannis	Contributor to 2.2.2, 4.2.2.2(1) and 4.2.3
ISEL	António Serrador Carlos Mendes Nuno Cota Nuno Cruz Nuno Datia	Contributor to 4.1.2.1
Nokia ES	Jaime Jesús Ruiz Alonso Raúl López López	Contributor to 4.1.2.1
SENSIBLE 4	Timo Mustonen Matthieu Myrsky	Contributor to 2.2.4 and 4.4
TNO	Bart Netten	Contributor to 2.1, 2.2.6, 3.x and 4.6
TURKCELL	Nazli Guney Murat Ünlüsan (TURKC-IL) Afrim Berişa (TURKC-IL)	Contributor to 2.2.2 and 4.2.6
TUB	Federico Murciano	Contributor to 2.2.3 and 4.3

TUBITAK	Arda Taha Candan	Contributor to 4.2.2.2 and 4.2.3
UMU	José Santa Pedro J. Fernández	Contributor to 2.4 and 4.1.6
VALEO	Christian Müller-Hirschhorn	Contributor to 2.2.3
VEDECOM	Ahmed Soua Jérémie Leska	Contributor to 2.2.5 and 4.5
VICOM	Gorka Vélez	Contributor to 2.2.3
WINGS	Christos Ntogkas Kostas Trichias Nelly Giannopoulou Paraskevas Bourgos Polina Dimiropoulou	Contributor to 2.2.2, 3.3, 4.2.1, 4.2.2.2(3) and 4.2.3

Control sheet

Version history			
Version	Date	Modified by	Summary of changes
0.1	07/04/2020	S. Zougari (AKKA)	Initial proposed ToC
0.2	31/08/2020	S. Zougari, section editors	Detailed ToC and section outlines
0.3-0.9	22/09/2020	S. Zougari, B. Dalet & section editors	Integration of section 1,2,3,4,5,6
0.10	16/11/2020	S. Zougari, section editors	Last integration
0.11	16/11/2020	Alain Renault, Kostas Trichias	Fine-tuning, sanity check and editing corrections
0.11-15	21/01/2020	S. Zougari, section editors	Document following EC Periodic Report 1 recommendations
1.0	27/01/2020	Marie-Laure Watrinet	Final check, document ready for submission

Peer review		
	Reviewer name	Date
Reviewer 1	Janie Baños (DEKRA)	01/12/2020
Reviewer 2	Nuno Cruz and Nuno Datia (ISEL)	26/12/2020

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

EXECUTIVE SUMMARY	15
1. INTRODUCTION	16
1.1. 5G-MOBIX concept and approach	16
1.2. Purpose of the deliverable	16
1.3. Intended audience.....	17
2. COMMON METHODOLOGY	18
2.1. Common data logging requirements.....	20
2.2. Time Synchronization approach for OBU, RSU and Application.....	21
2.3. Common data format	26
2.4. Data Quality Check Tools.....	27
3. 5G-MOBIX EVALUATION DATA	32
3.1. Run time conditions	32
3.2. Communication Level data	33
3.3. CCAM Application data	36
3.4. Additional information	39
4. DISTRIBUTED DATA MANAGEMENT	42
4.1. Spain-Portugal (ES-PT) Cross-Border specificities	42
4.2. Greece – Turkey (GR-TR) Cross-Border specificities	59
4.3. German (DE) Trial Site specificities	73
4.4. Finnish (FI) Trial Site specificities	80
4.5. French (FR) Trial Site specificities.....	86
4.6. Dutch (NL) Trial Site specificities	92
4.7. Chinese (CN) Trial Site specificities	97
4.8. Korean (KR) Trial Site specificities.....	100
5. CENTRALISED TEST SERVER DATA MANAGEMENT	103
5.1. CTS data management requirements.....	103
5.2. CTS functional description	104
5.3. CTS system deployment	106
5.4. CTS interfaces and access to data	107
5.5. CTS tools.....	112
5.6. Test data archive structure.....	116
5.7. GDPR Process/Implementation	117

6. CONCLUSION	118
REFERENCES.....	119
ANNEXES	120
ANNEX 1. TEST DATA DESCRIPTION FILE	120
ANNEX 2. CTS CONFIGURATION FILES	121

List of figures

Figure 1: Generic scheme of data collection in 5G-MOBIX.....	19
Figure 2: Data flow and the different stages before uploading to the CTS.....	27
Figure 3: Data Builder and DQCT cooperation schema.....	29
Figure 4: Generic TACS ₄ Performance System Overview.....	45
Figure 5: IQ-NPE system architecture.....	46
Figure 6: IQ-OBU functional architecture.....	47
Figure 7: QoS FSU functional architecture.....	48
Figure 8: IQ-MS architecture.....	48
Figure 9: Example of IQ MS application interface.....	49
Figure 10: PCOs in an example of Advanced Driving.....	52
Figure 11: PCOs in an example of Extended Sensors.....	53
Figure 12: PCOs in an example of Remote Driving.....	54
Figure 13: PCOs in an example of Vehicle QoS Support.....	55
Figure 14: Data Management in ES-PT CBC.....	56
Figure 15: Folder structure in ES-PT local server.....	56
Figure 16: Nemo RF traces example.....	62
Figure 17: CAMINO architecture.....	63
Figure 18: Remote driving user story deployment scenario.....	65
Figure 19: Log collection and management process.....	68
Figure 20: Collected log.....	69
Figure 21: Assisted “zero-touch” Border-crossing architecture.....	70
Figure 22: Measurement data collection set-up at GR-TR CBC.....	71
Figure 23: Prometheus server architecture for data collection.....	76
Figure 24: Grafana panel showing End2End bitrate metric collected with Prometheus as data source.....	77
Figure 25: General architecture of the data management in the DE TS.....	78
Figure 26: Netradar app.....	82

Figure 27: Results on frame drop (left) and latency (right) from measurements of KPIs using custom tool for LEVIS video streaming application.....	82
Figure 28: PCOs in FI TS extended sensors with redundant edge processing user story.....	84
Figure 29: PCOs in FI TS remote driving in redundant network environment user story	84
Figure 30: FR TS 5G modem manager tool	88
Figure 31: FR TS data evaluation client architecture	89
Figure 32: FR TS data server evaluation architecture	89
Figure 33: General architecture of the data server in the FR TS.....	91
Figure 34: NL TS website to upload log data. (Top, middle and bottom sections of the web page)	95
Figure 35: NLTS data management and server architecture	95
Figure 36: CN data flow and server architecture	99
Figure 37: Centralised Test Server and Builder application use cases	105
Figure 38: Centralised Test Server functional description	105
Figure 39: Centralised Test Server system architecture	107
Figure 40: CTS login	108
Figure 41: CTS test data list	109
Figure 42: CTS test data details	109
Figure 43: Swagger REST interface.....	111
Figure 44: CTS authentication procedure	112
Figure 45: CTS authorization procedure	112
Figure 46: Test data builder secured login	113
Figure 47: Test data builder main form	114
Figure 48: Test data builder adding test data file.....	115
Figure 49: Test data archive content	116

List of tables

Table 1: Types of measurements	18
Table 2: Common data logging requirements	20
Table 3: Summary of latency requirements	22
Table 4: List of Quality Check Requirements	28
Table 5: Explanation of each field in the formatting CSV file	29
Table 6: Run time conditions of mobile devices, UEs and communication units in vehicles	32
Table 7: 5G infrastructure Level 0 data measurements	33
Table 8: 5G infrastructure Level 1 data measurements	34
Table 9: 5G infrastructure Level 2 data measurements	35
Table 10: CCAM application observations	37
Table 11: Contextual information of a test run	39
Table 12: Safety Intervention Form	40
Table 13: ES-PT data logging requirements	42
Table 14: Measurement tools at the three levels of communication for Advanced Driving	52
Table 15: Measurement tools at the three levels of communication for Extended Sensors	53
Table 16: Measurement tools at the three levels of communication for Remote Driving	54
Table 17: Measurement tools at the three levels of communication for Vehicle QoS Support	55
Table 18: GR-TR data logging requirements	59
Table 19: Data collection and management responsibilities in the GR-TR CBC	72
Table 20: DE data logging requirements	73
Table 21: DE data collection and management responsibilities	78
Table 22: FI data logging requirements	80
Table 23: FR data logging requirements	86
Table 24: FR data collection and management responsibilities	91
Table 25: NL data logging requirements	92
Table 26: NL TS data collection	96

Table 27: CN data logging requirements.....	97
Table 28: CN data collection responsibilities	100
Table 29: Summary of latency requirements	100
Table 30: KR data collection responsibilities	102
Table 31: CTS non-functional requirements.....	103
Table 32: Test data description file content.....	120

List of listings

Listing 1: Example of one formatting csv file	30
Listing 2: Output of the quality check process in JSON format	31
Listing 3: Properties file of Test Data Builder	121
Listing 4: Configuration file of test data download script.....	122

ABBREVIATIONS

Abbreviation	Definition
AD	Autonomous/Automated Driving
AV	Automated Vehicle
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CBC	Cross Border Corridor
CCAM	Cooperative, Connected and Automated Mobility
CDN	Content Delivery Network
CLI	Command Line Interface
CN	Core Network
CNOM	Core Network Operation Manager
CPM	Cooperative Perception Message
CTS	Centralised Test Server
C-ITS	Cellular Intelligent Transport System
C-V2X	Cellular Vehicle to Everything
DENM	Decentralised Environmental Notification Message
DMP	Data Management Plan
DPIA	Data Protection Impact Assessment
DPO	Data protection Officer
DQCT	Data Quality Check Tool
E2E	End to End
EC	European Commission
ECU	Engine Control Unit
EDM	Edged Dynamic Map
eNB	eNodeB (Evolved Node B)

EPG	Evolved Packet Gateway. When it use with -C: Control plain, -U: User plain
eRSU	Extended Road Side Unit
GA	General Assembly
GDPR	General Data Protection Regulation
gNB	gNodeB (Next Generation NodeB)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HD	High Definition
HMI	Human Machine Interface
HTTPS	HyperText Transfer Protocol Secure
IMEI	International Mobile Equipment Identity
IPv4 / IPv6	Internet Protocol v4 / v6
IQ-FSU/MU/NPE	ISEL QoS – Fixed Side Unit / Management System / Network Performance Evaluation
ITS	Intelligent Transport System
JSON	Java Script Object Notation
KPI	Key Performance Indicator
LAT	Latitude
LON	Longitude
LRQ	Logging Requirement
LTE	Long-Term Evolution
MAC	Medium Access Control
MCM	Manoeuvre Cooperation Message
MEC	Multi-access/Mobile Edge Computing
MME	Mobility Management Entity
mmWave	Millimetre Wave
MOS	Mean Opinion Score

MQTT	Message Queuing Telemetry Transport
NAS	Non-Access Stratum
NG-RAN	Next Generation-Radio Access Network
NTP	Network Time Protocol
OBU	On-Board-Unit
ORDP	Open Research Data Pilot
PCAP	Packet Capture
PCFICH	Physical Control Format Indicator Channel
PCO	Point of Control and Observation
PDCP	Packet Data Convergence Protocol
PDCCH / PDSCH	Physical Downlink Control Channel / Physical Downlink Shared Channel
PLMN	Public Land Mobile Network
POPD	Protection Of Personal Data
PTP	Point To Point
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RLC	Radio Link Control
ROC	Remote Operations Centre
ROS	Robot Operating System
RPM	Revolutions Per Minute
RRC	Radio Resource Control
RSU	Road Side Unit
SAE	Society of Automotive Engineers
SFTP	Secured File Transfer Protocol

SINR	Signal to Interference and Noise Ratio
TCP	Transmission Control Protocol
ToC	Table of Content
TS	Trial Site
TSTS	Trial Site Test Server
UDP	User Datagram Protocol
UE	User Equipment
UCC	Use Case Category
US	User Story
USB	Universal Serial Bus
UUID	Universally Unique Identifier
VRU	Vulnerable Road Users
V2X	Vehicle to Everything
WLAN	Wireless Local Area Network
WP	Work Package
X-border	Cross-border
XML	eXtensible Markup Language
XSD	XML Schema Definition
5G NR	5G New Radio

EXECUTIVE SUMMARY

This deliverable presents 5G-MOBIX methodology and tools for collecting and managing the data needed by WP5 – Evaluation, to assess both network capabilities in an UCC/US-agnostic manner and user perceived performance in the context of the specific CCAM use cases/applications considered in 5G-MOBIX.

The deliverable defines a bottom-up and top-down approach enabling the definition of a common methodology. This methodology, used at trial site level and at centralised level, ensures the collection of consistent data across the different Cross Border Corridors (CBC) and Trial Sites (TS). Thereby, the data to be collected for the evaluation of communication performance has been identified and a common data logging format agreed, the data quality check process defined, the measurement tools selected, the time synchronisation approach taken into account and the partners involved in the data collection identified.

5G-MOBIX defines for each CBC/TS one distributed data management environment where the data are collected, processed and integrated.

5G-MOBIX proposes a unique platform called a Centralised Test Server (CTS) to upload, store and share collected data, allowing the evaluators to work with harmonized data. The CTS unifies the way the test data will be transferred and guarantees that all mandatory metadata will be provided. The resulting CTS platform is used to get the best value out of the test data collected during 5G-MOBIX test executions and eases the identification of research data that will be shared under Open Research Data Pilot (ORDP [1]).

The work presented here has been elaborated within Task 3.5 of 5G-MOBIX, in cooperation with WP5.

The document is organised in 5 main sections. Section 1, Introduction, presents 5G-MOBIX, the purpose of the document and its intended audience. Section 2, Common methodology, explains the methodology applied and describes the approach used for time synchronisation at each TS level. Section 3, 5G-MOBIX evaluation data, describes the datasets to be collected and used for KPI computation. Section 4, Distributed data management, describes at trial site level the data logging requirements, the measurement tools, the data management tools and the local organisation in terms of partners involvement in data collection and in GDPR process implementation. Sections 5, Centralised test server data management, describes the requirements, the architecture and the tools developed within the CTS to collect, store and share the data. Section 6 presents the conclusions.

1. INTRODUCTION

1.1. 5G-MOBIX concept and approach

5G-MOBIX aims to showcase the added value of 5G technology for advanced Cooperative, Connected and Automated Mobility (CCAM) use cases and validate the viability of the technology to bring automated driving to the next level of vehicle automation (SAE L₄ and above). To do this, 5G-MOBIX will demonstrate the potential of different 5G features on real European roads and highways, and create and use sustainable business models to develop 5G corridors. 5G-MOBIX will upgrade existing key assets (network infrastructure, road and cloud infrastructure and vehicles) to realise improved performance of involved systems. 5G-MOBIX will then utilize and operate co-existence of 5G technologies within a heterogeneous environment comprised of multiple incumbent short-range communication technologies such as ITS-G5 and LTE-V2X and existing LTE networks.

5G-MOBIX will execute CCAM trials along cross-border (x-border) and urban corridors using 5G core technological innovations to assess the 5G infrastructure and evaluate its benefits in the CCAM context. The Project has also defined deployment scenarios and identifies and responds to standardisation and spectrum gaps.

5G-MOBIX has defined critical scenarios needing advanced connectivity provided by 5G, and the required features to enable some advanced CCAM use cases. The matching of these advanced CCAM use cases and the expected benefits of 5G will be tested during trials on 5G corridors in different EU countries, as well as in Turkey, China and Korea.

The trials will also allow 5G-MOBIX to conduct evaluations and impact assessments and to define business impacts and cost/benefit analysis. As a result of these evaluations, and of international consultations with the public and industry stakeholders, 5G-MOBIX will identify new business opportunities for the 5G enabled CCAM and propose recommendations and options for its deployment.

Through its findings on technical requirements and operational conditions, 5G-MOBIX is expected to actively contribute to standardisation and spectrum allocation activities.

1.2. Purpose of the deliverable

The purposes of this deliverable are to:

- Define the requirements necessary for the collection, storage and provisioning of data for the evaluation.
- Define a common methodology to be implemented by each CBC and TS in order to collect data with good quality, which is mandatory for KPI computation, and consistent across the different corridors and trial sites.

- Describe the tools and the processes that will be used to perform the data collection and management.
- Describe the data that will be collected to enable KPIs computation according the WP5 requirements (mandatory for ORDP).
- Describe how the data quality check will be performed and how data will be managed at CBC/TS level and at Centralised level.

1.3. Intended audience

This document is public and therefore, the intended audience is considered to be anyone that is interested in data collection tools, data measurements tools and architecture, and data sharing solutions that could be implemented in large-scale data collection and management of test data.

Within the project, the main intended audience for this deliverable is considered to be all 5G-MOBIX participants involved in the implementation of the tools, the data management system at the CBC/TS and centralised level and those involved in the technical evaluation.

2. COMMON METHODOLOGY

The main objective of this deliverable is to define the methodology and to deliver the tools to be used in all 5G-MOBIX Cross Border Corridors and Trial Sites (CBC and TS). The data collection methodology has been created using both a bottom-up and a top-down approach. In the bottom-up approach, trial sites described their existing assets, tools and facilities, their data collection environments, and finally, their partners involvements and roles in this process. In the top-down approach, additional workshops and discussions with trial sites, data providers and evaluators were organised to explain evaluator's requirements and to refine the scope and the boundaries of the data collection methodology. The goal of these discussions was to find a common approach for data collection tools, measurement tools, time synchronization, data provisioning, data description, management tools and data sharing for all trial sites.

To ensure the comparability and consistency of collected data across trial sites, teleconferences were organized to define precisely the type of logging measurement to be collected. Two additional discussion threads were organized to complete the description of the data to be collected: the **"Logging Measurement Task Force"** and the **"Common Data Format Working Group"**.

The **"Logging Measurement Taskforce"** was jointly created by Task 3.5 and Task 5.2 under the control of Task 5.1 and worked in a Cross WP3/5 activity context with key partners from WP5 and WP3. This taskforce aimed to define the data categories to be collected and their attributes (type, range, unit, accuracy), and how the data should be collected and used for the computation of KPIs. In this frame, several data measurement templates (handed as Excel files) were shared with sites in order to capture and to assess their capabilities to provide the requested data. Hence, after consultations between trial sites, WP5 and Task 3.5, most of the data categories, except application data, were specified, and trial sites declared which data they will be able to provide. Due to several issues (5G chipset availability, capabilities to deploy measurement tools in 5G network segments) some trial sites were not able to confirm their ability to provide certain parameters. Table 1 summarizes the types of data logging that were discussed by the task force.

Table 1: Types of measurements

Type of measurement	Description
Run time conditions	Measurements associated to the movement of the UE
Level 0	Measurements associated to the communication access layer
Level 1	Measurements associated to the communication transport layer
Level 2	Measurements associated to the communication application layer

The **"Common Data Format Working Group"** was created by Task 3.5. The working group used the outcomes of **"Logging Measurement Task Force"** and validated with Task 5.2 the requirements to define

a common format for the log data that will be common between all trial sites (section 2.3). This format was used for the definition of the **common Data Quality Check Tool** described in section 2.4. This tool will be used by all trial sites to check that their data are following the agreed common format, and by the **Test Data Builder tool** described in Section 5.5.1, to control that only valid data can be uploaded to the CTS.

Finally, we strived to enforce the Open Research Data Pilot (ORDP [1]) initiative and apply the requirements defined in the data management plan. ORDP defines requirements related to sharing research data. These requirements are clarified in D1.6 – “Interim Data management Plan” and are used by Task 3.5 to guarantee that the collected data will be provided in compliance with the European Commission Guidelines on Data Management in Horizon 2020 [2]. Hence, a process of harmonization, quality checking, and data formatting will be done at the level of each CBC/TS, before transferring the data to the centralised server. A Data Manager at each CBC/TS shall be appointed to control this process and to perform all mandatory data quality check operations using a common data quality check tool developed by the project. Finally, thanks to this process, all the data stored at the central test server are good candidates for data sharing of research data according to the ORDP initiative. These data will be shared using Zenodo¹. A repository for 5G-MOBIX was created in Zenodo [3].

Figure 1 shows the data flow between the trialling activities, local data storage and processing facilities, the centralised test server and finally the data sharing

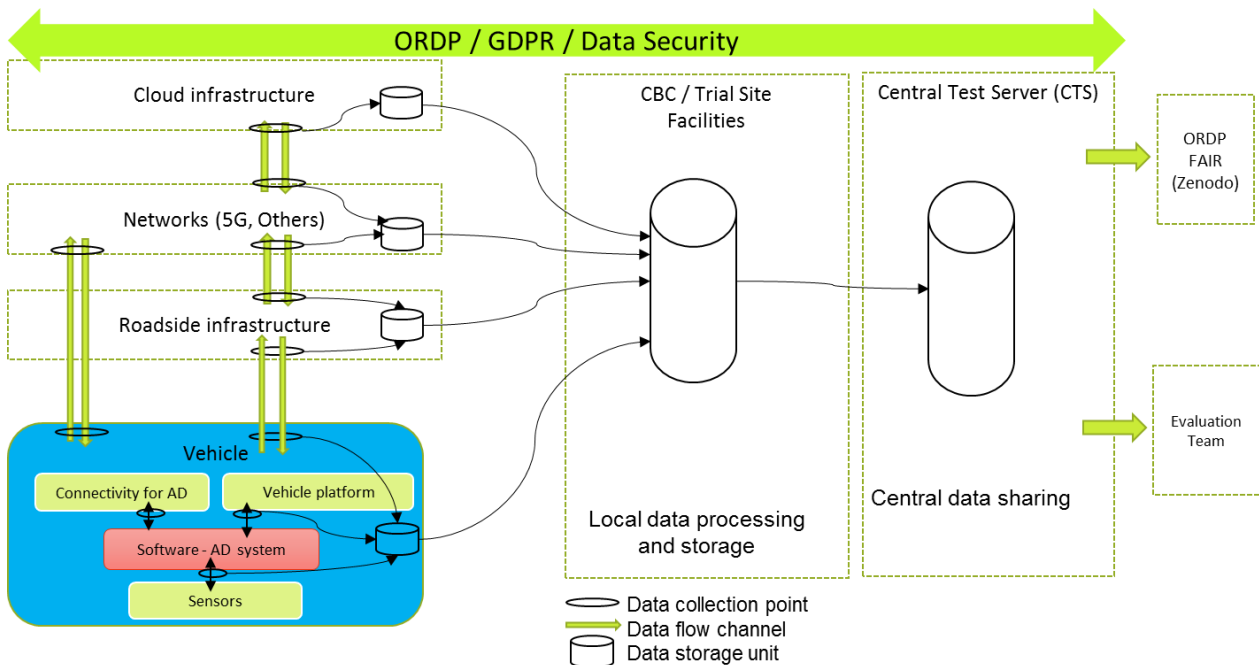


Figure 1: Generic scheme of data collection in 5G-MOBIX

¹ Zenodo is an open research repository for EC funded research created by OpenAIRE and CERN to provide a place for researchers to deposit publications, datasets and other research artefacts such as code, posters and presentations.

To secure the data collection and the data sharing process, each trial site was requested to have a concrete plan for the selection of the adequate tools, for the time synchronisation, for the provision of data storage and server tools, and for the implementation of the GDPR. All these aspects are described in Section 4 Distributed data management.

2.1. Common data logging requirements

This section summarizes the common logging requirements applicable to all trial sites that were used to build a common methodology.

Table 2: Common data logging requirements

ID	Description
LRQ-ALL-01	All systems and applications providing logging must be time synchronised.
LRQ-ALL-02	Each trial site must have the necessary tools to check data quality.
LRQ-ALL-03	All the provided data must be clearly described with metadata and timestamped.
LRQ-ALL-04	Each trial site must ensure the data collection according to the data quality requirements.
LRQ-ALL-05	Trial sites must support the application-agnostic measurement of network capabilities, as defined in Table 9 of D5.1 – “Evaluation methodology and plan”.
LRQ-ALL-06	Each trial site must support the logging of data required for the support of the application-specific measurements and the KPIs listed in D5.1 Appendix B.
LRQ-ALL-07	Trial sites must support data logging at Levels 0, 1 and 2, as defined in D5.1 (Section 3.2).
LRQ-ALL-08	Trial sites must support for synthetic traffic generation, for the evaluation of network capacity (application-agnostic).
LRQ-ALL-09	Data logging information must include at least the following information: timestamp, location, identity of the infrastructure element, identity of the PCO, Level specific information, Error Code and Error description (see D5.1).
LRQ-ALL-10	The source-destination pairs of logged/measurement traffic must be synchronised.
LRQ-ALL-11	A common <Error Code, Error Description> index must be defined to capture all data logging error conditions.
LRQ-ALL-12	All measurement data must be collected and sent to a local (per trial site) repository for quality check/verification.

LRQ-ALL-13	Quality check must be applied to the measurement data collected before the end of the trials, to allow for repetition of measurements if needed.
LRQ-ALL-14	Data logging (measurements) tools must be part of the verification and functional validation processes.
LRQ-ALL-15	Data logging must support the collection of measurement data before, during and after a handover/roaming event.
LRQ-ALL-16	Logged data (measurements) shall include a description of the experiment conditions i.e., vehicle, application, network.
LRQ-ALL-17	Logged data (measurements) must be retrievable on an experiment granularity (i.e., repetition)
LRQ-ALL-18	For application-specific measurements, data logging must be performed on a traffic flow granularity i.e., individual KPI measurements must correspond to data traffic corresponding to a single traffic each time, as described in D5.1.
LRQ-ALL-19	Logged data (measurements) must be retrievable on a traffic flow granularity i.e., selecting data logs for a specific traffic flow (as specified in D5.1).
LRQ-ALL-20	Evaluation Scenario metadata (see LRQ-ALL-16) should include information that (i) uniquely identifies the evaluation scenario description i.e., filled-in template, (ii) describes the current mobility conditions of a vehicle (where applicable) i.e., current location and speed at the time indicated by the data log entry timestamp.
LRQ-ALL-21	Data should be uploaded to the CTS in common log format from adopted within the project
LRQ-ALL-22	Data uploaded to the CTS should be grouped in archives (test data archive ²) containing additional file for evaluation scenario and data description, and a data quality check report file.
LRQ-ALL-23	Test data archive should contain when necessary, as described in section 3.4, a contextual information and a safety intervention file.

2.2. Time Synchronization approach for OBU, RSU and Application

Time synchronization is a key aspect for the establishment of an accurate measurement and evaluation framework, given the distributed nature of the multi-node, end-to-end CCAM environment. Namely,

² Test data archive content and format are defined section 5.6

measurements are in principle expected to take place across communicating network entities/nodes i.e., the RSUs, the vehicles, the OBUs and the servers used in the trial, evaluating performance aspects inherently related to time e.g., end-to-end latency, throughput, and mobility interruption time. The challenge is how to ensure that the clocks of all participating nodes/devices are synchronized at an acceptable accuracy level.

In order to identify the actual accuracy requirements, the project resorted to the most stringent end-to-end latency requirements across the UCC/US considered in the project: obviously the lowest the latency required and/or expected, the higher the impact of any clock shift; as such, the accuracy requirement was defined relatively to the latency requirements. Table 3 summarizes the end-to-end latency requirements as reported in D5.1, on a per UCC/US basis. Evidently, the low latencies expected, around a few milliseconds, imply a relatively high accuracy. In this context, and given the currently established practices, the project has decided to target a synchronization accuracy value in the 100 μ s-1ms range i.e., node system clocks should present synchronization looser than this value range.

Table 3: Summary of latency requirements

UCC	User scenario	Trial site	Leader	Latency requirements (ms) (D5.1)
UCC-1: Advanced Driving	Complex manoeuvres in cross-border settings	ES-PT	CTAG	200
	Infrastructure-assisted advanced driving	FR	VEDECOM	13
	Cooperative Collision Avoidance	NL	VTT	50
	Cloud-assisted advanced driving	CN	DUT(DALIAN)	20
	Automated shuttle driving across borders	ES-PT	CTAG	200
UCC-2: Vehicles Platooning	Platooning with "see what I see" functionality in cross-border settings	GR-TR	FORD	100
	eRSU-assisted platooning	DE	TUB	40
	Cloud assisted platooning	CN	DUT(DALIAN)	20
UCC-3: Extended Sensors	Complex manoeuvres in cross-border settings: HD maps	ES-PT	CTAG	200
	Complex manoeuvres in cross-border settings: Public Transport with HD media services and video surveillance	ES-PT	CTAG	200
	Extended sensors for assisted border crossing	GR-TR	WINGS	50
	EDM-enabled extended sensors with surround view generation	DE	VALEO	40

	Extended sensors with redundant Edge processing	FI	AALTO	100
	Extended sensors with CPM messages	NL	TNO	20
UCC-4 : Remote Driving	Automated shuttle remote driving across borders	ES-PT	CTAG	150
	Remote driving in a redundant network environment	FI	AALTO	80
	Remote driving using 5G positioning	NL	TUE	50
	Remote driving with data ownership focus	CN	DUT(DALIAN)	20
	Remote driving using mmWave communication	KR	KATECH	120
UCC-5: Vehicle Support	QoS Public transport with HD media services and video surveillance	ES-PT	CTAG	200
	Tethering via Vehicle mmWave communication	KR	ETRI	NA

In this context, the following sections report on the exact technical approach followed by each CBC/TS in order to achieve the desired timing accuracy. This includes information regarding the synchronization protocol employed e.g., NTP, PTP, the time references e.g., GNSS timings at specific nodes, and the overall setup of the clock synchronization solution.

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

The 5G network Cores will use GNSS time references. The Core will have a NTP server to provide network clients accurate synchronization that may be used by all equipment. The NTP server will be provided by Network Operators, Telefonica (ES) and NOS (PT). The NTP server will provide a stratum 1 for MEC applications (time synchronization accuracy around 1ms).

Devices, such as On Board Unities (OBU) and Road Side Unities (RSUs) will use direct GNSS time references to assure more accuracy and the internal clock for redundancy. The Network's NTP server can even be used for redundancy. However, it is expected that the time synchronization accuracy will decrease considerably.

The Road Side Unities from Instituto de Telecomunicações (IT) will use a GNSS receiver to synchronize with the GNSS time reference, synchronizing with the Traffic Radars through a PTP.

ITS Centres and Vulnerable Road Users (VRU) applications and applications that will run in the MEC will not have GNSS. They will use network's NTP server.

The Road Side Units from CTAG will use the GNSS time reference to synchronize the Traffic and Pedestrians radars.

On the Radio network, synchronization requirements are far more demanding (PTP and below 1ms). The Radio Network of NOS (operator on the PT side) will be synchronized with GNSS and PTP T-GM G.8275.1.

2.2.2. Greece – Turkey (GR-TR) Cross-Border Corridor

The IMEC OBUs and RSUs that are being deployed in the GR-TR Cross-Border Corridor are synchronised via GNSS using the built-in USB GPS device (Navilock NL-8012U) which is used as a source for the internal NTP daemon. The system clock will be set via the NTP daemon allowing an accuracy of the system clock of around 1-2ms. The provided accuracy is sufficient for the foreseen latency measurements as indicated in Table 3.

The WINGS OBU/RSU and app platform are using UTC Clock-epoch time/GNSS time. Synchronization to GNSS provided timing is also available via GNSS receiver and/or synch to GNSS server via URL. For the purpose of synchronization across all components used in the GR-TR trials, the WINGS OBU/RSU and platform have been upgraded to support synchronization with an external NTP server. The same NTP server that is used for the synchronisation of the Ericsson 5G network components will be utilized.

The Truck Routing Application is using NTP server on the cloud. This NTP server is receiving time from a cloud firewall which fetches time from 3 Internet time servers.

TURKCELL-Radio Network nodes will use both, "phase and time synchronization" (IEEE 1588v2) and "GNSS synchronization" as a backup. To achieve this, all transport network is upgraded to enable IEEE 1588v2 and GPS antenna installation will be done. These methods will guarantee the required synchronization accuracy for 5G technology. ERICSSON-Turkcell's underlying NTP infrastructure is re-used to synchronize 5G-EPC nodes via NTP protocol using Stratum-1 access via GPS enabled solution.

The Radio Access Network (RAN) at the GR side uses GPS for Traffic Synchronization, while NTP is used for the synchronization at the network level with Packet Core and ENM. RAN/Packet Core/ENM will be synchronized from a common NTP server (installed for the purpose of the trial) synchronizing with Internet NTP Server.

2.2.3. German (DE) Trial Site

The default time synchronization method for the VALEO test vehicle's OBU is NTP. NTP typically provides precision on the order of milliseconds [4], which satisfies the requirements for the DE TS use cases. Software modules for GNSS-based OBU time synchronization are also readily available, such that the system is flexible and can quickly adapt to a potential change in requirements. For the eRSU, time synchronization is done using the GNSS signal received by the C-V2X unit, but also NTP synchronization can be used for redundancy.

2.2.4. Finnish (FI) Trial Site

Multiple synchronisation methods are used in the FI TS. The synchronisation between the vehicle and the Remote Operations Centre (ROC) in the remote driving user story is based on NTP. The NTP synchronization is also used between the LEVIS video streaming server and clients. In the case of the 5G network PTP synchronisation is utilised. While these platforms may use different synchronisation methods, the latency measurements are conducted between synchronised and distributed DEKRA software agents, installed in PCOs between which latency measurements carried out. The source-destination pair of agents are synchronised using an improved version of NTP.

2.2.5. French (FR) Trial Site

The 5G network will use GNSS time references and will have a NTP server to provide to network clients accurate synchronization. Vehicles and OBUs will use GNSS time references (through the use of GPS-RTK [5] (Real-time kinematic)) to ensure higher accuracy and may use the Network's NTP server for redundancy. Real-time kinematic positioning is a satellite navigation technique used to enhance the precision of position data derived from satellite-based positioning systems like GPS.

On the other hand, the MEC infrastructure will use NTP to ensure time synchronisation between the MEC applications, VMs and containers.

2.2.6. Dutch (NL) Trial Site

All servers and services in the fixed network are synchronised via NTP. All mobile and field devices are synchronised on GNSS using Chrony³. All logging units (PCO) from Table 26 provide aggregated logging on time synchronisation accuracy estimates in the 'clockstatistics' log format (version 0.7.11 will be released on [6])

2.2.7. Chinese (CN) Trial Site

The OBU of the vehicle and RSU synchronize time through the NTP protocol. The 5G system uses GNSS for time service and uses NTP to ensure time synchronization with other equipment. Specifically, Chrony is used to synchronise the system clock with the NTP server reference clock. In addition, MEC, Central Cloud, to name a few also use NTP servers to ensure time synchronization between applications and database servers.

³ <https://chrony.tuxfamily.org/>

2.2.8. Korean (KR) Trial Site

The mmWAVE based 5G network in the test vehicle and the remote server are synchronized through NTP servers. The test vehicle has GNSS systems so that GNSS time is also referred to when the test vehicle and OBUs create messages.

2.3. Common data format

A common format for the data has been specified. This ensures data are consistent across all Trial Sites, are standardized for CTS upload, are defined to facilitate evaluation at project level, and have a standard format when reported publicly. This also applies to the aggregated communication performance measures that are used for calculating the communication performance KPIs defined in deliverable D5.1.

The common format complies with the following general specifications:

- All log items are timestamped in absolute UNIX timestamp in milliseconds since the Unix Epoch, midnight 1st January 1970 UTC, minus the leap seconds since 1970.
- Every measurement point, or Point of Control and Observation (PCO) is identified by:
 - stationId – this is the unique identifier of the physical UE, device, vehicle, server, or service
 - applicationId – this is an additional identifier that is only needed if a stationId has multiple PCOs.
- Locations are defined by the WGS84 coordinates with absolute geographical latitude and longitude in degrees in double value type with at least 0.1 micro degree precision.
- File format is a standard .csv file [7]:
 - Comma (,) separator – not as semi-colon (;)
 - Decimal separator is a point (.) – not a comma (,)
 - Header line with the exact parameter names in small letters – no capital letters nor CamelCase
 - Missing values are skipped and not replaced by some default character like a space
- Log file name convention is: **<log_item>_<stationid>_<utc_time_iso8601>.csv**
Where log_item is one of the following standard items and names:
 - accessaggr – for layer 0 access layer aggregated data
 - networkaggr – for layer 1 network and transport layer aggregated data
 - applicationaggr – for layer 2 application layer aggregated data
 - handoverevent – for hand over eventsAnd utc_time_iso8601 is the UTC start time of the log data in human readable time format of YYYYMMDD'T'HHmmss to enable manual search of data

Detailed specifications for each of the log_items are described in section 3 .

Note that raw communication logging, application and vehicle data logging will be collected in several formats in the trials. This logging may also be shared via the CTS.

2.4. Data Quality Check Tools

In order to ensure the required level of quality in the measured and logged data during the different tests, all the trial sites must use the same Data Quality Check Tool (DQCT) that is going to be provided. The University of Murcia has developed this tool using Java, generating a JAR file that can be used by other Java applications or directly via command line (CLI).

One important requirement of the DQCT is that the logging information has to be formatted in an agreed common data format, so testers have to generate their logs, or translate them, to this common data format. This can be easily seen in the Figure 2, where the different stages of the data flow are shown. In this common data formatting are also agreed the log files naming, the internal formatting of the files (CSV), the valid ranges of the stored values, and the presence of all mandatory fields.

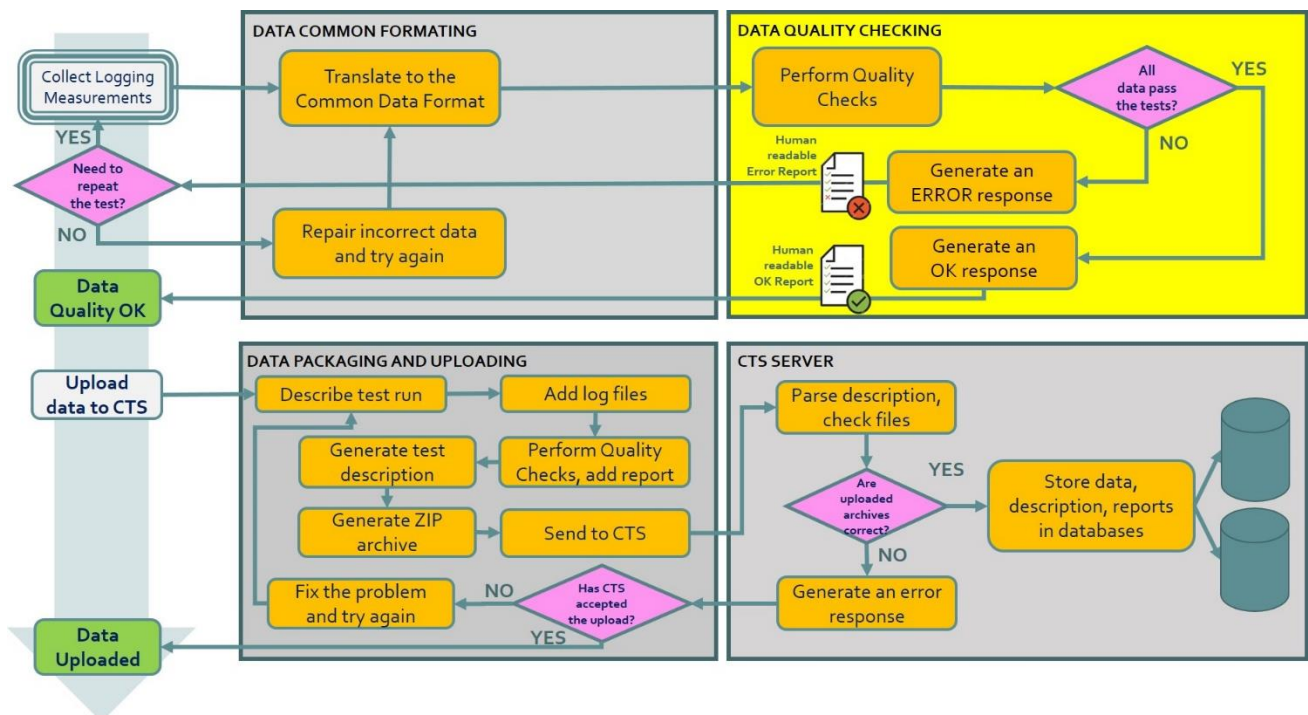


Figure 2: Data flow and the different stages before uploading to the CTS

On the one side, if any error is found in the logs, the DQCT will continue checking the rest of logs looking for a complete list of errors that will be returned to the tester. It is worthy to mention that the log files are not modified by this tool. Quality checking is a read-only task. To repair or re-generate new log files is responsibility of the tester or the data manager, based on the reported errors.

On the other side, if no errors are found, then the DQCT will return a PASSED notification to the tester, and the process can continue with the data packaging and uploading stage to save the data into the CTS.

From T5.2, a set of quality check requirements (QCR) was formulated and passed to the DQCT development team. These requirements define how the data are expected to be received by the evaluation team, establishing a certain level of quality that allows, for example, the calculation of the different KPIs in the evaluation phase. These requirements are listed in Table 4. All of them are successfully satisfied by the developed DQCT tool.

Table 4: List of Quality Check Requirements

ID	Description
QCR-COM-R1	The input data for the quality check tool must be previously formatted in a common data format .
QCR-COM-R2	The quality check tool must check if the file naming and the number of log files of each test-run are correct.
QCR-COM-R3	The quality check tool must provide a human readable output indicating possible errors in the data , that helps testers to correct them or decide to repeat the test if necessary.
QCR-COM-R4	The quality check tool must provide a proof file of the quality checking process in XML or JSON format. A formatting proposal is provided.
QCR-COM-R5	The quality check tool must verify conformance of data format (valid CSV, JSON, etc.).
QCR-COM-R6	The quality check tool must verify the existence of all mandatory (critical) measurements and assure that none of them is missing.
QCR-COM-R7	The quality check tool must verify the measurements are formatted in the correct type (integer, float, string, etc.), based in the information provided by the TF Logging Measurements.
QCR-COM-R8	The quality check tool must verify data measurements are in the range and valid values , based in the information provided by the TF Logging Measurements.
QCR-COM-R9	The quality check tool must verify if data have the minimum expected resolution of decimal data (e.g. 2 decimal places), based in the information provided by the TF Logging Measurements.
QCR-COM-R10	The quality check tool must verify the correctness of GNSS data , ensuring also that all the registered positions are inside the test site boundaries .
QCR-COM-R11	The quality check tool must verify the correctness of temporal data , ensuring that all measurements are between a certain starting and ending timestamps .
QCR-COM-R12	The quality check tool must not remove or alter any data present in the log files. This tool must only identify the errors and report them to TS testers.

For the sake of flexibility and reutilization purposes, the DQCT has been implemented in the form of a java library that could be used as an isolated application, or included as a functionality for other related applications. For example, the Test Data Builder application developed by AKKA for packaging and uploading logging measurements to the Central Test Server (CTS), uses the DQCT library to check the data quality (see Figure 3).

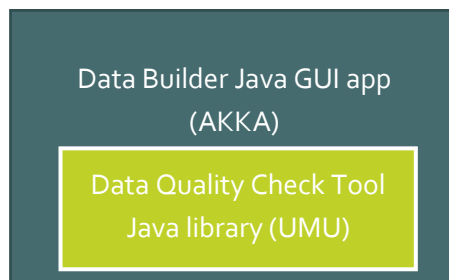


Figure 3: Data Builder and DQCT cooperation schema

The first goal of the DQCT implementation was to be flexible enough to support different input log file formats. Supposing that all the input files comply with the agreed common data format, the DQCT let the different partners provide CSV format files to define their logging formats. The description CSV files, define the logging format, including the name of each field, its type, whether it is mandatory or not, range of valid values, etc. The meaning of each field is defined in Table 5.

Table 5: Explanation of each field in the formatting CSV file

Field name	Explanation and possible values
fieldname	This is the name of the field that must be located in the first line of the CSV file
mandatory	"M" for mandatory; "O" for optional If you leave it in blank, it will be considered as optional
type	The type of the value. The supported types are: int, long, float, double, string, enum, boolean
min_value, max_value	Specifies the minimum and maximum value to qualify the measured value is in a valid range. If min_value and max_value are both zero, then all values are valid with no restrictions. If type is string, this is the minimum and maximum valid length for the string. If min_value and max_value are both zero, then there is no restriction for its length.
enum_valid_values	For the type enum, this value indicates an enumeration of valid and expected values. You must use "" and separate the valid values with the vertical bar Example: "SEND RECV"
semantic	This value indicates the meaning of the field, the semantic that the quality

	<p>check tool needs to perform more advanced checking. Indicates the meaning and the units. They can be:</p> <ul style="list-style-type: none"> timestamp_s, timestamp_ms, latitude_ndeg, longitude_edeg, altitude_m, speed_mps, speed_kmph <p>These semantic tags are not mandatory, but the fact of putting them indicates the tester is requesting the advanced checking tests, like looking for impossible values of speed, positions out of test boundaries, etc.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The name of the formatting CSV file will be the string that will match the beginning of the name of the corresponding log CSV files to be checked. In this way, a formatting file named "CAM.csv" will be applied in all logging files with names in the format "CAM*.csv" (where * can be any text).

For the sake of clarity, an example of formatting file that can be taken as a template is shown in Listing 1:

```
fieldname,mandatory,type,min_value,max_value,enum_valid_values,semantic
log_timestamp,M,long,0,0,,timestamp_ms
log_stationid,M,int,0,0,,
log_applicationid,M,int,0,0,,
log_action,M,enum,0,0,"SENT|RECEIVED",
log_communicationprofile,M,enum,0,0,"ITS_G5",
stationid,M,int,0,0,,
generationdeltatime,M,int,0,0,,
generationtimestamptai,M,long,0,0,,
generationtimestamputc,M,long,0,0,,
referencepositionlatitude,M,float,-90,90,,latitude_ndeg
referencepositionlongitude,M,double,-180,180,,longitude_edeg
referencepositionalaltitudevalue,M,int,-100,9000,,
referencepositionalaltitudeconfidence,M,int,0,65535,,
headingvalue,M,int,0,360,,
headingconfidence,M,int,0,65535,,
speedvalue,M,int,0,200,,speed_mps
speedconfidence,M,int,0,65535,,
drivedirection,M,int,0,360,,
longitudinalaccelerationvalue,M,int,-255,255,,
longitudinalaccelerationconfidence,M,int,0,65535,,
yawratevalue,M,int,-1000,1000,,
yawrateconfidence,M,int,0,65535,,
```

Listing 1: Example of one formatting csv file

The DQCT can generate a proof file in JSON format that summarises the checking process, adding a MD5_sum or equivalent for each file in order to detect cases of file manipulation or corruption. One example is shown in Listing 2:

```
{
  "test_id": "Test_01"
  "quality_check_datetime": "Sat Oct 24 01:01:14 CEST 2020"
  "successes": [
    {
      "file": "GPS_20KMPH.csv",
      "line": "0",
      "test": "Logging file has passed the quality check tests?",
      "result": "SUCCESS",
      "message": "All quality checks performed in this logging file have been
                  passed successfully. MD5_sum: [087acf7dae64f735083d772740e4225a]"
    }
    {
      "file": "GPS_40KMPH.csv",
      "line": "0",
      "test": "Logging file has passed the quality check tests?",
      "result": "SUCCESS",
      "message": "All quality checks performed in this logging file have been
                  passed successfully. MD5_sum: [67a1ae80763c6619b47e15b3c6c0d7e5]"
    }
  ],
  "warnings": [
  ],
  "errors": [
    {
      "file": "CAM_3902_20190829T101605.csv",
      "line": "6",
      "test": "Is calculated speed feasible taking into account the current and
              last position?",
      "result": "ERROR",
      "message": "Calculated speed [27467.10275516487] is over 120 Km/h. May be
                  it is too much."
    }
  ]
}
```

Listing 2: Output of the quality check process in JSON format

3. 5G-MOBIX EVALUATION DATA

This section presents the datasets that need to be collected for technical evaluation. Log data are divided in the following datasets that are specified in the sub sections:

- **Run time conditions** log the position and motion of vehicles and UEs.
- **Communication level data** include all log data needed for evaluation of the communication performance.
- **CCAM Application data** include all log data collected from the connected, cooperative and automated applications in vehicles, UEs, road-side or cloud services. These data are needed to evaluate the impact of communication performance on the application performance and automated driving functions.
- **Additional information** may be needed to differentiate the CCAM application performance for different environmental situations and conditions. These do not include the variations in communication network setup and communication technologies as these are already covered in the communication level data.

Trial Sites will collect their log data and also calculate their own technical KPIs. Nevertheless, the quality and contents of the data to be collected, how data should be analysed and evaluated should be common. The following sections define the contents of the data that needs to be collected by all Trial Sites in a similar manner.

For aggregated communication logging, a common format is agreed that all Trial Sites must provide and share via the CTS, to enable data quality validation and cross Trial Site evaluations.

3.1. Run time conditions

Table 6 identifies the run time conditions, i.e., the position and motion of vehicles and UEs during the execution of a test, it is also indicated if these measurements are mandatory for the evaluation or optional. These data allow geolocation of communication or application performance measurements.

Table 6: Run time conditions of mobile devices, UEs and communication units in vehicles

Name	Description	Mandatory / Optional
Timestamp	Timestamp in UTC	M
Node ID	Unique ID of the node (end point of the communication) that is logging	M
UE speed	Speed of the UE	M
UE acceleration	Acceleration of the UE m/s ²	O

UE longitude	Longitude provided by the GNSS in WGS84	M
UE latitude	Latitude provided by the GNSS in WGS84	M
UE 5G longitude	Longitude provided by the 5G network	O
UE 5G latitude	Latitude provided by the 5G network	O
Lane	Lane in which the UE is driving	O

3.2. Communication Level data

Data collected for evaluation of communication performance is carried at different levels in the communication stack and devices. Table 7-Table 9 list the parameters that are expected for evaluation of specific analysis. It is also indicated if they are mandatory, optional or conditional meaning that the CBC/TS could choose to log the C1 either the C2 measurements.

Table 7 lists the measurements needed from the radio access layer of communication (Level 0) of devices and network nodes, identifies which networks and media are accessed, and the measurements needed to evaluate radio access performance.

Table 7: 5G infrastructure Level 0 data measurements

Name	Description	Mandatory / Optional
Timestamp	Timestamp at which the node logs the data row	M
Node ID	Unique ID of the node (end point of the communication) that is logging	M
Action	Action that triggered the login event	M
NAS procedures	Procedures of Non-Access Stratum	M
RRC procedures	Procedures of Radio Resource Control protocol	M
Protocol	Communication protocol at the Access Layer	M
RAT mode	Radio Access Technology	M
Cellular MNC	Mobile Network Code	M
Cellular MCC	Mobile Country Code	M
Cellular RAT	Radio Access Technology	M
LTE ARFCN	LTE Absolute radio-frequency channel number	O
LTE PCI	LTE Physical Cell Identity	M

LTE Cell ID	LTE Cell ID	O
LTE eNBID	eNB identifier	M
LTE TAC	LTE Tracking Area Code	O
LTE RSSI	LTE Received Signal Strength Indicator	O
LTE RSRP	Reference Signal Received Power	O
LTE RSRQ	Reference Signal Received Quality	O
NR PCI	NR Physical Cell Identity	M
NR ARFCN	NR Absolute radio-frequency channel number	O
NR SS-RSRP	NR Reference Signal Received Power	O
NR SS-RSRQ	NR Reference Signal Received Quality	O
NR Availability	Indicates if there is 5G coverage in current location	O
Cellular CQI	Channel Quality Indicator	O
LTE TA	LTE Timing Advance	O
Error ID	Identifier of the error code during the measurement	O

Table 8 identifies the measurements needed at the transport layer of communication (Level 1) from the devices and network nodes, and identify the packets delivered and transmission performance of the different networks and media.

Table 8: 5G infrastructure Level 1 data measurements

Name	Description	Mandatory / Optional / Conditional
Timestamp	Timestamp at which the node logs the data row	M
Node ID	Unique ID of the node (end point of the communication) that is logging the message	M
Destination Node ID	Unique ID of the node (end point of the communication) that is receiving the message	O
Action	Action that triggered the login event	M
Direction	Communication direction of the data flow, i.e. uplink or downlink	M
Protocol	Communication protocol at the Transport layer	M

Packet content	Content of the packet	C1
Packet size	Length of the packet	C1
DL Throughput	Amount of successful information downloaded in bits per second	C2
LTE DL Throughput	Amount of successful information downloaded through LTE channels	C2
NR DL Throughput	Amount of successful information downloaded through NR channels	C2
UL Throughput	Amount of successful information uploaded in bits per second	C2
LTE UL Throughput	Amount of successful information uploaded through LTE channels	C2
NR UL Throughput	Amount of successful information uploaded through NR channels	C2
One-way delay	Time required for a packet to travel through two end points of the network	C2
Jitter	Variation of the one-way delay between packets	O
Packet Loss rate	Percentage of packet loss in the unit time	O
Error ID	Identifier of the error code during the measurement	O

Table 9 identifies the measurements needed from the application layer of communication (Level 2) of devices and network nodes. Measurements are needed to characterise the data flows between sending and receiving applications, the communication performance in terms of message size and rate. Optionally, it has also information on the message generation time and impact on device resources is collected.

Table 9: 5G infrastructure Level 2 data measurements

Name	Description	Mandatory / Optional / Conditional
Timestamp	Timestamp at which the node logs the data row	M
Node ID	Unique ID of the node (end point of the communication) that is logging	M
Action	Action that triggered the login event	M
Protocol	Communication protocol at the Application layer	M
Flow type	Type of message to be transferred	O

Payload	Data content	C1
Message ID	Identifier of the message	C2
Message size	Length of the message in size	C2
Error ID	Identifier of the error code during the measurement	O
Battery level	Battery level	O
Battery Consumption	Battery consumption of the app during testing	O
CPU usage	CPU usage when Message generation time (percentage)	O
RAM usage	RAM when Message generation time (percentage)	O
App in background or in foreground	Checks if app is running in foreground or in background	O
Message generation start	Timestamp of the moment in which message generation started (UTC time)	O
Message generation end	Timestamp of the moment in which message generation ended (UTC time)	O
Message Rate	Value in Hz of the application message rate. Defined as T_CheckCamGen on CAM standard)	O
Message status	Description of Error related with Message Generation.	O

3.3. CCAM Application data

The project objective is to identify whether and to what extent the communication performance affects the user stories that make use of CCAM. The high-level research question for the user stories is *if and how communication performance affects the automated driving functions and services*. The baseline scenario is the scenario defined by the existing communication network, e.g. a 4G LTE network given functionality of the vehicles and UEs and behaviour under a baseline communication test case. The level of automation and the performance of the vehicles functions themselves is not subject of evaluation in this project.

Direct effects of communication performance issues are the delayed reception of messages (i.e., latency), corrupted or entirely missed messages (packets lost), e.g. during a hand-over event at x-border. Effects on the user story can be classified as impact on continuous control processes or impact on event-based control processes.

Continuous control processes, such as steering and speed control of automated or remote driving functions, are typically safety related and are robust against external sources, such as communication related issues. Hence, the driving functions will adapt and compensate for the communication performance issues during the execution of a test (test run). The adaptation of the continuous control commands may be difficult to detect, for example as delayed manoeuvres that are followed by corrections. Effects on continuous vehicle control can be detected from vehicle log data, including:

- automated control and response parameters, such as speed (throttle and brake), steering, absolute positioning and trajectories or relative position on the road.
- on-board sensors information and detections of other road users, obstacles and lane markings.

Adaptations in vehicle control may also be discrete events, such as the decision to fall back to another control mode, handover of control to the test driver, or an intervention by the test driver.

Communication may also affect the internal decision logic of driving functions, such as the timing and magnitude of actions, control decisions, events and transitions in state machine models.

All these measurements are based on the CCAM application level observations reported by the CBC/TS (Table 10).

Table 10: CCAM application observations

UCC	User scenario	Trial Site	Application level observation
UCC-1: Advanced Driving	Complex manoeuvres in cross-border settings	ES-PT	Degree of safety (number of accelerations/decelerations) when performing the lane merge / overtaking manoeuvre
	Infrastructure-assisted advanced driving	FR	Delay or gaps in detecting the collision risk between automated vehicle and basic vehicle Delay or missed planned trajectories guidance from the infrastructure to the AV via MCM messages
	Cooperative Collision Avoidance	NL	Delay or missed collision detections and planned trajectories for evasive actions via MCM message
	Automated shuttle driving across borders	ES-PT	Degree of safety (number of accelerations/decelerations) when performing the braking manoeuvre from the Remote Control Centre
UCC-2:	Platooning with "see what I see"	GR-TR	Platoon manoeuvres, continuous video

Vehicles Platooning	functionality in cross-border settings		streaming and autonomous path following in customs area
	eRSU-assisted platooning	DE	Messages arrive too late or do not arrive
UCC-3: Extended Sensors	Complex manoeuvres in cross-border settings: HD maps	ES-PT	Degree of safety (number of accelerations/decelerations) when the rear vehicle drives through the road works with the information provided by the front vehicle
	Complex manoeuvres in cross-border settings: Public Transport with HD media services and video surveillance	ES-PT	Degree of safety (number of accelerations/decelerations) when the rear vehicle drives through the road works with the information provided by the front vehicle
	Extended sensors for assisted border crossing	GR-TR	Detection of contraband, smuggling attempts, cross-checking of license plates, increased safety of customs agents, degree of successful threat assessment, remote stopping capability
	EDM-enabled extended sensors with surround view generation	DE	Delay and outages from video stream; speed of streaming playback
	Extended sensors with redundant Edge processing	FI	Delay, gaps or outages in vehicle status messages, sensor data and video stream from vehicle to ROC (UL) Delay, gaps or outages in vehicle command messages from ROC to vehicle (DL)
	Extended sensors with CPM messages	NL	Delays and gaps in neighbouring vehicle manoeuvres (CPM), errors in the estimations of gaps and commands for speed and lane changes
UCC-4: Remote Driving	Automated shuttle remote driving across borders	ES-PT	Degree of safety when the shuttle is driven remotely from the Control Centre
	Remote driving in a redundant network environment	FI	Delay, gaps or outages in video stream from vehicle to edge server (UL) Delay, gaps or outages in HD maps from edge server to vehicles (DL)
	Remote driving using 5G positioning	NL	Delays and gaps in video streaming (UL) and remote commands (DL) Delays in vehicle manoeuvres and corrective manoeuvre commands

UCC-5:		Public transport with HD media services and video surveillance	ES-PT	Quality of the video received in the Control Centre from the camera on the bus
Vehicle	QoS			
Support				

Trial sites and use cases may collect logging on CCAM applications in various ways and formats. The AUTOPILOT [8] project developed log formats that can be reused here and extended for the 5G-MOBIX user stories. More specifically, the vehicle data log format and application log formats from AUTOPILOT is reused in the NL Trial Site.

3.4. Additional information

Results of a test execution may be affected by conditions or unexpected events not considered in the test case description. For example, weather conditions. To be able to analyse the impact of those conditions on the results, additional information may need to be collected. These additional information has been classified as "contextual information" (e.g., weather) and "safety intervention" (e.g., type of intervention).

3.4.1. Contextual information

The execution of tests shall also record contextual information on, for example, test vehicle configuration, traffic situation, test users, weather and safety. Table 11 gives a list of contextual and situational information that can be included in the test case descriptions and test plan and may be needed to distinguish situations in the evaluations.

There is no need to specify that the condition is 'normal' if no specific traffic, road surface, or weather condition is intended and planned in a test case, unless an abnormal condition is expected in some other test cases.

When the context changes during the execution of a test, in such a way that it may affect evaluation results, a context report should be uploaded to the CTS describing the changes to the test case and any anomalies. If a change is planned and also used for more than one execution of a test case, then it makes sense to develop a specific new test case.

Table 11: Contextual information of a test run

Field or suggested field name	Description	Values
Number of vehicles	Specify the number of vehicles that participated in the test. If the type or number of vehicles deviates from the test case, then describe this in the Test Comment.	Integer (not zero)
Number and	Specify the number of UEs that participated in the test. If	Integer

type of UEs	the type or number of UEs deviates from the test case, then describe this in the Test Comment.	
Number of users	Specify the actual number of users that participated in the test. If the number or type of users in the test deviates from the test plan, then further information should be provided on a separate form and added to the uploaded data archive.	Integer
Traffic	Traffic situation is an enumeration to distinguish abnormal traffic conditions.	normal, congested, priority lane open, priority at traffic lights
Road	Road surface condition is an enumeration to distinguish abnormal road surface conditions that may impact AD.	normal, slippery, precipitation, snow, ice
Weather	Weather conditions is an enumeration to distinguish abnormal weather situations that may impact AD.	normal, fog, heavy rain, snow, ice
Safety Limitations	Free text description to indicate any limitations or restrictions imposed on automated driving, test drivers or users. For example, is automated driving not authorised, or specially trained test drivers allowed. Give a brief summary if this is already described in the test plan, so the impact on AD can be considered in the evaluations.	Text
Number of safety Interventions	Upload a safety intervention report for each safety intervention.	Integer

3.4.2. Safety intervention

All incidents with automated driving in real traffic conditions that require unplanned human intervention (i.e. disengagement from automated driving) shall be reported. This is only relevant for tests involving vehicles that are automated or remotely controlled. For manual driven vehicles, safety interventions do not need to be reported, unless the safety interventions are clearly related to a communication issue and relevant for evaluation. Table 12 gives the list of fields needed to describe a safety intervention.

Table 12: Safety Intervention Form

Field	Description	Values
Timestamp	Date and time of the safety intervention. It shall have the same format as the logged data timestamp. This timestamp is used to identify approximatively the time of the intervention in the logs.	Integer
Type	Type of intervention is determined by who or what	Test driver, User as Driver,

	intervened during automated driving.	Passenger, Other road user, Service Operator, Bystander.
Cause	Assumed cause of the unsafe situation that required the interventions.	Weather condition, inattentive road user, unwanted vehicle manoeuvre, perception discrepancy, HW discrepancy, SW discrepancy, road works, emergency vehicle, road surface condition, obstacle on the road.
Intervention description	A description of what happened.	Text
Severity perception	<p>Assess the severity of the exposed danger and the required intervention.</p> <ul style="list-style-type: none"> • Dangerous: an accident could have happened if the subject would not have intervened. (E.g. a system's failure that can cause an accident and only the reaction of the driver may avoid it). • Moderate: an accident could have happened, but the intervention to avoid it was trivial or automatic. (E.g. a system's failure that can cause an accident, but there are means that act automatically or driver is warned with enough time). • No risk: even if the subject would not have intervened, there was no risk. (e.g. the driver has intervened due to a system fault, but if he had not intervened, an accident was unlikely to happen). 	Dangerous, Moderate, No risk.
AD vehicle situation	Describe the activated AD functions and systems, their modes/states, and observed behaviour or malfunctioning.	Text
Traffic situation	Describe or sketch the traffic situation, traffic control and road users.	Text

4. DISTRIBUTED DATA MANAGEMENT

4.1. Spain-Portugal (ES-PT) Cross-Border specificities

4.1.1. ES-PT Data logging requirements

This section summarizes the specific requirements applicable to data logging at the ES-PT CBC (see Table 13).

Table 13: ES-PT data logging requirements

ID	Title	Description
LRQ-ES-PT-01	Time synchronization	A common NTP server with a GNSS correction must provide time synchronization among PCOs.
LRQ-ES-PT-02	Data collection	The data must be logged at each PCO, or at the segment between them.
LRQ-ES-PT-03	Local storage	The data must be stored at the end of the trial's day in the local format on the local server. Every partner is responsible for dumping the data of its own devices.
LRQ-ES-PT-04	Common data format	The local logging must be translated into the common data format with a local tool. A partner is assigned to this activity.
LRQ-ES-PT-05	Local quality check	A local quality check must be performed in the common format in order to detect failures in the PCOs or the communications among them. A partner is assigned to this activity.
LRQ-ES-PT-06	Upload of data to CTS	The data must be uploaded to the CTS in the common format and after passing the local quality check. A partner is assigned to this activity.
LRQ-ES-PT-07	Extended evaluation: data collection	In the Extended Evaluations, the local site that is performing the contribution must ask the CBC for the measurements needed for the Evaluation for the contribution.
LRQ-ES-PT-08	Extended evaluation: data management	In the Extended Evaluations, local sites are in charge of the complete data management process (except the logging of the measurements that asked the CBC).

4.1.2. ES-PT Measurement tools

ES-PT CBC is using different measurement tools for the UCC/US agnostic test cases and the UCC/US specific ones, for completeness and complementarity, and because of the different sources of the data flow.

4.1.2.1. Tools for UCC/US Agnostic Test Cases

The characterization of the ES-PT network capabilities is performed by means of different tools, allowing to cover a wide range of parameters in the definition of the synthetic traffic, and also increasing the measurement capabilities. These measurements are taken from both OBU and 5G smartphones against MEC, Core and/or ITS Centre, allowing to compare results. The description of tools used is given in the next subsections.

1) DEKRA TACS₄ Performance Tool

The TACS₄ Performance System delivers concurrent performance testing and user experience analytics of wired or wireless access networks for voice and data services. The TACS₄ Performance System is comprised of two solutions:

- TACS₄ Performance Test Tool, which is the subject of this section.
- TACS₄ Performance Testing Platform.

The TACS₄ Performance System is suitable for measuring:





- Cellular networks and devices.
- WLAN networks and devices.
- Fixed access networks performance.
- Embedded Car Unit telematics performance.
- In the laboratory or production networks in outdoor and indoor scenarios.

The TACS₄ Performance System target test scenarios are:

- Research & Development.
- Integration Verification.
- System Performance.
- Benchmarking.
- Carrier Acceptance.
- Customer Experience.

Architecture

The TACS₄ Performance System is a software system composed of four elements:

	TACS4-Controller	<p>Computer software to be installed in a Windows PC. It provides the GUI for running the tests with multiple TACS4-Agents. TACS4-Controller controls multiple TACS4-Agents, collects the measurements from them, computes the KPIs and presents the results to the User in different formats (graphical, tabular and map). Additionally, it can upload the results to the TACS4-Web, which provides extended Data Analytics features.</p> <p>The TACS4-Controller is commanded by the User (graphical user interface).</p>
	TACS4-Agent	<p>Computer multi-user and multi-platform software to be installed in the data endpoints or nodes (laptops, smart phones, tablets, etc.) which participate in the test scenario.</p> <p>It exchanges data traffic with other TACS4-Agents for QoS measurements (e.g., Throughput), perform QoE measurements (e.g., YouTube MOS), automate external hardware (e.g., Programmable Attenuator), and read some relevant parameters from the hosts where they run (e.g., GPS, WLAN RSSI, LTE RSRP).</p> <p>The TACS4-Agents are commanded by the TACS4-Controller.</p>
	TACS4-Mobile (part of TACS4 Performance Testing Platform)	<p>Mobile application (a.k.a. App) to be installed on devices that participates in the test scenario.</p> <p>It exchanges data traffic with other TACS4-Agents for QoS measurements, performs QoE measurements, and reads some relevant parameters from the host where it runs.</p> <p>The TACS4-Mobile can be commanded either by the User (graphical user interface) or by the TACS4-Web Application (unattended mode).</p>
	TACS4-Web (part of TACS4 Performance Testing Platform)	<p>HTML5 application for tests results collection, extended data analytics (e.g., KPI aggregation from test data from different days, devices, etc.) and TACS4-Mobile remote management. It is hosted in a DEKRA secured data centre.</p> <p>Both TACS4-Controller and TACS4-Mobile can upload results to the TACS4-Web Application.</p> <p>It is operated by the User from a web browser with Internet access.</p>

The TACS₄ Performance System solution overall architecture is depicted in Figure 4.

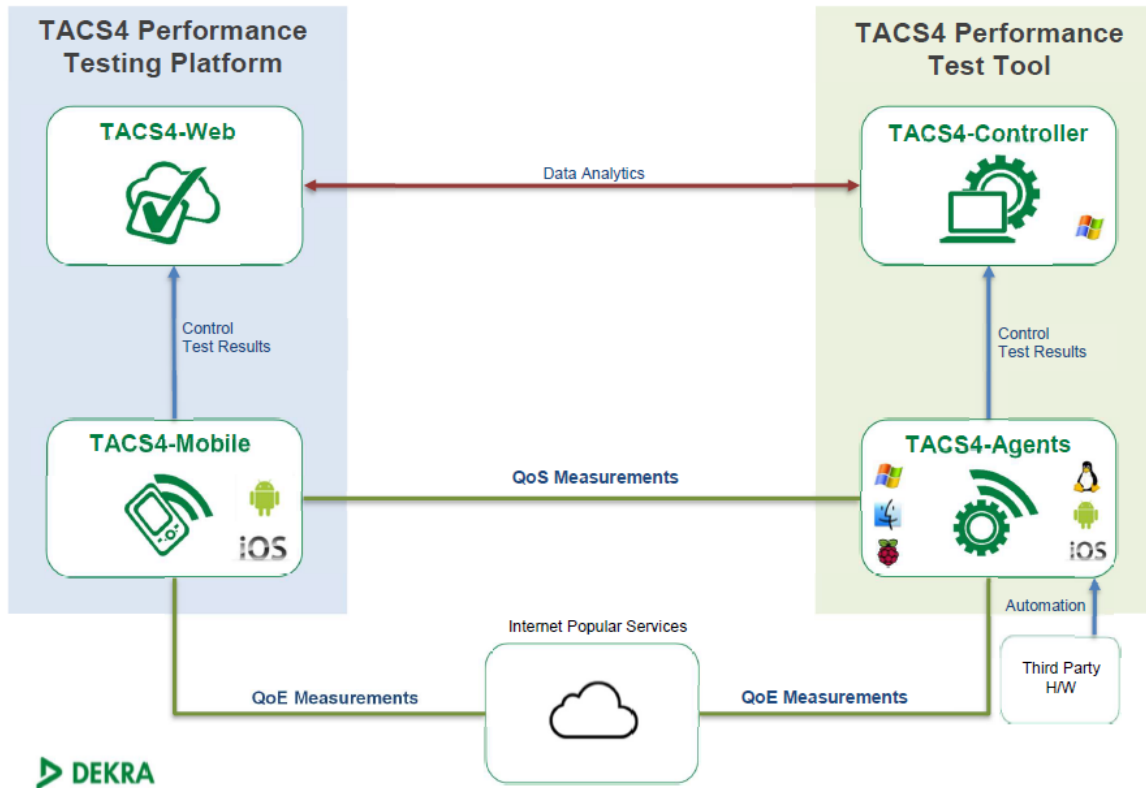


Figure 4: Generic TACS₄ Performance System Overview

TACS₄ agents have client and server functionality and can create data streams to measure the throughput between the two ends in one or both directions. TACS₄ will be used to perform QoS and QoE tests on the different configurations. TACS₄ can also be used to generate CCAM-like synthetic traffic and configure controlled loaded scenarios.

2) ISEL platform

The ISEL QoSNetwork Performance Evaluation (IQ-NPE) system (depicted in Figure 5) is based on a set of QoS probes (On-Board Units and Fixed Side Units) and a centralised management platform, and will be used to perform QoS tests over a 5G network, according to different UCC/US as well as agnostic scenarios. Moreover, synthetic/background traffic will be used to test the network under stressful traffic loads. The IQ-NPE system will collect and process performance parameters obtained from the probes and compute the key performance indicators (KPIs) identified in [deliverable D5.1](#). The system architecture is composed by the following components:

- IQ- OBU (ISEL QoS-On Board Unit), which is a hardware and software probe to be installed on vehicles in order to generate CCAM traffic and collect performance measurements at different PCOs. During the tests, four IQ-OBUs will be available to be used in the ES-PT CBC, thus allowing an increase in the traffic load level;

- IQ-FSU (ISEL QoS Fixed Side Unit), which is a software agent to be installed on both PT and ES MECs as well as at the PT and ES ITS centres. The IQ-FSU will be used to generate traffic and collect performance measurements on the network side, on both downlink and uplink traffic flows;
- IQ-MS (ISEL QoSManagement System), which is a centralised software platform used to manage the probes, including probe configuration and test plans. It will also be responsible for collection of all performance assessment results obtained during test trials.

The system uses out-of-band control connections, based on legacy 3G/4G cellular networks, to manage the QoS Probes installed on vehicles, in order to avoid traffic interference on the 5G access network interface. Depending on the test type, traffic flows can be generated/received by OBUs, as well as FSUs.

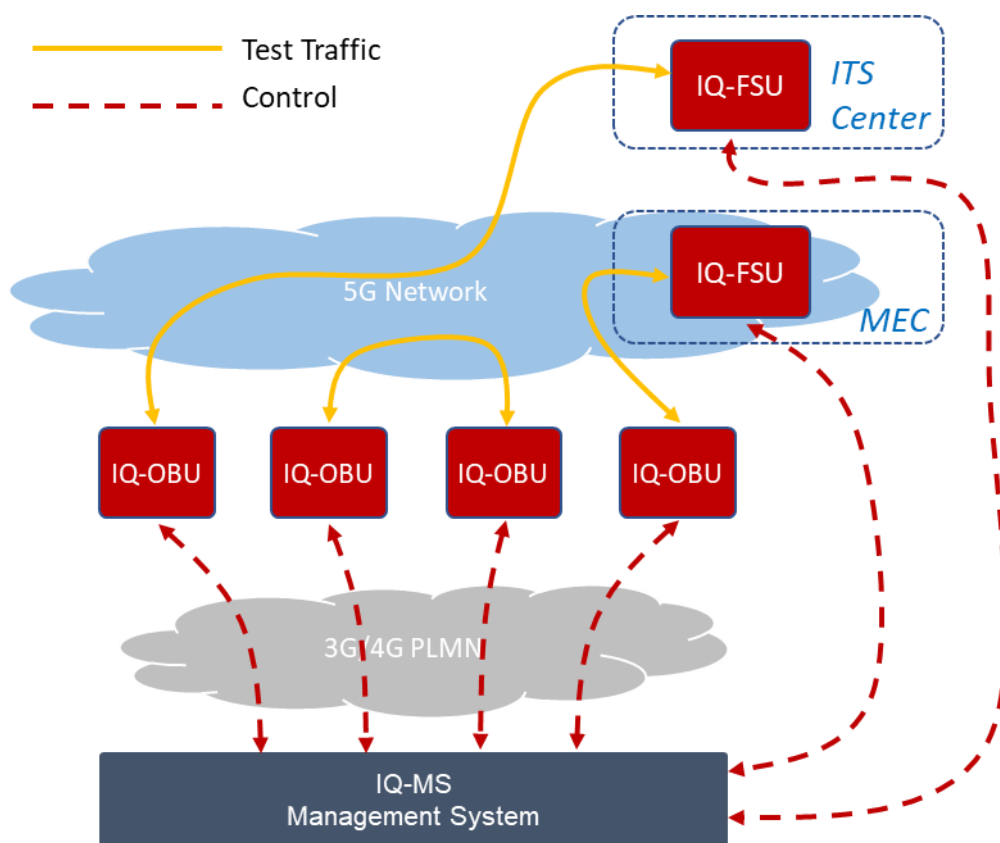


Figure 5: IQ-NPE system architecture.

The IQ-OBUs are physically based on automotive industrial PCs, prepared to host several mobile modems, on which the tests will be carried out. In addition to the 5G modem, the system has a 3G / 4G modem to establish control connections, as well as a GNSS receiver for location and data synchronization. Another important aspect that should be highlighted, is the availability of an interface with the vehicle's CAN BUS, which can be used to collect vehicle information, allowing for a greater accuracy in the analysis over the generated data. Figure 6 shows the IQ-OBU functional architecture. It should be noted that this

equipment also provides a local web interface, allowing the probe to be administered directly, without the intervention of the management system.

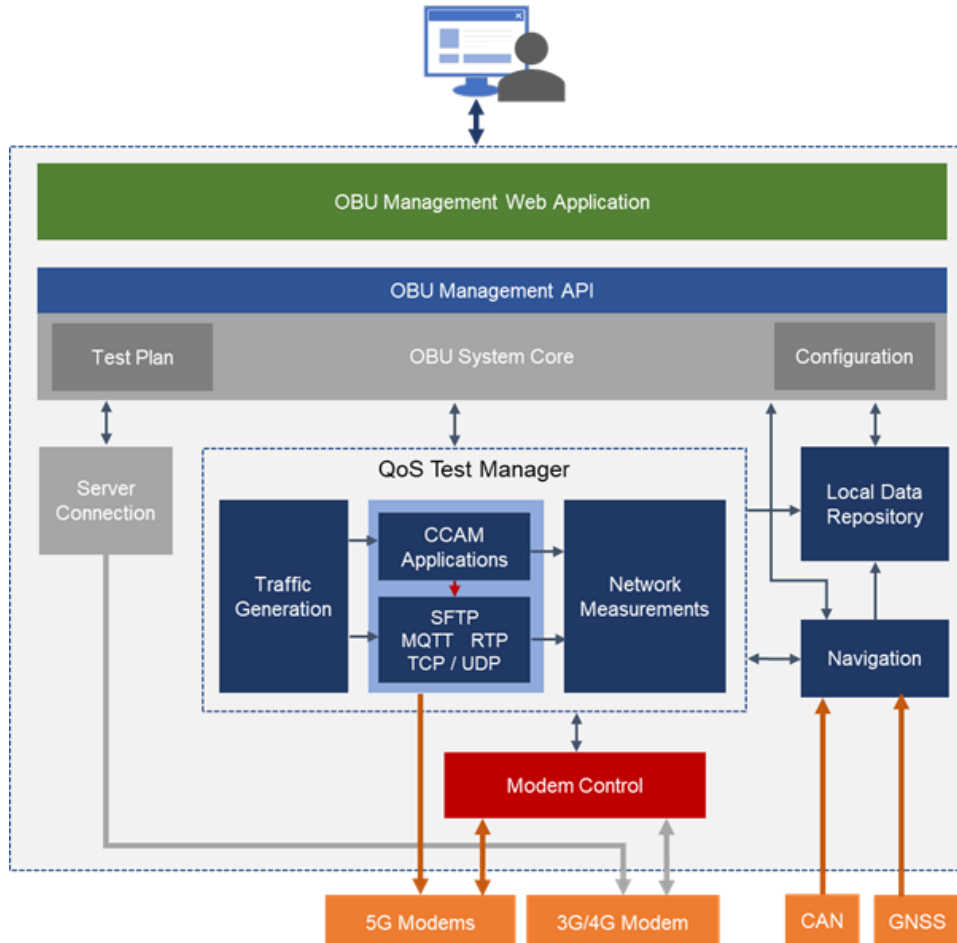


Figure 6: IQ-OBU functional architecture.

The IQ-FSU (running on a dedicated virtual machine) is intended to be installed in the 5G network MEC and/or in the ITS Centre. This allows a direct connection to the 5G network core, enabling an end-to-end traffic transmission measurement between OBUs and other elements of the ITS system. The IQ- FSU functional architecture, presented in Figure 7, is very similar to QoS OBU, with the exception that it does not have modems for cellular network communication and navigation interfaces (GNSS or CAN BUS).

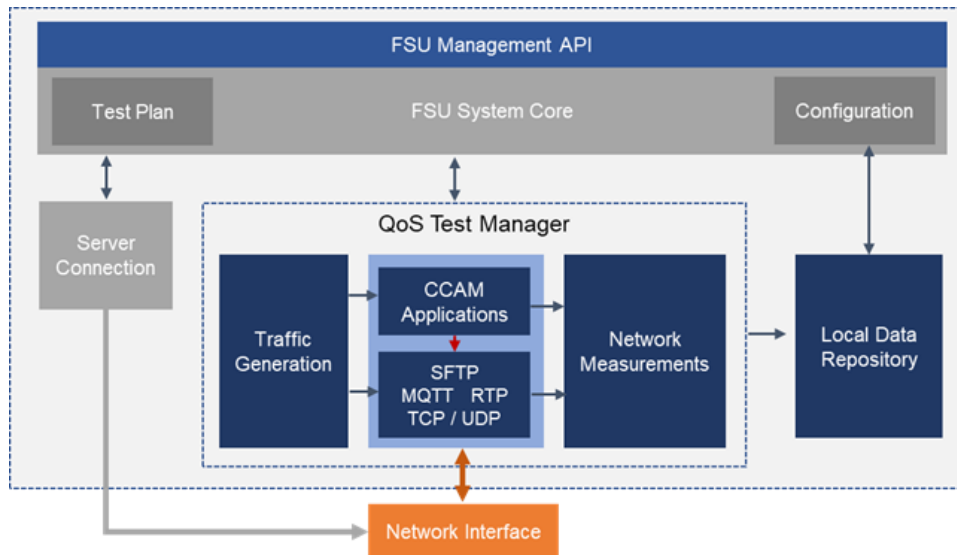


Figure 7: QoS FSU functional architecture.

The IQ-MS, as shown in Figure 8, is a central software platform, composed of several components, allowing for probes administration, test plans management, measurements collection, and their subsequent processing and KPI generation. Two APIs will be available. One that provides a set of methods that allows the system to be fully managed through a web application. A second API that assures the probes communication for sending test plans, system configuration, and collecting test measurements. The IQ-MS will also process other sources of information, merging collected data and allowing performance indicators extraction. An example of the user interface provided by the system administration web application can be seen on Figure 9.

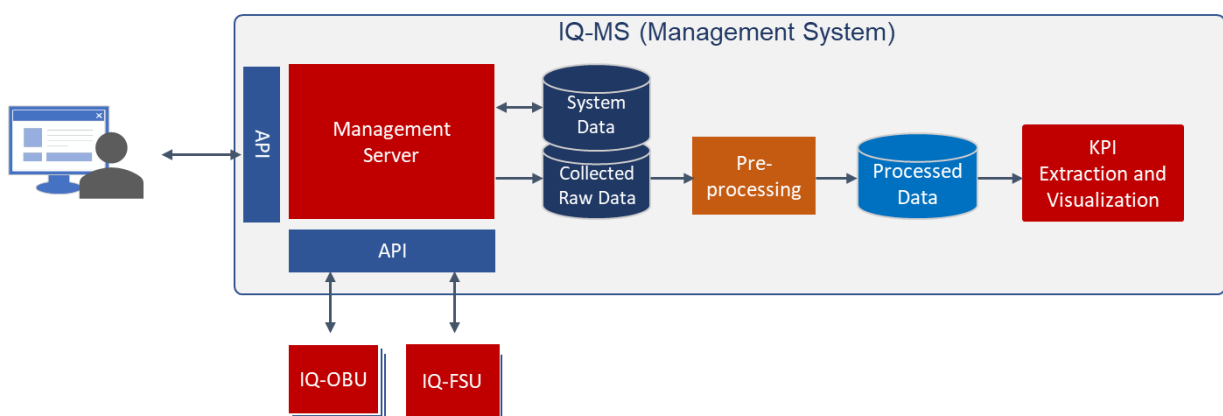


Figure 8: IQ-MS architecture.

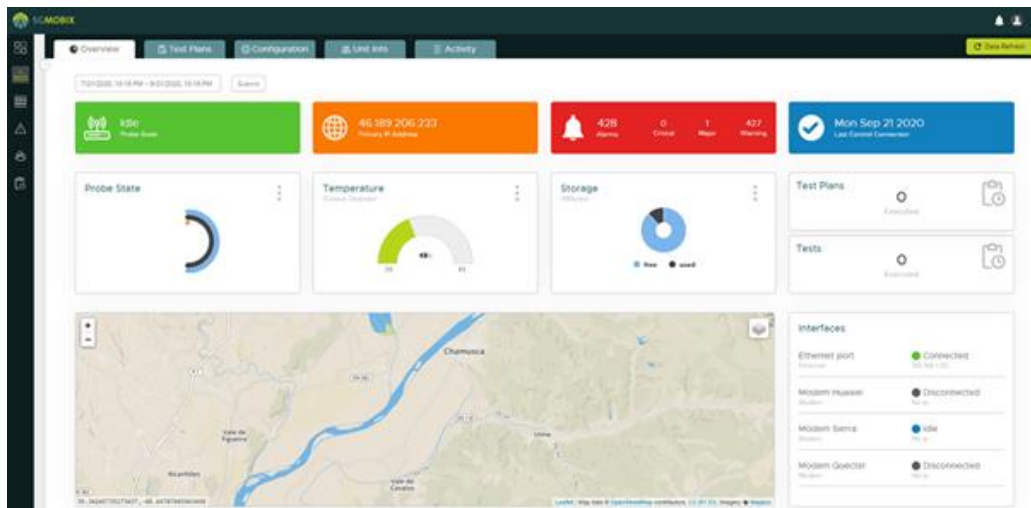


Figure 9: Example of IQ MS application interface.

3) Keysight Nemo Handy and Keysight Nemo Outdoor

Keysight Nemo Handy is a tool implemented by Keysight Technologies and which is installed on Android-based devices. The main functions of this software are to provide real-time measurement visualization and log of all RF and signalling data of a mobile network, including throughput, using iPerf3 with TCP or UDP protocols, and latency, moreover measurements and reliability statistics, all of them through adapted configurable scripts to allow the data collect to obtain QoS (Quality of Service) and QoE (Quality of Experience).

The logs are made up of several files, one with all the signalling and radio frequency data in Nemo's proprietary format, GPS coordinates and a csv file that includes a summary report of the tests carried out and some network indicators. These records must be post-processed with Nemo post-processing tools, like Keysight Nemo Outdoor or other post-processing tools.

Keysight Nemo Outdoor is a designed desktop application by Keysight Technologies. The tool provides a vast range of testing options and the generation of 5G NR KPIs for SA and NSA networks.

Nemo Outdoor helps to analyse the logs files in Nemo file format saved by Nemo Handy. The measured relevant KPIs for this project are exported to csv format to be adapted to the defined Common Data Format of this project.

KPIs to be measured in this project like User Experienced data rate (KPI 1.1), Throughput (KPI 1.2), reliability (KPI 1.6), NG-RAN Handover Success Rate (KPI 2.1), Application-level handover success rate (KPI 2.2) and Mobility interruption time (KPI 2.3) can be acquired by means of this tool.

In addition to register a wide volume of 5G KPIs, this software allows to trace all RF signalling from UE to eNB/gNB, hence, is used to parse huge amount of data that is collected by L3 Data Collector (Radio

Resource Control (RRC) protocol) for LTE and/or NR network. As it is well-known, RRC is a layer between UE and eNB/gNB and exists at the IP level (Layer 3).

RRC messages are transported via the PDCP protocol. These messages provide relevant information about air interface communication. Through Layer 3 messages it collects data about connection establishment and release functions, broadcast of system information, radio bearer establishment, reconfiguration and release, RRC connection mobility procedures, paging notification and release and outer loop power control procedures. Furthermore, the user and control planes are configured by means of signalling functions of RRC according to network status and allows for Radio Resource Management strategies to be implemented. In this project, the RRC messages play a fundamental role in some KPI measurements, for instance, in the control plane latency (KPI 1.4).

4) iPerf3

iPerf3 is a free open source and cross-platform command-line tool, widely used for network performance measurement and tuning in IP networks (supports IPv4 and IPv6). This is a simplified code base version of iPerf2. Another feature of iPerf3 is a single threaded tool and admits a library version which enables other programs to use the provided functionality (e.g. Nemo Handy).

Tests are totally configurable thanks to the presence of various adjustable parameters. iPerf3 has client and server functionality and can create data streams to measure the throughput between the two ends in one or both directions. Typical output contains a time-stamped report of the amount of data transferred and the throughput measured which can be saved in a detailed JSON output.

The data streams can be either User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). For the former, iPerf3 allows the user to specify the datagram size and provides results for the datagram throughput and the packet loss, and for the latter, the tool measures the throughput of the payload.

5) Other tools

In addition to the previous tools, the analysis of 5G KPIs can be performed using Nokia's proprietary tools like Nokia BTS Manager, which acquires the KPIs reports required for the project under the authorization of Mobile Network Operator (MNO) every five minutes, and other applications hosted in the MEC. The End-to-End latency (KPI 1.3) and User plane Latency (KPI 1.5) will be measured with these instruments.

4.1.2.2. Tools for UCC/US Specific Test Cases

Each one of the four out of five UCC/US developed in ES-PT CBC demands specific management of the measurement tools because of the different PCOs in the endpoints of the communications segments, and the particularities of the data flows. Most of these tools were developed ad-hoc for each particular PCOs and they are proprietary tools. Following the description of the commercial UCC/US specific tools:

1) Access layer

G-NetTrack Pro

G-NetTrack Pro is a network monitor and drive test tool application for 5G/4G/3G/2G networks. This tool is available for smartphones with Android OS. It allows monitoring and logging of mobile network serving and neighbour cells information, without using specialized equipment. It exports logs in CSV format. This is a commercial application and is not specific for 5G communication. The app is installed and configured in the Termux⁴ terminal, an android terminal emulator and Linux environment app.

This tool will be used to log metrics related with network in the VRU app, installed in an Android smartphone. The VRU app will be used in the UCC#1 US 5.

2) Transport layer

HTTP Canary

HttpCanary is a HTTP/HTTPS/HTTP2/WebSocket/TCP/UDP packets capture and analyser app designed for Android platform. It exports packages captured in several formats, including CSV.

It uses a proxy to allow the reading of packages without requiring the rooting of the phone. Since phones used to test the VRU app will not be routed, this is an essential feature. This is a commercial application. The app is installed and configured in the Termux terminal.

This tool will be used to log metrics related with network in the VRU app, installed in an Android smartphone. The VRU app will be used in the UCC#1 US 5.

TCPdump

TCPdump is a data-network packet analyser computer program that runs under a command line interface. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

Figure 10 - Figure 13 details the PCOs within the UCC/US and Table 14 - Table 17 the tools employed to collect their data at the three levels of communication.

⁴ <https://termux.com>

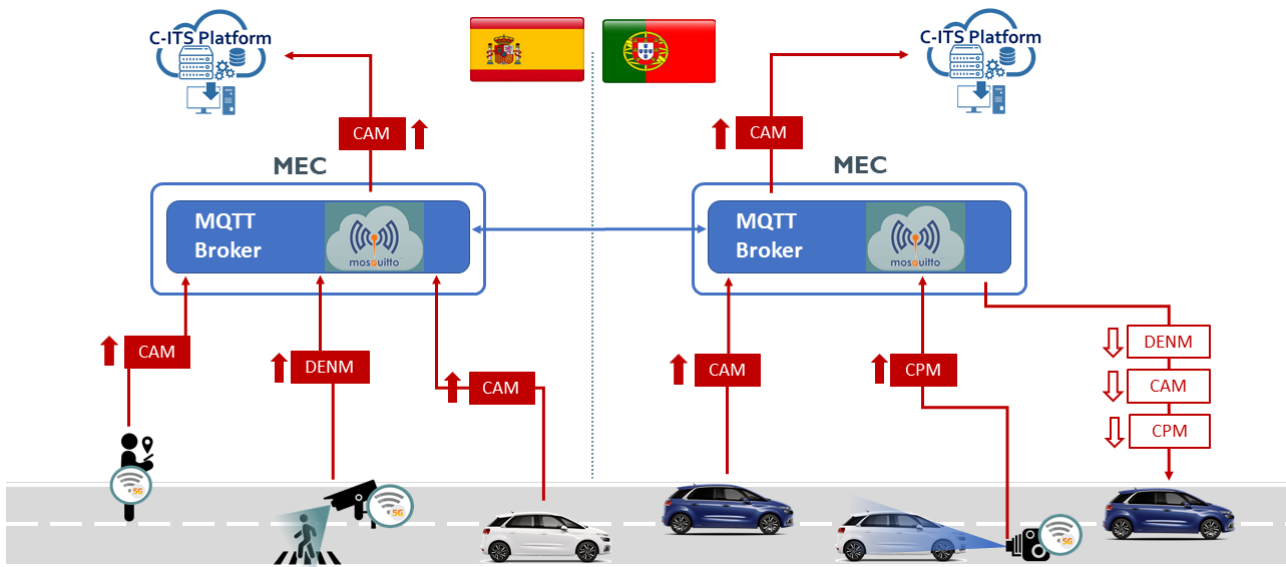


Figure 10: PCOs in an example of Advanced Driving

Table 14: Measurement tools at the three levels of communication for Advanced Driving

	Access layer	Transport layer	Application layer
UE (smartphone)	G-NetTrack Pro	HTTP Canary	Proprietary tool
UE (QoS OBU)	ISEL QoS	ISEL QoS	ISEL QoS
UE (PT CV OBU)	Proprietary tool ⁵ based on chipset library	TCP Dump	Proprietary tool
UE (OBU)	Proprietary tool based on chipset library QXDM	TCP Dump	Proprietary tool
UE (RSU)			
MEC	NA	TCP Dump ISEL QoS	Proprietary tool ISEL QoS
ITS Centre	NA	TCP Dump ISEL QoS	Proprietary tool ISEL QoS

⁵ This tool to be developed will be an in-house developed program to measure the specified network parameters.

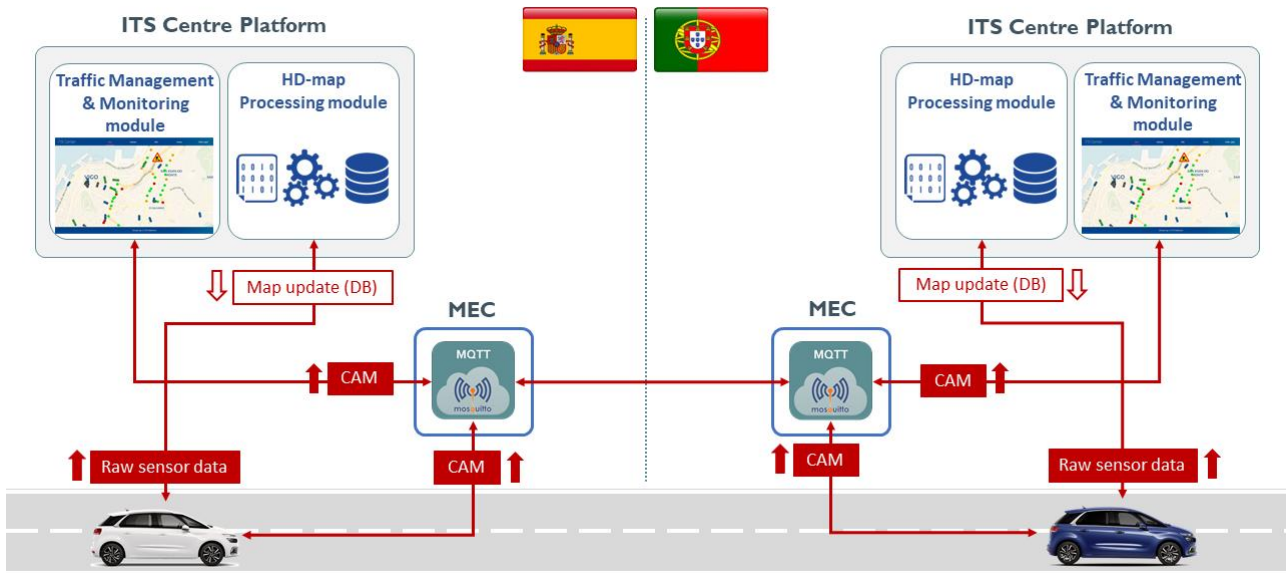


Figure 11: PCOs in an example of Extended Sensors

Table 15: Measurement tools at the three levels of communication for Extended Sensors

	Access layer	Transport layer	Application layer
UE (QoS OBU)	ISEL QoS	ISEL QoS	ISEL QoS
UE (PT CV OBU)	Proprietary tool based on chipset library	TCP Dump	Proprietary tool
UE (OBU)	Proprietary tool based on chipset library QXDM	TCP Dump	Proprietary tool
ITS Centre	NA	TCP Dump ISEL QoS	Proprietary tool ISEL QoS

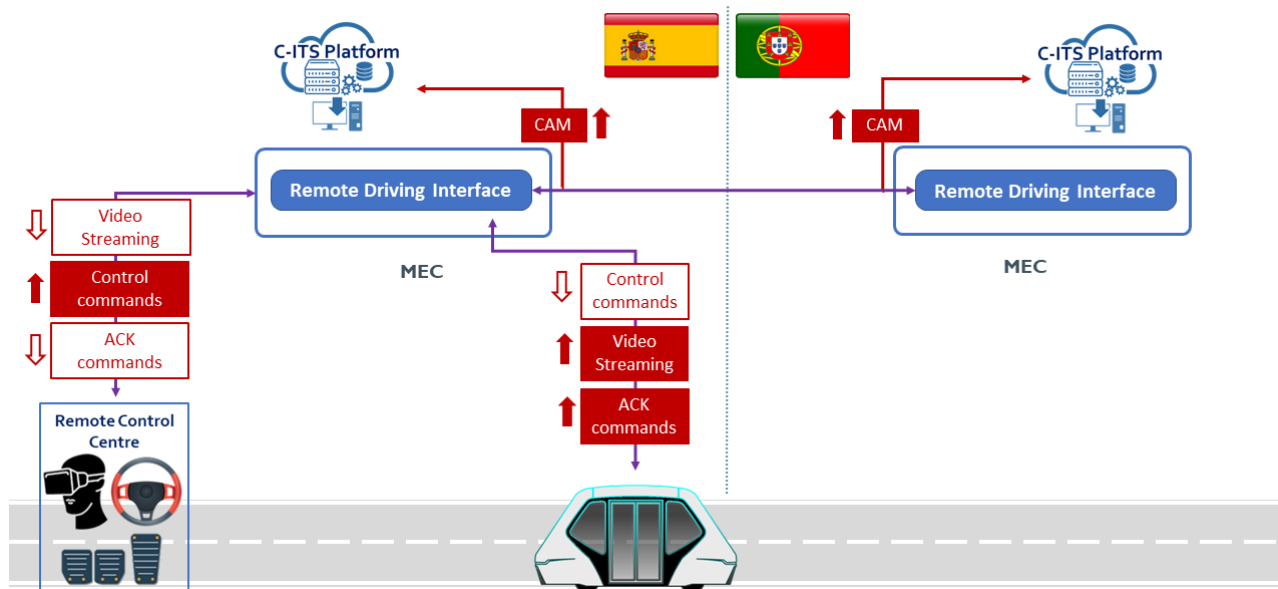


Figure 12: PCOs in an example of Remote Driving

Table 16: Measurement tools at the three levels of communication for Remote Driving

	Access layer	Transport layer	Application layer
UE (OBU)	Proprietary tool based on chipset library QXDM	TCP Dump	Proprietary tool
Remote Control Centre	NA	TCP Dump	Proprietary tool
MEC	NA	TCP Dump	Proprietary tool

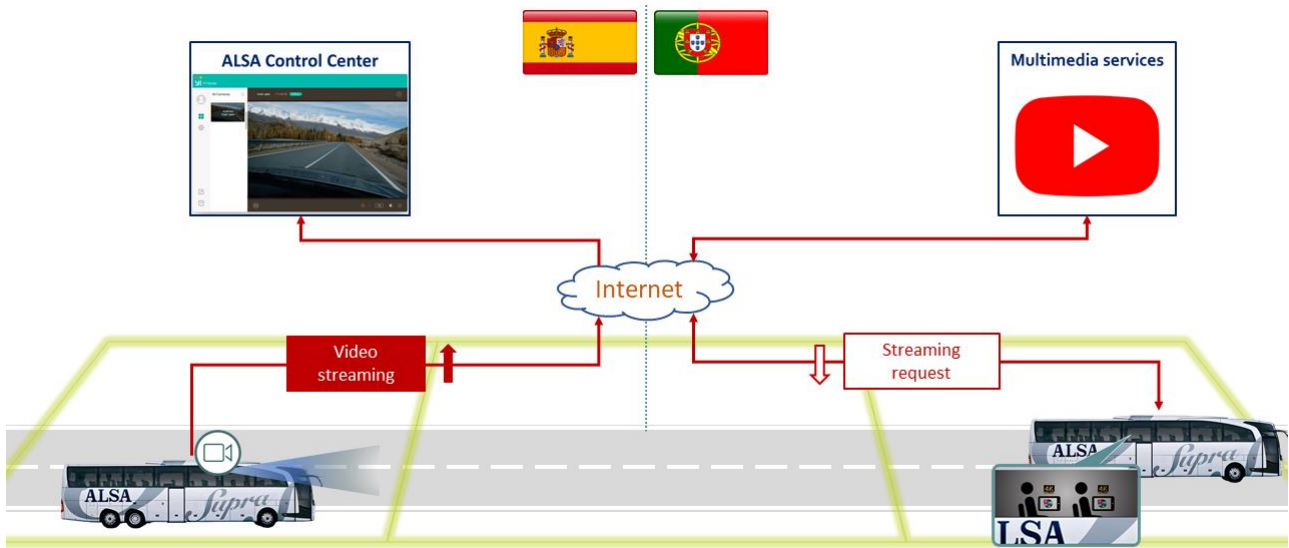


Figure 13: PCOs in an example of Vehicle QoS Support

Table 17: Measurement tools at the three levels of communication for Vehicle QoS Support

	Access layer	Transport layer	Application layer
UE (OBU)	NO	TCP Dump	Proprietary tool
Media Server	NA	TCP Dump	Proprietary tool
Control Centre	NA	TCP Dump	Proprietary tools

4.1.3. ES-PT Data management tools

This section describes the data management tools defined to collect process and provide data for technical evaluation.

As detailed in the previous section, the logging is performed at each PCO within the UCC/US. Figure 14 shows the complete data flow, from the collection of the local logs to their storage on the CTS. Each partner has its own devices to store the data recorded at its PCOs and in addition there is an ES-PT local server to dump and gather all the logs collected during the trials, perform a local quality check and transform them into the common data format. Then, these logs are processed with the Test Data Builder tool in order to perform a second quality check, and also pack the valid ones to be uploaded to the CTS.

In order to fulfil the common data format requirements, a post process is carried out in such a way that the entries in the origin and destination files are matched for *a posteriori* calculation of the aggregated measurements. These additional tools are common for all UCC/US at transport layer, but at application layer one tool is required per UCC, as it depends strongly on the particularities of the data flows (e.g., type of message, communication protocol...).

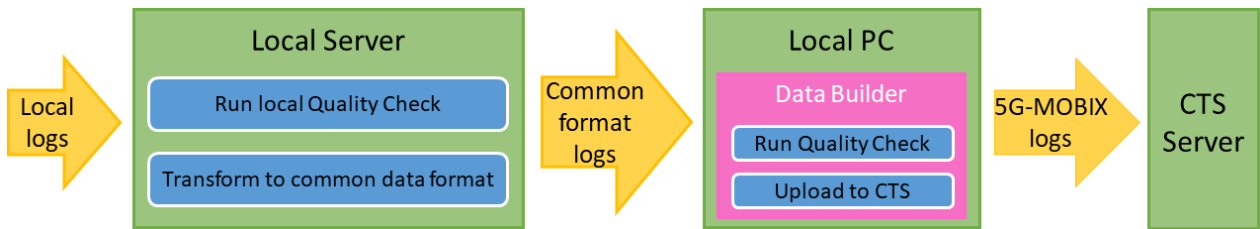


Figure 14: Data Management in ES-PT CBC

4.1.4. ES-PT Test data server architecture

The description of the data management tools in the previous section includes the need for a local server in order to have a common place to: dump the logs provided by each partner, perform the local quality check and also perform the transformation of these logs into the common data format. This local server is hosted by CTAG and consists of a machine with sFTP access and processing capabilities where all the partners have a user to dump and collect information.

The folder structure to store this information is previously agreed (Figure 15) in order to sort the information to make it easily accessible.

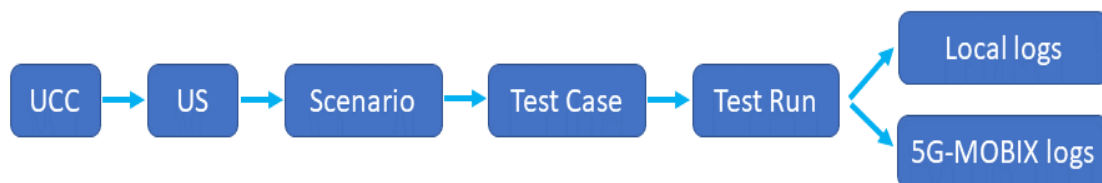


Figure 15: Folder structure in ES-PT local server

4.1.5. ES-PT Partners involved in data collection and management

Each ES-PT partner is working on the integration of a particular PCO in the particular UCC/US architecture and the same partner is responsible for collecting the data associated to that PCO. In addition, this partner is in charge of uploading the corresponding log files to the local TS server. The UCC/US and the partners involved are listed below:

- Advanced Driving / Complex manoeuvres in cross-border settings (lane merge for automated vehicles):
 - UE:
 - Radar (RSU): CTAG (ES side) and IT (PT side)
 - Vehicles performing the manoeuvre (OBU): CTAG (ES and PT sides) and IT (PT side)
 - Vehicle for visualization purposes (PT CV OBU): AtoBe/IT (ES and PT sides)
 - Vehicle for replay data traffic and traffic generation purposes (QoS OBU): ISEL (ES and PT sides)

- MEC (MQTT): CTAG (ES and PT sides)
- ITS Centre: CTAG (for monitoring purposes on ES and PT sides) and ISEL (for replay data traffic and traffic generation purposes on ES and PT sides)
- Advanced Driving / Complex manoeuvres in cross-border settings (automated overtaking):
 - UE:
 - Vehicles performing the manoeuvre (OBU): CTAG (ES and PT sides) and IT (PT sides)
 - Vehicle for visualization purposes (PT CV OBU): AtoBe/IT (ES and PT sides)
 - Vehicle for replay data traffic and traffic generation purposes (QoS OBU): ISEL (ES and PT sides)
 - MEC (MQTT): CTAG (ES and PT sides)
 - ITS Centre: CTAG (for monitoring purposes on ES and PT sides) and ISEL (for replay data traffic and traffic generation purposes on ES and PT sides)
- Advanced Driving / Automated shuttle remote driving across borders (cooperative automated operation):
 - UE:
 - Smartphone (OBU): CCG (ES and PT sides)
 - Shuttle (OBU): CTAG (ES and PT sides)
 - Detector of pedestrians (RSU): CTAG (ES and PT side)
 - Vehicle for replay data traffic and traffic generation purposes (QoS OBU): ISEL (ES and PT sides)
 - MEC (MQTT): CTAG (ES and PT sides)
 - ITS Centre: CTAG (for monitoring purposes on ES and PT sides) and ISEL (for replay data traffic and traffic generation purposes on ES and PT sides)
- Extended Sensors / Complex manoeuvres in cross-border settings (HDMaps)
 - UE:
 - Vehicles performing the manoeuvre (OBU): CTAG (ES and PT sides) and IT (PT side)
 - Vehicle for visualization purposes (PT CV OBU): AtoBe/IT (ES and PT sides)
 - Vehicle for replaying data traffic and traffic generation purposes (QoS OBU): ISEL (ES and PT sides)
 - ITS Centre: CTAG (ES side), IP (PT side) and ISEL (for replay data traffic and traffic generation purposes on ES and PT sides)
- Extended Sensors / Public transport with HD media services and video surveillance
 - UE:

- Vehicle (OBU): CTAG (ES and PT sides)
 - Bus (OBU): ALSA (ES and PT sides)
- ITS Centre: CTAG (ES side), IP (PT side) and ISEL (for replay data traffic and traffic generation purposes on ES and PT sides)
- Remote Driving
 - UE:
 - Shuttle (OBU): CTAG/Nokia ES (ES and PT sides)
 - Camera (OBU): Nokia ES (ES and PT sides)
 - Remote Control Centre (PC): Nokia ES (ES and PT sides)
 - MEC: Nokia ES (ES and PT sides)
- Vehicle QoS Support
 - UE:
 - Camera (router 5G): ALSA/CTAG (ES and PT sides)
 - Mobile device (router 5G): ALSA/CTAG (ES and PT sides)
 - Media Server: ALSA (ES and PT sides)
 - Control Centre: ALSA (ES and PT sides)

To adapt the local log files to the common data format a transformation has to be done:

- Advanced Driving -> DEKRA / CCG
- Extended Sensors -> CTAG
- Remote Driving -> Nokia ES / CCG
- Vehicle QoS Support -> UMU

Each partner is in charge of running the Test Builder tool to perform the quality check and also upload the valid data to the CTS.

4.1.6. GDPR Process/Implementation

As the GDPR implementation has been split by trial site, in this subsection the steps taken to implement the enforcement of the GDPR law in the ES-PT CBC are explained.

First of all, it is worthy highlighting the two basic principles of personal data protection that GDPR defines: the **purpose limitation**, and the **data minimization**. These principles state that personal data must be only collected for specified, explicit and legitimate purposes, and not further processed in a manner that is incompatible with those purposes. Also, no more data than needed for the successful completion of a particular task should be processed.

Trying to unify the GDPR implementation in the context of the project, deliverable D1.4- “Initial Data Management Plan” establishes a common methodology that can be followed to implement the GDPR by the different involved partners. The first step defined in the deliverable D1.4 is the need of appointing the different roles for proper GDPR implementation. The most important figure is the **Data Protection Officer** (DPO) that is nominated at project level. Focusing on this concrete CBC test site, it has been nominated a **Data Controller** (CTAG), and some **Data Processors** (CCG and CTAG). These partners are going to process, store and protect the personal data that is requested by the external participants of the tests. In order to elaborate a detailed cartography of the data processing flow, these partners have filled the requested forms according to the templates indicated in the D1.4 (Section 7.2.2.1), including the Data Protection Impact Assessment (DPIA) where the risks are identified and the protection measures are established.

In order to comply with the **explicit consent** of potential external participants in some of the test cases for user acceptance analysis purpose, an informative sheet with the Project’s objectives, duration, consortium members, funding, etc., is ready to be delivered to the participants prior to performing the tests. In addition to this informative sheet, a consent form specifying the purpose for which their personal data are being collected, whether and with whom it will be shared, and how and for how long their data will be stored is also ready to be delivered, are handed to the participant, that must sign it before participating in any test. As all the tests that are going to be performed are aimed to have the same purposes and objectives, only one form per person is enough for giving the explicit consent to all the set of tests.

4.2. Greece – Turkey (GR-TR) Cross-Border specificities

4.2.1. GR-TR Data logging requirements

This section summarizes the specific requirements applicable to data logging at the GR-TR CBC (see Table 18).

Table 18: GR-TR data logging requirements

ID	Title	Description
LRQ-GR-TR-01	Local storage	The data must be stored at the end of the trial's day in the local format on the local server. Every partner is responsible for dumping the data of its own devices. Responsibilities among partners are depicted in Table 19.
LRQ-GR-TR-02	Common data format	The local logging must be translated into the common data format with a local tool. This is the responsibility of the respective partners (depending on use case and component) as depicted in Table 19.
LRQ-GR-TR-03	Upload of data to	The data must be uploaded to the CTS in the common format

	CTS	and after passing the local quality check. This is the responsibility of the respective partners (depending on use case and component) as depicted in Table 19.
LRQ-GR-TR-04	Integration (RAN – CORE)	All logical definitions must be complete before UCC/US agnostic tests.
LRQ-GR-TR-05	Transmission	Required bandwidth must be ready and connected to TS.
LRQ-GR-TR-06	Radio	All radio related parameters must be checked before UCC/US agnostic tests.
LRQ-GR-TR-07	Permissions	All necessary permissions must be ready before roaming tests.
LRQ-GR-TR-08	Tools – SIM Cards	All testing tools and SIM Cards must be ready prior to initiation of testing.
LRQ-GR-TR-09	Pre-trial logging test	A quick test must be performed prior to starting the trials each day, so that each logging partner can verify the correct logging of information.

4.2.2. GR-TR Measurement tools

This section describes the tools selected for measurements in the GR-TR CBC. Different tools are selected to support different measurements e.g., measuring network capabilities (UCC/US agnostic) against measuring UCC/US specific logging information.

4.2.2.1. Tools for UCC/US Agnostic Test Cases

The network based agnostic tests will be performed by Ericsson GR and TR respectively, following their standard process of drive-tests before delivery of the network for trials, while measurement will also be taken during the trial phase, whenever recalibration or configuration takes place.

At the TR edge, the **Accuver – XCAL tool**⁶ will be used for the UCC/US agnostic field measurements. The Accuver XCAL, one of the most prominent drive test tools, is designed to troubleshoot, monitor, maintain and optimize wireless voice and data network performance – all of these in real-time. The tool is capable of collecting vast amounts of data (1h continuous data collection in the same log file), tailored-made to the needs of the network and the test. Information is collected on all layers (1, 2 and 3), while details are also provided for Message and TCP/IP packet information by interworking with the utilized device and scanner. Some of the key information collected per layer from this tool are:

⁶ http://accuver.com/acv_products/xcal/

- L1: PDCCH/PCFICH/PDSCH, etc.
- L2: MAC/RLC/PDCP, etc.
- L3: RRC/NAS, etc.

The *iPerf2 (TR)*, *iPerf3 (GR)* synthetic traffic tool will be used in UCC/US agnostic tests. See Section 4.1.2.1.

The data streams can be either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).

At the GR edge, the Keysight Nemo Outdoor Platform will be used (See section 4.1.2.1). The measurement setup will include:



- Laptop with Keysight Nemo Outdoor 8.7.0.11
- One Plus 7 pro 5G terminal (with Cosmote FW)
- PCTEL HBFlex Scanner 4G-5G Scanning
- GPS measurements

The tool will measure the following 4G- 5G KPIs:

- Peak DL/UL Throughput
- DL/UL TCP/UDP throughput
- RSRP/RSRQ/ SINR/MiMo Usage
- Various Ping size completion time
- NR RRC Idle -> NR Connected time transition
- Intra Cell Mobility

Keysight Nemo Outdoor allows in all cases to trace all RF signalling from UE to eNB/gNB, hence, is used to parse huge amount of data that is collected by L3 Data Collector (Radio Resource Control, RRC, protocol) for LTE and/or NR network. Sample recording is illustrated Figure 16.

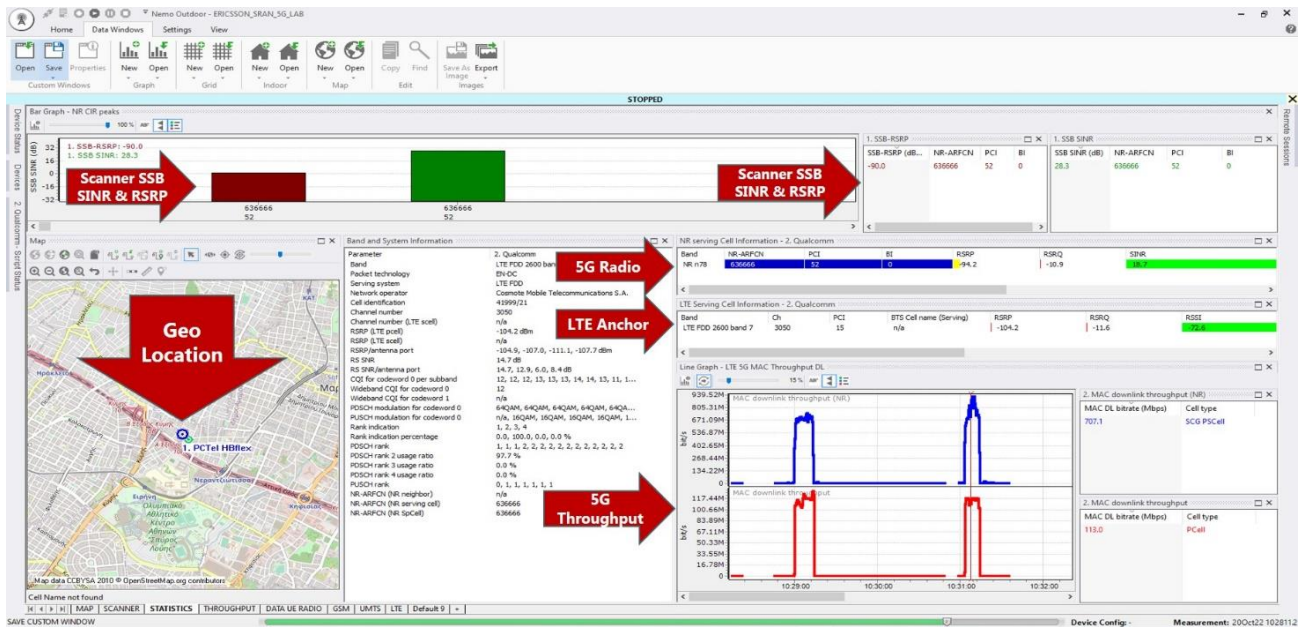


Figure 16: Nemo RF traces example

In both Turkey and Greece CN, the Core Network Operation Manager (CNOM) will be used as KPI extractor, analyser and storage. This tool will extract the PM measurements produced by MME and EPG-C and EPC-U and represent them with graphical statistics. It can further store the KPIs in CSV template. CNOM Works autonomously with pre-defined schedules. It will provide off-the-shelf performance management reports for the available KPIs.

4.2.2.2. Tools for UCC/US Specific Test Cases

The different OBU/RSU and application developers of the GR-TR corridor, namely WINGS, ICCS, IMEC and TUBITAK, have implemented component-specific measurement tools to capture the necessary data during the trials from all their developed components. As the different use cases implemented via these components have different needs, use different sensors, and focus on different KPIs, a compartmentalized approach was necessary. All developed measurement and logging tools adhere to the general 5G-MOBIX evaluation methodology, and all measurements will be transformed to the common 5G-MOBIX data format before being stored to the Central Test Server (CTS). The description of the respective measurement/logging tools is provided below.

1) IMEC Measurement tools– Platooning, Truck Routing, See-What-I-See UC

IMEC has a lot of experience in experimenting with several wireless technologies including V2X long-range (e.g. 5G and 4G C-V2X Uu) and short-range C-V2X PC5 and ITS-G5. Hence, it has a lot of expertise in developing measurement tools that can be used for assessing the performance of different communication technologies, the communication capabilities of the wireless devices under test, supporting and enabling use cases, logging data and post processing them for evaluation of defined KPIs.

In order to facilitate the experimentation and the centrally management the measurements and logging, IMEC has designed and developed a vehicular communication management framework, also known as CAMINO. CAMINO is designed to be flexible and dynamic, aiming to become the core framework for managing multiple vehicular communication technologies and the services running on top of them. Figure 17 presents the overall architecture of the CAMINO framework. As is shown, CAMINO resides in the middle and then there are the northbound and southbound interfaces.

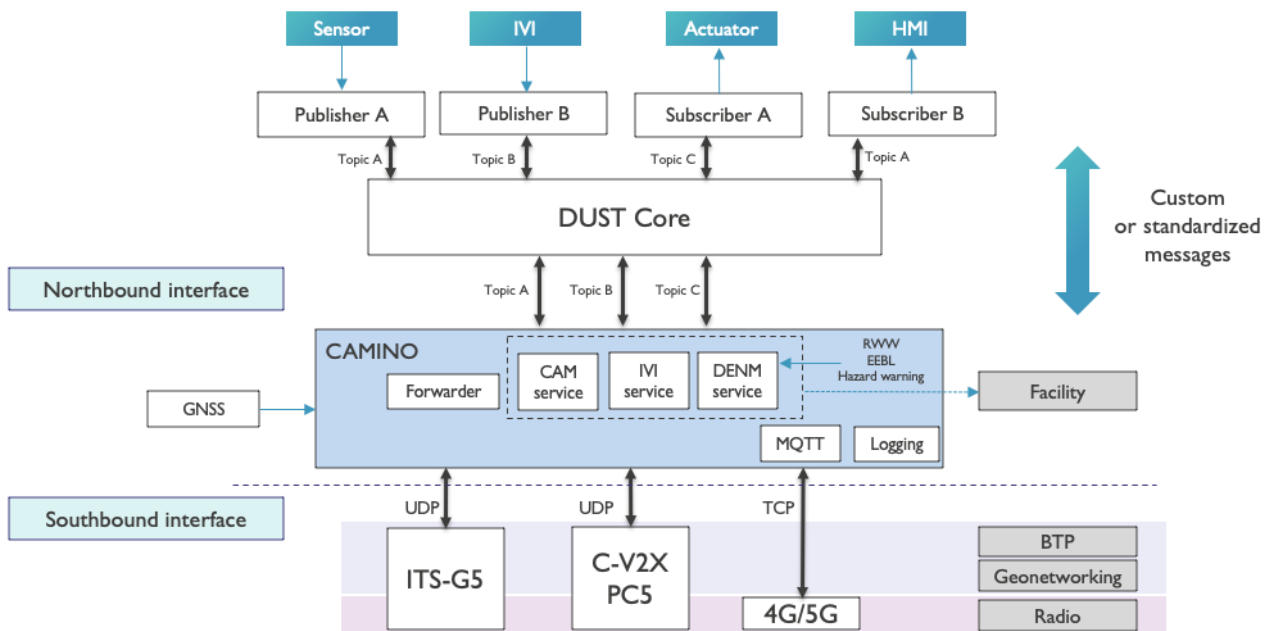


Figure 17: CAMINO architecture

At the southbound interface, CAMINO interconnects with the different V2X wireless technologies. For example, we can see that it can communicate with the C-V2X PC5 module via UDP socket and with an MQTT broker via a 4G/5G radio over TCP.

The lower layers of the ITS stack such as the BTP/GeoNetworking and radio are managed by the V2X wireless modules, while the facility layer of the ITS stack is managed by CAMINO.

CAMINO allows different services to run concurrently, each transmitting standardized messages such as CAM, IVI and DENM. Additionally, it allows the transmission of custom messages, while it also uses the forwarder service that may forward the messages coming to CAMINO via any V2X technology. Thus, in the context of 5G-MOBIX project, CAMINO will be extended to support the exchange of V2X messages as defined by each UCC.

At the northbound interface, CAMINO will be interconnected with a vehicle's HMIs, sensors and actuators by using a publisher/subscriber framework (DUST Core [9]). This way, publisher-scripts can be used to trigger specific services at CAMINO, based on information that may derive from the CAN BUS or the

sensors of the vehicle, while subscriber-scripts may relay the information from the different ITS services in the vehicle. This information then may be used to trigger an actuator or to be visualized using an HMI.

A configuration file allows an experimenter to configure the technologies that he/she would like to use, including C-V2X PC5 and C-V2X Uu (through 5G), as well as the characteristics of the transmitted packets, such as the size of the packets, the inter-packet interval, etc. CAMINO allows logging at communication unit level of the packets that are transmitted and received between the different nodes of each technology. The data are logged locally and among other information may include the timestamp of the message generation at the transmitter side, the timestamp of reception at the receiver side, the coordinates of the vehicle or the RSU, the IDs of the ITS stations, etc.

Linux-based networking tools such as iPerf, ping, Wireshark, as well as scripting using bash, python and other languages will also be used for the evaluation of wireless communications within 5G-MOBIX. Such tools will be used for throughput and latency measurements of the 5G network. Linux-based tools will be used in order to collect and log network related (e.g., cell ID) and signal related parameters (e.g., RSRP, RSSI...) that will further assist in the evaluation of the 5G network being developed within the GR-TR pilot.

2) ICCS Measurement tools - See-What-I-See UC

This section describes the tools selected for the measurements of the See-What-I-See application. As can be easily understood by the use case description, the crucial parts of evaluating the integrity of the See-What-I-See streaming application are the processes of making sure that two particular KPIs are monitored and met. In particular, we need to make sure that the *E2E latency* is as low as possible, so that the actual video displayed by the following vehicle is as close to the real time view of the leader as possible (in the realm of hundreds of milliseconds with an upper-bound of 200ms). Secondly, we need to be able to define the high resolution (4K) of the video displayed by the following vehicles. In the use case description, it is explicitly defined that the 4K resolution is essential for minimizing the anxiety of the driver [10].

LEVIS Client/Server functionality

LEVIS application is a platform that enables cloud-based video streaming between two ends, namely sender and receiver, at low latency. 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 See What I See (SWIS) user story at GR-TR corridor is depicted in the Figure 18 below. From this figure, we identify the following different components:

- **Leader Vehicle (LV):** it represents the stream sender. On the LV, a Single Onboard Computer (SOC) is deployed, which consists of a Jetson Xavier that has a camera attached to it. Both the Xavier and the LV are connected to outside world through the same OBU.

- **Streaming Server:** it is located at ICCS premises. It receives the stream from the LV and makes it available to some other consumers.
- **Follower Vehicle (FV):** it represents the stream (sent from the LV) consumer or receiver. The FV receives the stream sent by the LV through the cloud server (at ICCS premises).

SWIS user story consists of a platoon of trucks, in which we have one leader and one or more followers. In this scenario, the driver of the follower vehicle is the initiator of the live stream. Accordingly, the driver of the leader vehicle can either accept or reject the live stream request. In case of stream request rejection, the initiator will be notified by the follower decision. Similarly, at the acceptance of the request, in addition to sending the notification, the live stream is sent to the server. Accordingly, the server makes the received stream from the LV available for follower vehicles. As soon as the stream arrives to the server, the follower vehicle is notified and accordingly it starts the player that connects to the server to display the live stream.

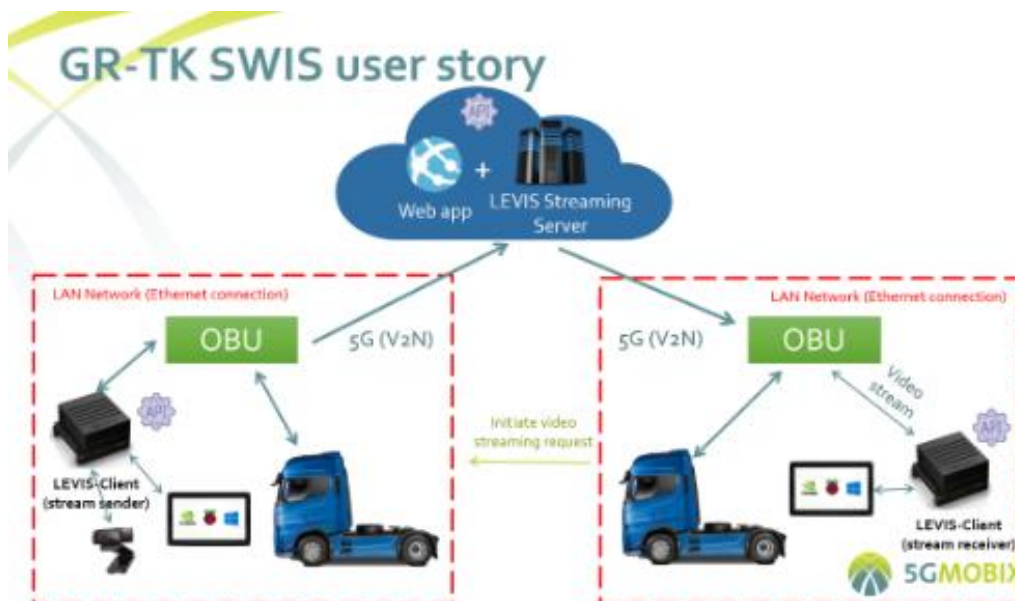


Figure 18: Remote driving user story deployment scenario

An important part of the process consists of being able to measure the E2E latency of the UDP control commands for orchestration between the LEVIS Server, the LEVIS client and the ICCS management module. In order to achieve this, every command has a unique identifier and a time stamp at the moment of being issued towards the respective module of the architecture. For measuring E2E latency, the commands are acknowledged in the receiver by sending a new CMD-ACK with the same sequence number (unique identifier) and associated timestamp. Once the CMD-ACK arrives RTT (Round-Trip Time) can be computed as the system has the information regarding the two timestamps and the sequence number.

- Step by Step process for measuring latency (disregarding the part of the orchestration)

- Use a tablet running a clapperboard application
- Film it with a camera connected to the video encoder part of the LEVIS client
- Publish the video stream to the origin (LEVIS Server)
- Deliver to the display through a CDN
- Put the display alongside the clapperboard tablet
- Take a picture of the two screens
- Perform the timecode subtraction and obtain the metric

It should be noted that the TCPdump/TShark are going to be the logging tools of choice. When testing IP streams, the best way to arrive at a pristine reference is to use the same encoder to create the reference as you do to create the downstream deliverables. When you reduce the resolution using the target processing device to match the test profile's resolution at the highest possible bit rate, you minimize the differences created in picture artefacts when scaling and deinterlacing an image. While this process slightly compromises the video quality of the source, it is the only way to ensure that the reference and the low-bit-rate versions are being deinterlaced and/or scaled according to the same algorithms, and the best way to get a quality measurement that is as true as it can be to the index that is being used. Distinct steps for measuring the video streaming quality delivered at the on-displays of the following vehicles:

1. Generate a mezzanine-quality reference for each profile using the highest possible bit rate and optimal encoding parameters.
2. Generate each profile test signal using the application's encoding parameters.
3. Calculate quality using the MS-SSIM on DMOS scale, comparing each test profile signal to the appropriate reference profile signal.
4. Analyse results on DMOS scale against visual comparisons of source and downstream profile test segments.

Finally, a plethora of tools will be provided such as those offered by the PRTG to monitor the bandwidth and the upstream and downstream of the application, along with the possibility of detecting packet drop. The correlation of the packet drop ratio and the video resolution is strong and recent research work [11] has shown the elevation of discomfort from the viewer to be really high when video frames are absent from the sequence (packet loss) and in comparison with the diminished video resolution effect on discomfort.

3) WINGS measurement tools - Assisted "zero-touch" Border-crossing

Both the WINGS OBU and RSU, utilized for the *Assisted "zero-touch" Border-crossing* use case as well as the application server are Linux based. Hence, they can host any other Linux compatible logging tool desired by the application developer. Currently, our implementation uses the following measurements tools:

- Python scripts/time measurement functions (OBU/RSU)
- Java log4j2 logging package (Application server)

Both implementations output their logging per executed test, using timestamped files that are automatically stored on the WINGS server. The files are formatted as delimited log files, allowing for easy parsing and post-processing, which is especially handy for the later transition of KPIs to the common data format. The information/measurements logged by the files are:

- **OBU/RSU logging**: timestamp, IP, IMEI, ECU info (speed, RPM, temperature), sensor info (CO₂, temperature, distance (Lidar), humidity, luminosity, NFC response), GNSS info (LAT, LON), Human detection (binary), Transmitted packets, dropped packets, Delay (s), E2E latency of the received response (s).
- **Application/server logging**: Timestamp, CO₂, Distance, LAT, LON, ECU info, IP, IMEI, Sending time, Response, Server Process Time, Transmitted Bytes.

4) TUBITAK Measurement Tools – Truck routing UC

Truck Routing Application runs on Robot Operating System (ROS) - Linux distribution. Therefore, any ROS package or Linux compatible tools can be used in order to measure and manage data when needed. Currently, the implementation uses the following for logging and measuring on the application server:

- Python functions and scripts
- ROS functions and packages

The application automatically stores logging files with timestamps on the local file system. These logs will be formatted and parsed in order to support a common data format. Our logs will contain the following fields:

- Timestamps, truck IPs, truck info (plate number etc.), forwarded waypoints, events (truck registration, disconnection, reaching the destination etc.) truck location, latency, response time, time passed until destination reached.

4.2.3. GR-TR Data management tools

Each of the GR-TR partners engaged in measurements and logging, will store and properly label the data after each experimentation. Each partner uses proprietary servers for local storage of the data. After a consistency check of the data and their transition to the common data format, all data will be uploaded to the common CTS (after the end of the experimentation session) under the responsibility of each partner, respectively. More details regarding the data management tools of each GR-TR partners are provided below.

Ericsson data management tools – Agnostic UCC/US measurements

A similar tool as the one used for the UCC/US agnostic measurements, i.e. the Accuver – XCAP⁷ tool, will also be used for post processing for. This tool allows for a deep analysis and detailed reporting on the collected data gathered with the Accuver XCAL tool (see Section 4.2.2.1). Some of the main functionalities of Accuver XCAP tool are:

- Quick troubleshooting and conduct root-cause analysis of wireless networks
- Provide detailed reports on service-specific and overall network performance
- Benchmark radio and application-level metrics
- Build and present KPI reports across the enterprise
- Analyse trends and perform complex statistical analysis
- Visualize service and network-level performance geospatially

Ericsson TR and Ericsson GR will be responsible for verifying RAN performance. For that purpose, four sites in Turkey and one site in Greece will be tested individually, and UCC agnostic tests will be performed before UCC/US specific tests execution. Ericsson TR and Ericsson GR will be responsible for storing log files. Ericsson TR and Ericsson GR will be responsible convert data into common format.

Figure 19 illustrates log processing procedure.

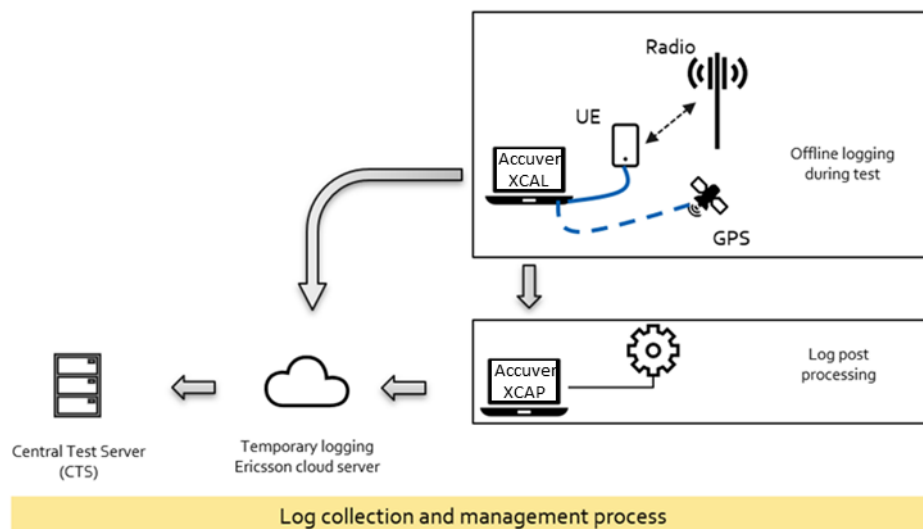


Figure 19: Log collection and management process

⁷ http://accuver.com/acv_products/xcap/

Collected log files will be stored offline in the test laptop during UCC/US Agnostic tests. Accuver XCAP tool will be used during tests. After completion of every test, collected log file will be copied to Ericsson Cloud Server. This will be a manual process.

+ New Upload Share Copy link Sync Download Automate				
My files > 5G-Mobix_GR-TR				
Name	Modified	Modified By	File size	Sharing
Eskisehir Trial Site	October 13		1 item	Shared
GR_Data	October 13		0 items	Shared
TR_Data	October 13		0 items	Shared
My files > 5G-Mobix_GR-TR > Eskisehir Trial Site > TC01				
Name	Modified	Modified By	File size	Sharing
112_TCP_DR02123602-M1-M1.drm		ersan.algul@turkcell.com.tr	164 MB	Shared
112_UDP_DR02123602-M1.drm		ersan.algul@turkcell.com.tr	236 MB	Shared

Figure 20: Collected log

Ericsson TR and Turkcell, Ericsson GR and Cosmote will be responsible for data quality check. Once the tests are complete, collected log files will be processed using Accuver XCAP tool (applicable for TR) and Keysight Nemo (applicable for GR). Prepared report and processed logs will be copied to Ericsson Cloud server for future use and backup. This will be a manual process. It's also applicable for CN data collection using CNOM tool.

Central Test Server (CTS) will be used as final destination of the logging procedure. Ericsson TR and Ericsson GR will be responsible for copying all these files to the CTS Server. Data quality check tool will be use if it's allowed.

IMEC data management tools – Platooning, Truck Routing, See-What-I-See UC

Section 4.2.2 describes the measurement tools that will be used from IMEC in the context of the Platooning, Truck Routing and See-What-I-See UC. The same tools will be used for management of the logged data. During an experiment, initially the IMEC OBUs and RSUs will log all the collected data locally. Such local logging will ensure data will not be lost due to potential connectivity loss between the OBUs, the RSUs and the remote logging servers. The logged data will be stored according to the agreed common data format. Subsequently, the local data will be logged remotely to both the IMEC server and the CTS server. The logging to the CTS will allow the post-processing of the logged data for the evaluation of the

UC, while the IMEC server will be used for back-up reasons. GDPR regulations will be applied to data logged to ensure anonymity of the data. For uploading to CTS, we will use Test Data Builder and CTS website provided by AKKA and documented in Section 5.

WINGS data management tools - Assisted “zero-touch” Border-crossing

The measurement and logging tools to be used as well as the logged data for this use case are discussed in Section 4.2.2. The server and OBU/RSU logging mechanisms ensure that each experimentation file will be saved both locally (local server and OBU/RSU), as well as on the WINGS server with the proper title and metadata to be able to uniquely identify each experiment. As the WINGS servers are continuously and automatically backed-up, loss of experimentation data are ruled out. Additional HTTP requests can be utilized via an API that will receive requests with appropriate inputs (use case, dates, etc.) and return data on the predefined format, thus guaranteeing continuous and ubiquitous access to experimentation data, for authorized users.

The experimental setup including the WINGS Cloud platform, where the data management tools are also stored is depicted in Figure 21. For uploading to CTS we will use Test Data Builder and CTS website provided by AKKA and documented in Section 5.

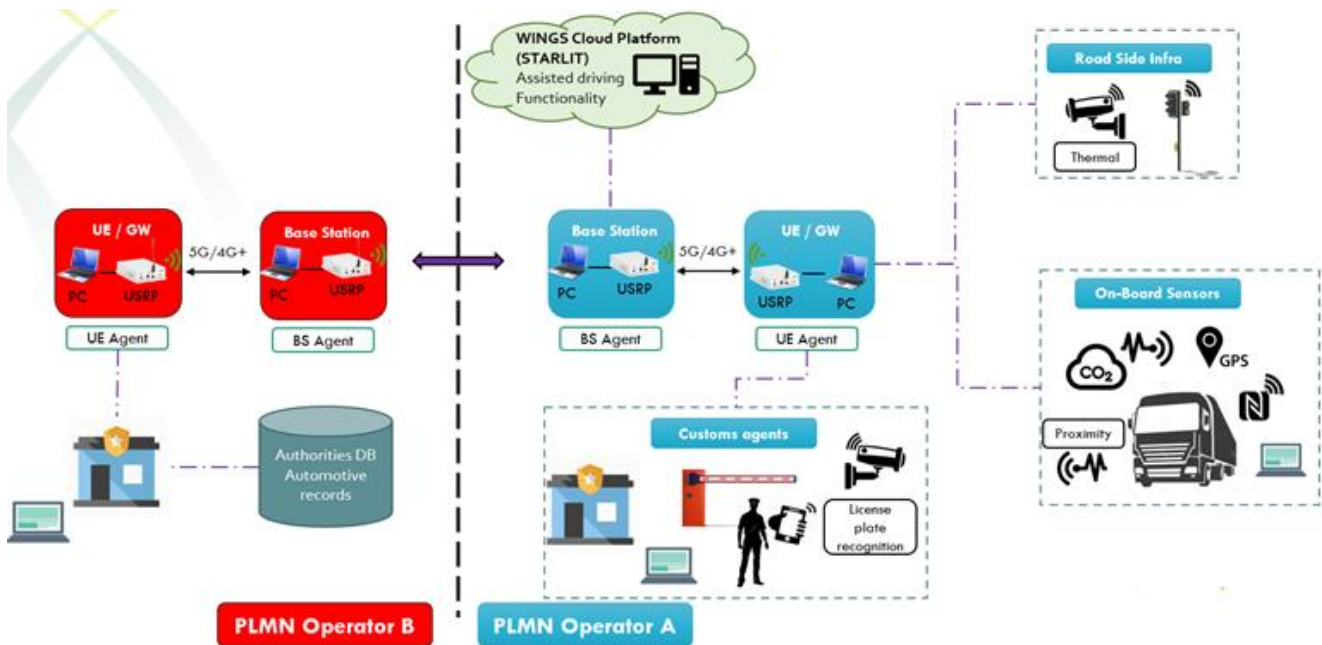


Figure 21: Assisted “zero-touch” Border-crossing architecture

4.2.5. GR-TR Partners involved in data collection and management

As the E2E trials at the GR-TR CBC involve multiple HW and SW components, developed and operated by different stakeholders, the data management and collection in the GR-TR CBC is treated as a delegated process. Each of the participating partners is responsible for the proper logging, labelling and storage of the data defined according to the guidelines of T3.5 and WP5, for their respective part in the trials. Each partner is also responsible for transforming the logged data into the commonly agreed data format of 5G-MOBIX and uploading them to the Central Test Server (CTS) after each experimentation session. The exact responsibilities are shown in Table 19.

Table 19: Data collection and management responsibilities in the GR-TR CBC

GR-TR data collection responsibilities	
Network data	Ericsson TR + GR, Cosmote, Turkcell
Platooning Application, OBU and RSU data	IMEC, Ford
See-What-I-See Application data	ICCS, AALTO
Assisted Border-Crossing Application, OBU and RSU data	WINGS
Truck Routing Application, OBU and RSU data	TUBITAK

4.2.6. GDPR Process/Implementation

For the GR-TR corridor testing and trialling activities, Turkcell will ensure that the correct procedures to exercise full compliancy with the GDPR are followed for the collection of personal data, since the operator has a team of experts from a variety of backgrounds, including legal and technical domains, who collectively are responsible from the protection of personal rights at Turkcell⁸. The distribution of roles in accordance with the GDPR is described in data management plan (D1.6 – “Interim Data management Plan”).

The project work and the flow of user stories are introduced to this team to get their initial view on how the GDPR process should be implemented. Based on the information given, their feedback is that, with the current setup, it is unlikely for the GR-TR partners to capture personal data, which requires getting the consent of individuals (i.e., the data owners), due to the fact that the trial will be using a 5G test network (no subscription and thus no personal data) and the user stories that perform video transmission will not store the streams that might involve personal data. During the trials, the directives received from this team will be used to get the necessary consent from all individuals who are affected by or require some kind of interaction with the activities of the corridor whenever necessary.

⁸ Although it is not a member of the EU, Turkey also has a similar regulation called KVKK as here: <https://www.kvkk.gov.tr/en/>

4.3. German (DE) Trial Site specificities

4.3.1. DE Data logging requirements

This section summarizes the specific requirements applicable to data logging at the trial site level (see Table 20).

Table 20: DE data logging requirements

ID	Requirement	Description
LRQ-DE-01	Instant Handshake	The communication between spontaneously co-located systems must be driven by the infrastructure as a default trust system to provide tokens or certificates to allow data exchange and prevent from environment hacking
LRQ-DE-02	Data Encryption	The data transmission between instantly communicated systems is protected by tokens or certificates
LRQ-DE-03	Data validity	Past data not being consumed is automatically discarded after some time to avoid real/time systems to discard out of dated data
LRQ-DE-04	Data integrity	Vehicles and RSUs generated data include fast checksum mechanisms to allow resilience and error checks
LRQ-DE-05	Data units	Data schema include sampling rate, scale, units and nominal bitrate in order to avoid inconsistencies when fusing/combining different data from different vendors and systems
LRQ-DE-06	Geographic metadata	Data schema include a global reference to support the area being covered to relate field of view/field of sensing to the other sensing systems and to facilitate the frame-to-world translation
LRQ-DE-07	Fast Data process	Data compression format or data schema/structure facilitates the random access or navigation to specific areas or time
LRQ-DE-08	Data compression	Data are compressed to alleviate data throughput communications and perform fast/quick transmission
LRQ-DE-09	Streams compression	Streams are compressed to alleviate data throughput communications and perform fast/quick transmission. Furthermore, encoding settings target low latency without frames requiring presentation reorder
LRQ-DE-10	Data sync	Data captured simultaneously to data streams are compressed together to ensure frame accuracy synchronization and timely inputs
LRQ-DE-11	LOD Data	Geographic data structure and order facilitate fast and simple LOD techniques to generate subsampled data
LRQ-DE-12	Data	Data compression facilitates fast and simple multi-resolution techniques

	resolution	to generate subsampled data
LRQ-DE-13	Computing capacity	TS must provide processing capacity to parse and deliver up to 4 vehicles data streams concurrently
LRQ-DE-14	Timestamp	Universal Time Coordinated (UTC) is proposed as the single time reference (with high resolution, at least 1ms)
LRQ-DE-15	Analysis Trigger	Test leader must assure and confirm that all log data, meta data and test description are uploaded to the TSTS for every test execution, including contributions from all partners and systems acting in the test execution. This confirmation will trigger the checks and analyses of the test execution and a first feedback to the test leader on the data quality and success (or failure) of the executed test.
LRQ-DE-16	Log Format	All log data are uploaded to the TSTS in common log format from AUTOPILOT, INTERCOR or C-Mobile, as specified in " https://ada1.tno.nl/logformats/ "
LRQ-DE-17	Redundancy	Data logging should be redundant to avoid data loss
LRQ-DE-18	Dual storage	Partners should log data from their developed devices and store them internally in their organization + one central repository per TS/CBC should exist where all data are also stored.
LRQ-DE-19	Data quality check	After each test partners should check and verify that the data have been correctly stored (uncorrupted), in the proper format and with all required fields.
LRQ-DE-20	Data origins	Logged data should be clearly marked with their origin network, platform, device, model or version, in order to trace back easier.

4.3.2. DE Measurement tools

In the DE TS different tools have been selected and adapted in order to produce the metrics capabilities and the KPIs. Different approaches are followed because, regarding a concrete component of the network, an RSU for example, the origin of the metrics is different. The tools are listed and described below:

1) ModemManager

ModemManager is a tool for controlling mobile broadband modems. It includes an interface for the command line, called mmcli, which allows to interact and control the modem. In the DE TS, this tool is used to set-up and interact with the 5G modems installed in OBUs and RSUs. First, the ModemManager tool is started in debug mode, allowing advanced features such as the transmission of AT commands to the modem. Secondly, through the mmcli interface in the command line, a large variety of requests can be sent to the modem. This request can contain AT commands or options, defined in the mmcli

documentation. The modem will respond to this request with information that will be useful for level of relevant measurements. Some examples of this logging information are received signal strength and current radio access technology.

2) WebRTC stats

The stats object was implemented with the aim of monitoring video, audio or data packets in a peer-connection in WebRTC applications. It provides many metrics from both sides of a peer-connection, transmitter and receiver. In the DE TS it is especially useful in order to monitor video streams transmissions between network components. For example, the stats object provides metrics like bytes transmitted or received, round-trip time and jitter.

3) iPerf3

See Section 4.1.2.1.

4) Tshark

It is a command line tool included in Wireshark software used to capture and analyse network data packets from the available interfaces in a device. When it is called with the right options, it can be used to read from a packet capture (PCAP) file, apply the desired filters, and export the data in a specific JSON format that is valid to be ingested by an Elasticsearch database. The last feature is what makes this tool so useful for post-processing the data once it is stored in the database.

5) Node exporter

This is a tool that collects tons of metrics relative to the device where it is running and expose them in an IP:port end-point where a Prometheus server can collect them. Node exporter provides metrics from all network interfaces, such as bytes sent/received by each interface, CPU and memory used, and many more relevant to the OS.

6) Grok exporter

Originally, Grok is a tool used to provide data in a form that Elasticsearch⁹ can store, query and index. It extracts and parses data from unstructured log files to a structured format. In the case of the Prometheus Grok exporter, the Grok tool is used with the goal of extracting unstructured data and exposing them in the Prometheus metrics format in this case, so that they can be scraped by Prometheus from an endpoint. To achieve this task, the Grok exporters need to be configured to search for text patterns that match with the structure from the log files. Grok patterns are defined by the two parameters SYNTAX and SEMANTIC as %{SYNTAX:SEMANTIC}. The parameter SYNTAX contains the name of the pattern that is needed, for example a number, a date, an IP, etc. SEMANTIC is the name given to the variable that is extracted. This

⁹ <https://www.elastic.co/fr/elasticsearch/>

and more information about Grok patterns can be found in Elasticsearch⁹ official website, as it is actually a plugin that they have developed.

4.3.3. DE Data management tools

This section describes the data management tools used in the data collecting process and provide evaluation data for technical evaluation.

1) Prometheus

Prometheus is an open-source project originally developed to monitor the music social media Soundcloud. It is now widely used to monitor different types of applications, as it is relatively simple to use and very flexible. It is a full monitoring system that stores data as time series, which means that each sample is saved with a timestamp, organizing the data in a timeline. It also has the advantage of being supported by its community, making easy to find all kind of tutorials for use cases that can be adapted to one's needs. It includes a potent query language, called PromQL, which easily allows the aggregation and representation of the collected time series. In Figure 23, an example of the Prometheus architecture for the DE-TS is depicted.

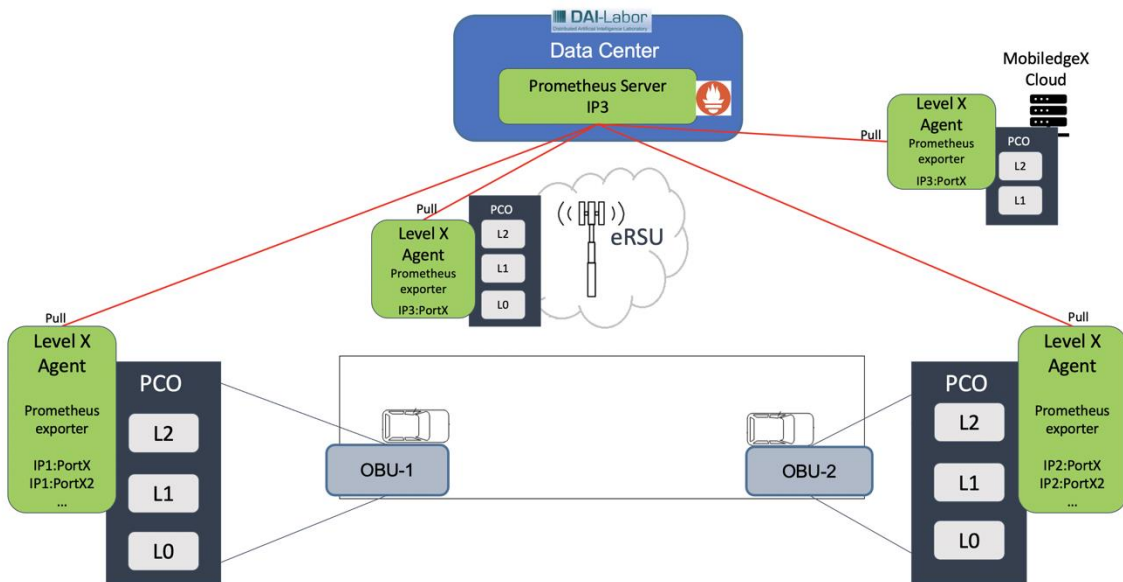


Figure 23: Prometheus server architecture for data collection

2) Grafana

Grafana is an open-source software used to create dashboards where data can be visualized and analysed. Data from different sources are available for Grafana by indicating in its configuration the end-point where they are located, so that they can be queried, plotted and even aggregated. It is also possible to create alerts depending on data values. Grafana offers different visualization options, where many features can

be configured as desired. There are also many dashboard templates and plugins available on the Internet that can be reused and adapted for the desired use cases. Figure 24 depicts a panel created with Grafana showing the end-to-end bitrate.

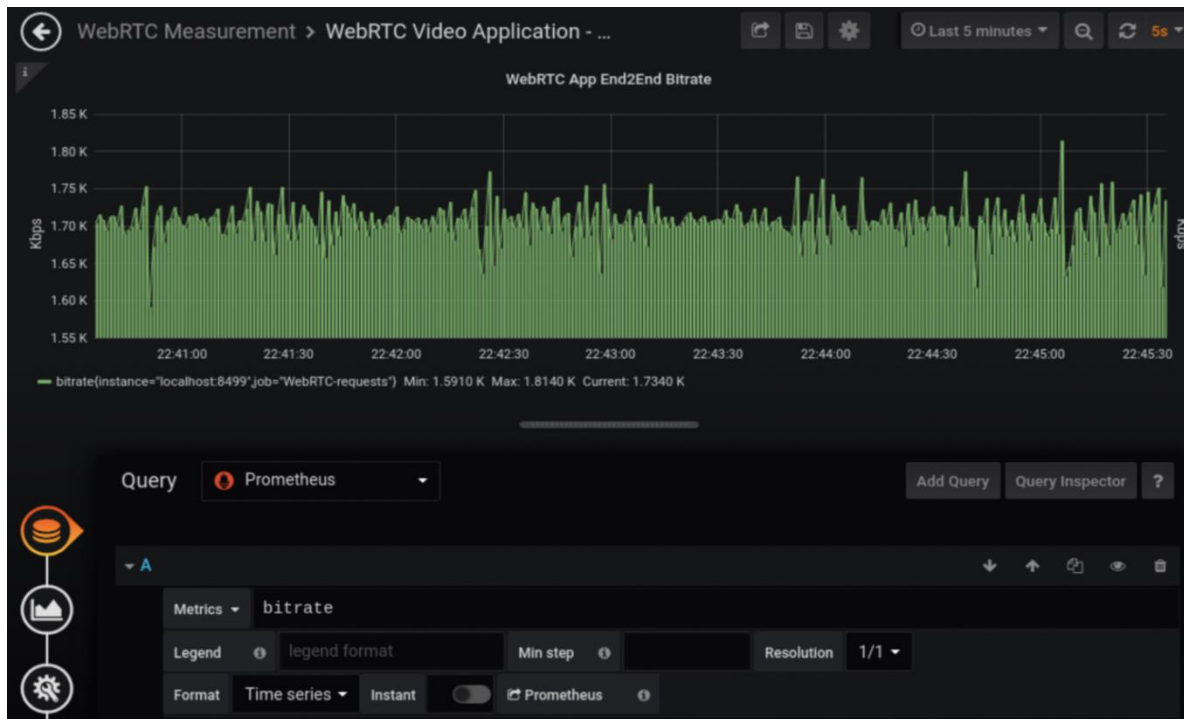


Figure 24: Grafana panel showing End2End bitrate metric collected with Prometheus as data source

3) ElasticSearch

ElasticSearch is an open-source tool that has many features, among the most interesting are its capacity for searching, aggregating, indexing and storing huge amounts of data from different ecosystems. The data are stored in different indices that are based on encoded JSON documents, which makes it one of the fastest searching engines.

4.3.4. DE Test data server architecture

This section contains the description and architecture of the local trial site test data server or management facility.

In Figure 25, the designed data management architecture for the DE-TS is shown. For data storage, two instances of Elasticsearch databases are deployed in the data centre. In the first database, various indices are created to store all data collected following the first approach, which uses Prometheus to scrap data from the end-points in each component of the DE TS, and following the second approach, which extracts the relevant information of captured data packets transmitted via PC5 interface. In the first database, all collected metrics are available, which are then processed and filtered, in order to save them afterwards in

the second database, which only contains metrics regarding the common data format and a certain quality level. Data in the second database should be ready to be uploaded to the CTS.

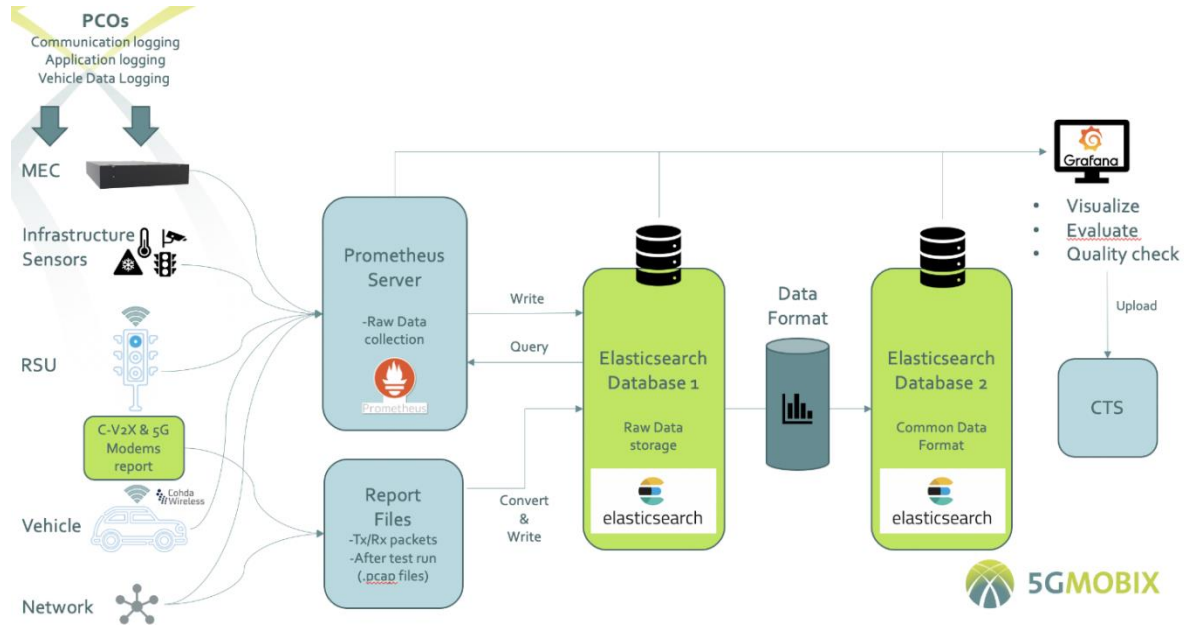


Figure 25: General architecture of the data management in the DE TS

4.3.5. DE Partners involved in data collection and management

Table 21 indicates the partner which is responsible for accomplishing the following tasks:

Table 21: DE data collection and management responsibilities

DE data collection responsibilities	
Data collection (Data manager)	TUB
Save, centralize and store the data at TS level after each trial	TUB
Verify and perform data quality check	TUB
Identify who will transfer data to the CTS	TUB

4.3.6. GDPR Process/Implementation

The German trial site develops and demonstrates an automotive surround view system. Video data and non-personal metadata extracted from the video (e.g. location and velocity of vehicles) are streamed to nearby vehicles and or infrastructure nodes at the roadside. The shared data augments the situational awareness of the drivers in the recipient vehicles - e.g. informs about obstructions in the driving lane – and may advise actions to the driver.

What data are collected?

In order to develop the surround view functionality and the data transmission system, video footage of typical vehicle surroundings in public traffic is required. This includes traffic signs, traffic lights, lanes and road edges, pedestrians, bicyclists, powered two-wheelers, cars, trucks, etc.

How data are collected?

VALEO will conduct video recordings of public traffic by means of two distinct camera systems:

1. a front camera installed behind the vehicle's windshield. The approximate maximum field of view of the front camera is 75 degrees.
2. a system of 4 fish-eye cameras whose combined field of view covers the complete 360° surrounding of the vehicle.

The VALEO vehicles are clearly marked as testing vehicles. Moreover, they are equipped with stickers informing about potential recording activities including a QR code linking to the VALEO data protection measures.

How do we use the data?

The recordings are used for the development of the DE TS use cases, i.e.:

- "Surround View Generation with Connected Vehicle Sensors"
- "Edge-Dynamic-Map (EDM)-enabled Surround View Generation with Connected Vehicle and RSU Sensors"

In the data processing, the camera system extracts non-personal data about the objects in the vehicle's surrounding. These data include the position, velocity and object type (car, truck, pedestrian, bicycle, motorcycle, etc.). The extracted data does not contain any personal information.

How is the data stored?

The recorded video data are securely stored on encrypted drives with access restricted to stakeholders of the DE TS involved in the development of the envisioned functionality.

In the final application, the video data will not be stored, but only buffered in the camera system for the time required to complete the data processing (typically tens of milliseconds). Once the video data are processed, the raw footage is overwritten and deleted.

No personal data (e.g. faces, license plates, observation time and location of people and vehicles, etc.) is ever evaluated or stored.

The following are the key actors from the DE TS perspective:

- DPO: ERTICO as project coordinator
- Data controller: VALEO and VICOMTECH as partners with surround view camera systems in their vehicles.
- Data processor: VICOMTECH and VALEO as each partner can render the surround view from the other vehicle in his vehicle.

The assignment of the data controller and data processor roles depends on the use of the VALEO/VICOMTECH vehicle as surround view video publisher and subscriber, respectively. The roles may vary in the course of the trials, as both partners' vehicles are capable of fulfilling these tasks.

4.4. Finnish (FI) Trial Site specificities

4.4.1. FI Data logging requirements

This section summarizes the specific requirements applicable to data logging at the trial site level (Table 22).

Table 22: FI data logging requirements

ID	Title	Description
LRQ-FI-01	Data recoverability	In the event of corruption of data, it must be possible to restore uncorrupted data from back-up to a suitable point to resume processing
LRQ-FI-02	Data accessibility	Trial/CBC sites may have the possibility to retrieve data from CTS generated from corresponding UCC/US of other sites for benchmarking etc.
LRQ-FI-03	Data flexibility	The local data processing and storage facilities should be able to ingest data from devices and services that may represent data using different information models.
LRQ-FI-04	Verifiability	Trial /CBC sites must have means to verify that the local processing of the data does introduce any erroneous alterations prior to submission to CTS.
LRQ-FI-05	Timestamp uniformity	Universal Time Coordinated (UTC) must be used single time reference for all data logging.
LRQ-FI-06	Data usability	All data should be uploaded to the CTS in common log format from adopted within the project
LRQ-FI-07	Test Run	Test run is a single execution of a test case. Log data and meta data are uploaded to the TSTS per test run

		<p>and does not contain data of other test runs.</p> <p>Meta data includes a report on the executed test, including a list of participating vehicles, users, and any deviations to the planned test session such as system failures or deviations in vehicle trajectories or events.</p>
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.4.2. FI Measurement tools

The tools selected for FI TS measurements comprise mostly tools for different network (UCC/US agnostic) KPI measurements, as well as a few custom tools for UCC/US specific measurements. The use of multiple measurement tools allows for cross-validation of measurement results and capturing of KPIs or visualisations which are otherwise unavailable in one common tool. The network measurement tools include the following:

- Professional drive/walk-test tools:
 - DEKRA TACS₄ performance tool described previously in Section 4.1.2.1;
 - Keysight Nemo Handy and Nemo Outdoor tools described previously in Section 4.1.2.1. It is noted that Keysight tool availability is usually limited by time-constrained license (this is a specific constraint for FI-TS), so will only be acquired for occasional measurement campaigns.
- Android-based free network measurement apps:
 - Speedtest app¹⁰ from Ookla;
 - Netradar app¹¹ developed initially within AALTO and now maintained by a spinoff company (see screenshot in Figure 26).
- Open-source software tools:
 - Wireshark;
 - Ping and traceroute;
 - iPerf₃ described previously in Section 4.1.2.1.

¹⁰ <https://www.speedtest.net/>

¹¹ <https://www.netradar.com/>

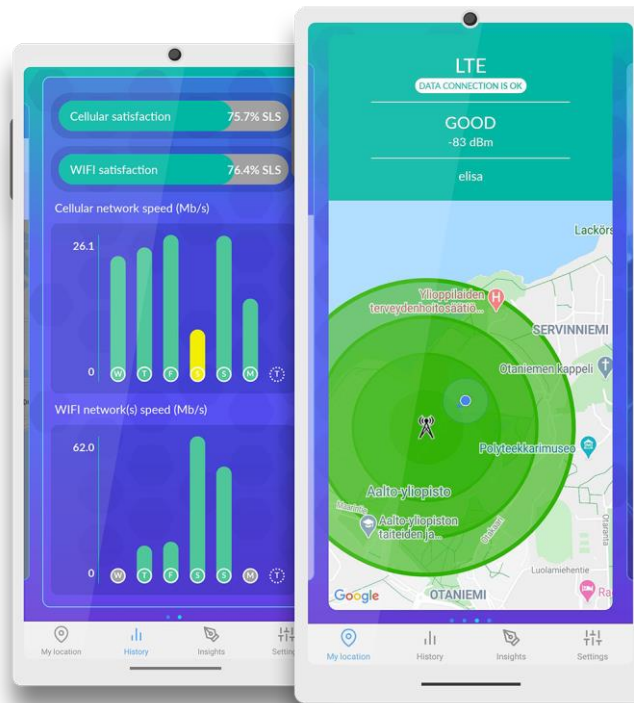


Figure 26: Netradar app

In addition to network measurement tools a few UCC/US, specific tools are also being developed. Notable among these is a custom AALTO measurement setup for capturing specific KPIs (frame loss, glass-to-glass latency etc.) for video streaming (e.g. used in remote driving user story). Example results from measurements conducted with this custom setup is shown in Figure 27 below.

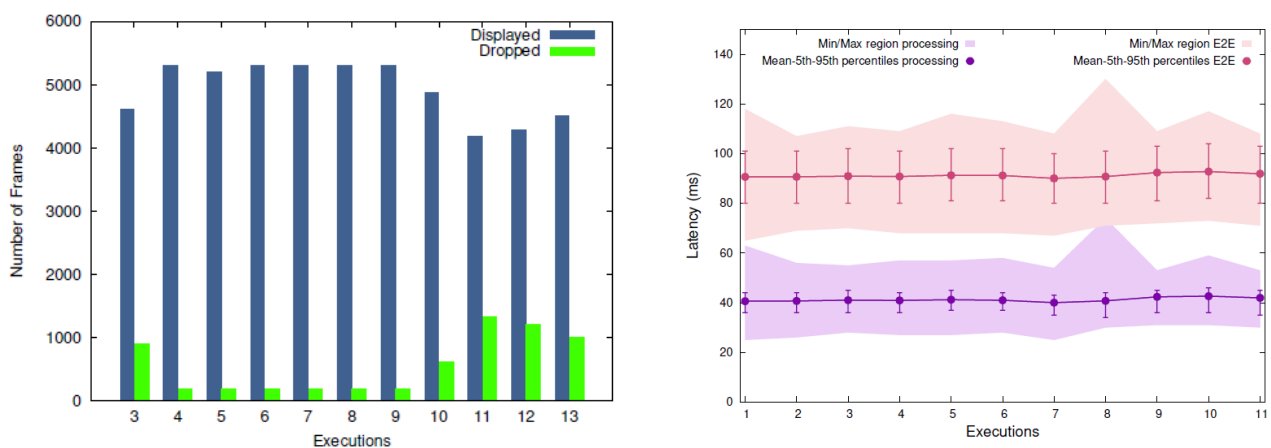


Figure 27: Results on frame drop (left) and latency (right) from measurements of KPIs using custom tool for LEVIS video streaming application

4.4.3. FI Data management tools

The local test server for storage of FI TS measurement data will utilise an appropriate file structure (organised by UCC/US, evaluation scenario, test case etc.) to simplify management of measurement data files in different phases of the trials. The local test server is based on the TeamWork¹² file storage service provided and maintained by Aalto University IT services. A TeamWork storage space has been created for 5G-MOBIX and is only accessible by authorised project researchers from within the campus network or externally via VPN connection. The visualisation of the collected measurement data are done using GUI provided by respective measurement tools or using visualisation tools provided by general purpose computing tools (e.g. Matlab). Finally, the FI TS will utilize a combination of manual, custom and project provided tools (described in Section 2.4) for data quality checks prior to data submission to CTS.

4.4.4. FI Test data server architecture

The FI TS leverages a simplified arrangement for collection and storage of measurement data. The distributed architecture for measurement data gathering and storage is UCC/US dependent, with some slight differences in the utilised PCOs. As an example, Figure 28 and Figure 29 illustrate the logging carried out in different PCOs for the extended sensors and remote driving user stories, respectively. The upload of measurement data are done manually from each measurement tool platform. For instance, in the case of the DEKRA TACS₄ tool, the upload of measurement data will be done from the TACS₄-controller to the local test server.

¹² <https://www.aalto.fi/en/services/file-storage-space-for-research-and-groups-teamwork>

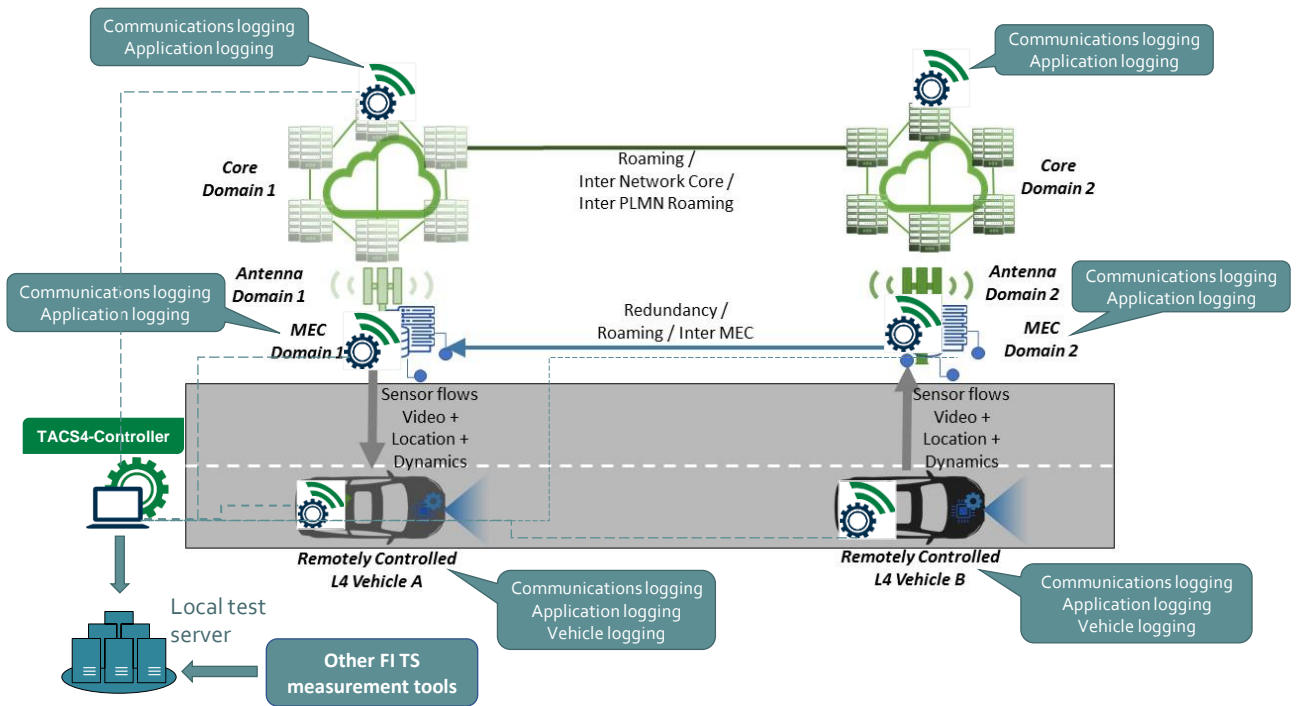


Figure 28: PCOs in FI TS extended sensors with redundant edge processing user story

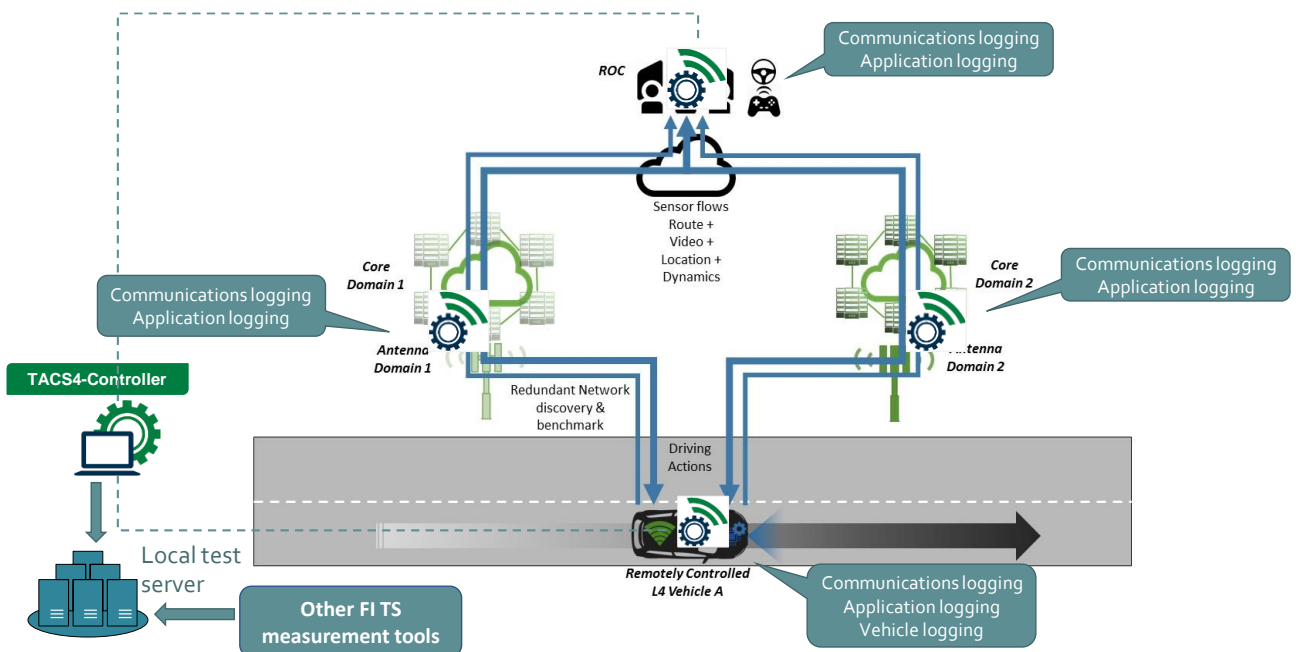


Figure 29: PCOs in FI TS remote driving in redundant network environment user story

4.4.5. FI Partners involved in data collection and management

The partners involved in data collection and management processes from FI TS perspective and their roles are listed below:

- Initial specification of Test Scenarios and planning Test Sessions for each of the two user stories:
 - Extended sensors: AALTO as user story owner
 - Remote Driving: SENSIBLE₄ as user story owner
- Logging of test data during Test Run: AALTO & SENSIBLE₄;
- Upload Test Session data to Local Test Server: AALTO & SENSIBLE₄;
- Completion of test reporting after Test Runs: AALTO & SENSIBLE₄;
- Ensuring compliance to agreed 5G-MOBIX data format and use of data adaptors: AALTO & SENSIBLE₄;
- Manual upload to the Central Test Server: AALTO.

4.4.6. GDPR Process/Implementation

A number of processes and direct interventions are employed to ensure GDPR compliance for the FI TS developments and trials. The GDPR assessment for FI TS is informed by local procedures and guidelines including *Aalto Personal Data Processing Data Protection Security Guide* and *Aalto's privacy notice template*. A preliminary assessment of potential ethical issues has noted the following:

- Both FI TS user stories involve the recording of video footage from moving vehicles. The images or video captured may be considered 'personal data', if it contains images of people that could be used to identify these people either directly or indirectly (when combined with other pieces of information).¹³ This would raise valid ethical concerns.
- The participants in both FI TS user stories will exclusively be members of research staff from AALTO and SENSIBLE₄. Therefore, this removes the requirement of obtaining informed consent which is only necessary enabling third party persons to participate as subjects in a research or investigative activity.

Based on the assessment above of the following two compliance measures are instituted and will be followed throughout the project duration for FI TS.

- The videos captured for FI TS user stories are solely for identification and localisation of objects in a monitored driving route and is no way intended for identification and/or tracking of specific person(s). In guaranteeing this declared intention, the FI TS leader (AALTO) takes specific measures to ensure careful

¹³ European Data Protection Supervisor advisory note on video surveillance https://edps.europa.eu/data-protection/data-protection/reference-library/video-surveillance_en

archiving and processing of the collected videos and images in accordance with stated principles Article 5 relating to processing of personal data (Article 5 of GDPR¹⁴).

- In relation to above, the selection of the local test server is also done with GDPR compliance in mind. This includes ensuring that data access to the server is audited and data are deleted after defined time.

The process of ensuring GDPR compliance is a continuous one and requires clearly assigned roles. The following are the key actors from the FI TS perspective:

- DPO: ERTICO as project coordinator
- Data controller: AALTO as leader of FI TS
- Data processor: AALTO and SENSIBLE⁴ each as a user story owner

4.5. French (FR) Trial Site specificities

4.5.1. FR Data logging requirements

This section summarizes the specific requirements applicable to data logging at the FR trial site level (Table 23).

Table 23: FR data logging requirements

ID	Title	Description
LRQ-FR-01	Test Plan	Test Plan must define test sessions, and per session the purpose of the test (what is different from other tests), intended vehicle trajectories and virtual or real events, ...
LRQ-FR-02	Test Run	Test run is a single execution of a test case. Log data and meta data must be uploaded to the TSTS per test run and must not contain data of other test runs. Meta data includes a report on the executed test, including a list of participating vehicles, users, and any deviations to the planned test session such as system failures or deviations in vehicle trajectories or events.
LRQ-FR-03	Upload	Log data generated by systems should be uploaded shortly after the execution of every test run to the TSTS
LRQ-FR-04	Analysis Trigger	Test leader must ensure and confirm that all log data, meta data

¹⁴ Regulation (EU) No 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679>

		and test plan are uploaded to the TSTS for every test run, including contributions from all partners and systems acting in the test run. This confirmation will trigger the checks and analyses of the test run and a first feedback to the test leader on the data quality and success (or failure) of the test run.
LRQ-FR-05	Analysis Check	Test leader must upload (Test Run) and trigger the analysis (Analysis Trigger).
LRQ-FR-06	Log Format	All log data must be provided in the common log data formats version 0.7.11 as specified on [6]. This is the 5G-MOBIX update of the INTERCOR, C_MOBILE and AUTOPILOT log formats.
LRQ-FR-07	Log Verification	Test leader and Data Manager must verify and approve all test run log data and meta data (Test Run), e.g. on sample data, before a pilot or test session can be officially run and before test run data will be checked and analysed (Analysis Check)
LRQ-FR-08	Test data description interface	Test data uploaded to CTS must comply with the predefined test description interface (test data xsd interface)

4.5.2. FR Measurement tools

This section describes the tools selected for the measurements. Different tools can be selected to support different measurements e.g., measuring network capabilities (UCC/US agnostic) against measuring UCC/US specific logging information. Different tools can be even selected by different UCC/US within each trial site.

For the FR TS different tools were chosen in order to measure the network performances and the KPIs. Different approaches are used depending on the feasibility and the availability of the tools on the entity where the KPI is calculated. In the following, we present the different possibilities:

1) 5G Modem monitoring and control tool (gcom)

gcom¹⁵ is a tool to monitor the board as well as the network to which it is connected. Gcom tool is a built on a tool from OpenWRT. This tool allows us to get data from the 5G modem on demand. Through the mmcli interface in the command line, a large variety of requests can be sent to the modem.

2) 5G modem manager

5G modem manager is a web interface tool offered by the board. It can show several KPIs related to the different modems linked to the board (Figure 30).

¹⁵ <http://www.linuxcertif.com/man/1/gcom/>



Figure 30: FR TS 5G modem manager tool

3) iPerf

This tool is already defined in Section 4.1.2.1

4) Wireshark

Wireshark is a network packet analyser.

5) TCPdump

The description of this tool is already provided in Section 4.1.2.2

4.5.3. FR Data management tools

During FR TS trials, VEDECOM will use its server/client based data management framework which will use the tools cited in previous sections to collect, process and provide data to technical evaluation. For that purpose, an evaluation framework is designed based on a server/client architecture as illustrated by Figure 31 and Figure 32.

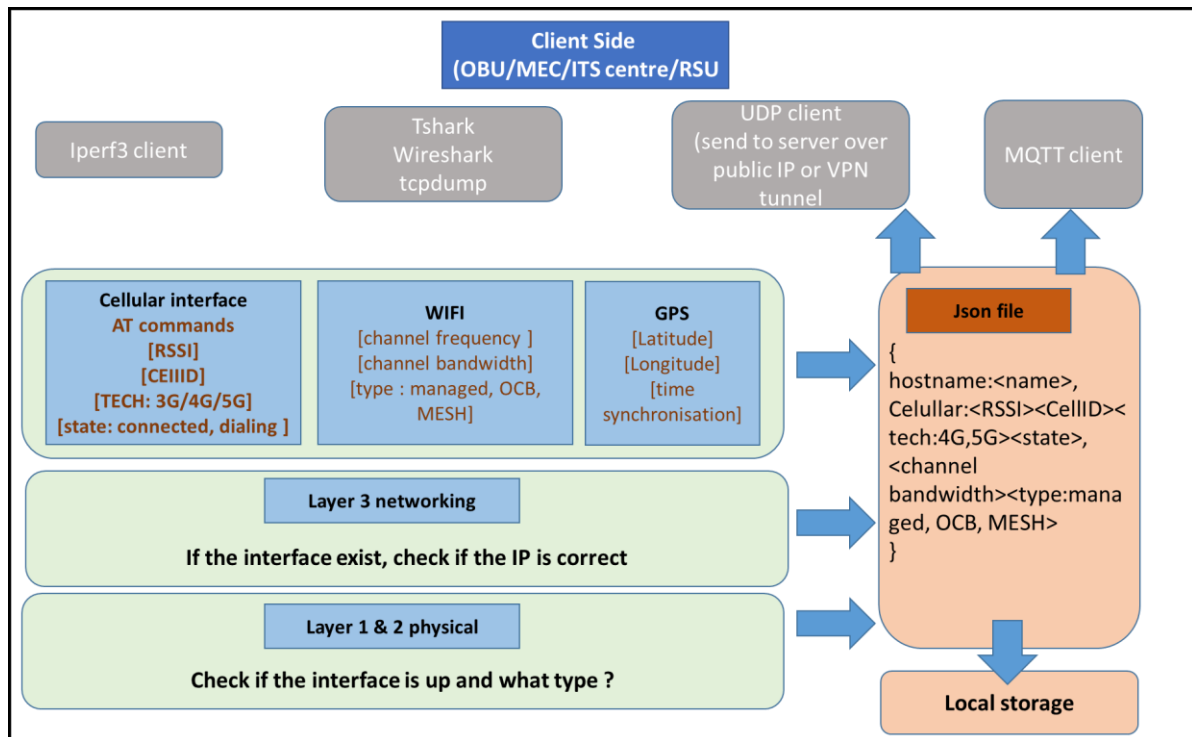


Figure 31: FR TS data evaluation client architecture

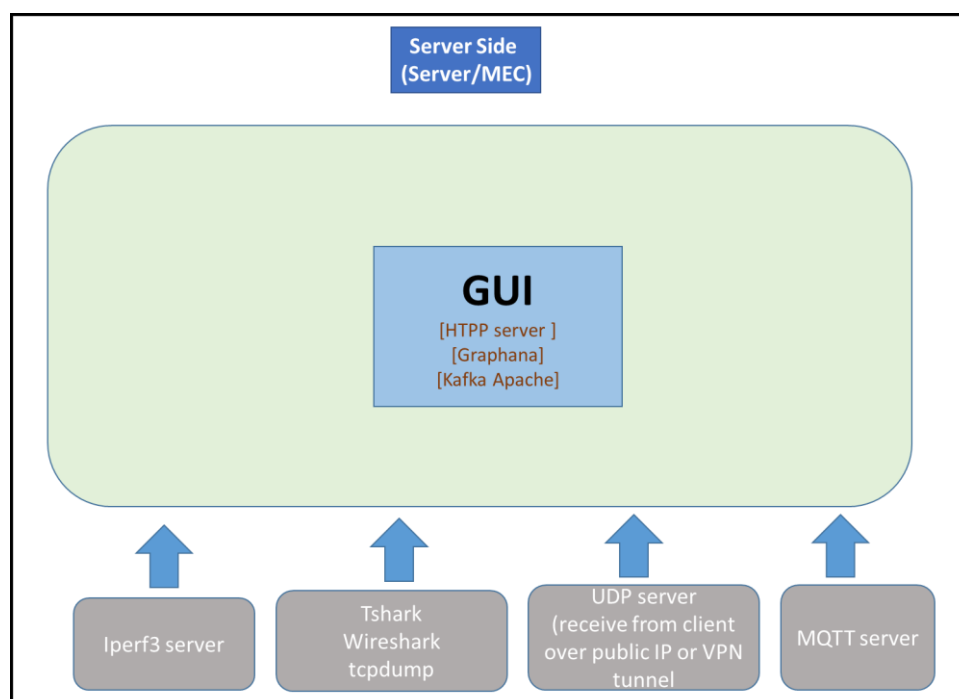


Figure 32: FR TS data server evaluation architecture

The client architecture presents the different PCOs that will be defined in each of the communication layers. This will be implemented in each device (OBU, MEC, etc.) taking part of the tests. Different tools will be used to collect data or to carry out agnostic tests, as previously mentioned.

For example, for the 5G-OBU, at the physical layer, developed tools will monitor periodically the 5G modem and check for:

- APN
- RSSI
- CellID
- Technology [3G, 4G or 5G]
- State [Attached, Connected, Dialling]

This data will be logged every second alongside with the GNSS coordinates. This combination will allow the server to know the network coverage for each coordinate in the map.

At the client side, the logged data will be provided in a JSON file and saved locally before it is sent to the server.

In the server side, server version of the applications already installed will be deployed (iPerf server, MQTT server, UDP server, etc.) and they will be receiving the data, logging it and also displaying it on a HMI.

4.5.4. FR Test data server architecture

When dealing with test data storage and management, FR TS is deploying a local data storage server in which the local data, issued from tests and trials, will be stored. First, the logging data collected from the different clients and at different PCO levels are collected and uploaded to the data collection and logging server and formatted to a local format. Then, the local formatted data will be transferred to a local data storage for a permanent storage. In addition, a common data format technique will be applied to have the required format to be uploaded to the CTS (see Figure 33).

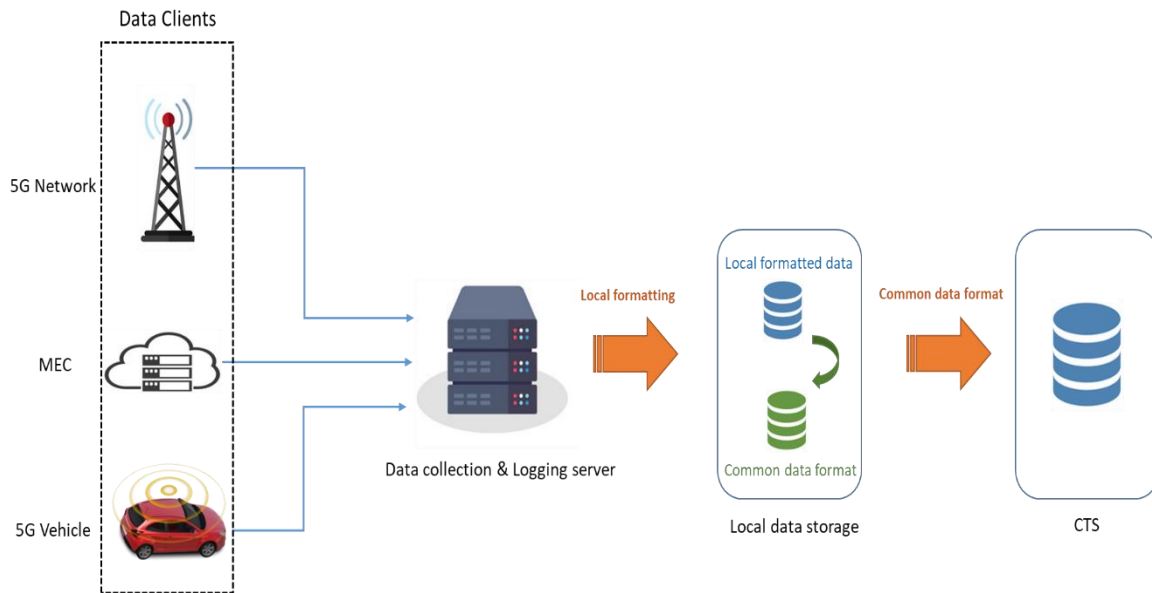


Figure 33: General architecture of the data server in the FR TS

4.5.5. FR Partners involved in data collection and management

Table 24 identify the different persons/entities that will be responsible for carrying out the following tasks:

Table 24: FR data collection and management responsibilities

FR data collection responsibilities	
Data collection operation (Data manager)	VEDECOM
Data saving, centralisation and storage at TS level after each trial	VEDECOM
Data quality check verification	VEDECOM
Data transfer for evaluation to the CTS	VEDECOM

4.5.6. GDPR Process/Implementation

FR TS is demonstrating a highway advanced driving scenario in which only vehicles will be used to demonstrate a lane changing manoeuvre. During tests and trials, videos extracted from cameras and metadata extracted from LIDARS are transmitted to the MEC server to augment the perception of the vehicle. During these trials, even if the vehicle is on an autonomous mode, the driver must be in the vehicle to intervene only when it is needed. For that purpose, and to comply with the explicit consent of potential participants, an informative sheet with the Project's objectives, duration, consortium members, funding, etc., is ready to be delivered to the driver prior to performing the tests.

The participants in FR TS user story will exclusively be members of research staff from VEDECOM. Therefore, this removes the requirement for obtaining informed consent which is only necessary for enabling third party persons to participate as subjects in a research or investigative activity.

The following are the key actors from the FR TS perspective:

- DPO: ERTICO as project coordinator
- Data controller: VEDECOM as leader of the FR TS
- Data processor: AKKA, CATAPULT, Orange/Ericsson, Bouygues, TDF, VEDECOM as they are FR TS partners

4.6. Dutch (NL) Trial Site specificities

4.6.1. NL Data logging requirements

This section summarizes the specific requirements applicable to data logging at the trial site level (Table 25).

Table 25: NL data logging requirements

ID	Title	Description
LRQ-NL-01	log formats	All log data must be provided in the common log data formats version 0.7.11 as specified on https://ada1.tno.nl/logformats/ . This is the 5G-MOBIX update of the INTERCOR, C_MOBILE and AUTOPILOT log formats
LRQ-NL-02	log data repository	All log data, meta data, and analysis results are stored on the NLTS data repository on https://ada1.tno.nl/5g-mobix . Access to this repository is secured and restricted to NLTS partners
LRQ-NL-03	uploading	All log data must be uploaded to the NL TS data repository on https://ada1.tno.nl/5g-mobix/upload/ . Data can be uploaded automatically or manually as specified on https://ada1.tno.nl/5g-mobix/data-provisioning.php
LRQ-NL-04	experiment	An experiment on the ADA repository is a single test run of a test case. All log data must be organised per experiment and uploaded in the experiment folder in the log data repository
LRQ-NL-05	meta data	Once all log data are uploaded for an experiment, the meta data must be provided on https://ada1.tno.nl/5g-mobix/data-provisioning.php
LRQ-NL-06	trigger experiment analysis	Once all log data and meta data are provided, the analysis of the experiment data must be triggered on and experiment analysis

		must be triggered
LRQ-NL-07	experiment analysis	Analysis of all experiment data are executed on the NLTS data repository. This included data quality checking, filtering, aggregation and uploading all relevant data to the Central Test Server
LRQ-NL-08	experiment results	Results from the experiment analysis are accessible on the website of https://ada1.tno.nl/5g-mobix/

4.6.2. NL Measurement tools

The measurement tools used in the NL TS provide logging for technical evaluation of the user story at the application level.

To check and compare the communication performance with Task 3.3 and Task 3.6 tests, the same generic communication network tools are used for the Use Case Agnostic measurements: iPerf3, ping and TCPdump.


Proprietary data logging tools are developed and integrated in all UEs, MECs, message brokers and cloud services. These tools will provide the CCAM application-level logging, communication logging and vehicle data as run time logging. These tools have been used in earlier projects such as AUTOPILOT [8] and CONCORDA and are extended for 5G-MOBIX.

Communication logging can be provided and uploaded to the CTS from the application layer 2 and to some extent from the network layer 1. No data from the access layer level will be provided to the CTS. Three approaches will be used by the different partners to provide logging in the common format as required in Table 25:

- The tool TCPdump is used to log at the transport layer in PCAP files. These PCAP files are converted into common communication logging in csv files.
- Vehicle data from automated vehicles is collected in ROS bags. These data are converted into vehicle and application logging in csv files.
- Proprietary scripts are used to log application and communication data directly into csv files.

4.6.3. NL Data management tools

The NL TS data repository uses a file server to collect all log data. Figure 34 shows the web page to manually upload log data, and instructions for automated uploading on <https://ada1.tno.nl/5g-mobix/data-provisioning.php>. The log data and analysis results are stored in a PostgreSQL database. Generic tools to process and analyse the data are psql, PostGIS, and R.



Driving forward Connected and Automated Mobility

Data Provisioning

The data from on-board units, IoT platforms and road side stations is provided by participants.

Data is organised in directories per participant, per use case and per session or experiment. A directory is only accessible to the participant or owner of the data. Provided data is not made accessible to others via this portal.

Providing Experiment data

Data can be uploaded after every session or experiment using the following form.

Upload logfile:

Experiment name:

Logfile: No file chosen

Data can also be uploaded using a HTTP POST with form data from a script. The following is an example BASH script to upload one file (/logging/my_logfile.csv) for one experiment (2020-10-20_Session1).

```

#!/bin/bash

URL=https://ada1.tno.nl/5g-mobix/upload/upload_file.php

USERNAME=<username>
```

Triggering Experiment Analysis

Once all data for an experiment has been provided, the automated analysis can be started.

Trigger Processing of Experiment

Trigger Processing (and Uploading to CTS) of Experiment

Based on [5G-MOBIX-CTS-Test-Data-Interface-V2.0.xsd](#)

Experiment Name of uploaded logfiles

Experiment Name:

Test Plan

Trial Site:

Use Case Category:

User Story:

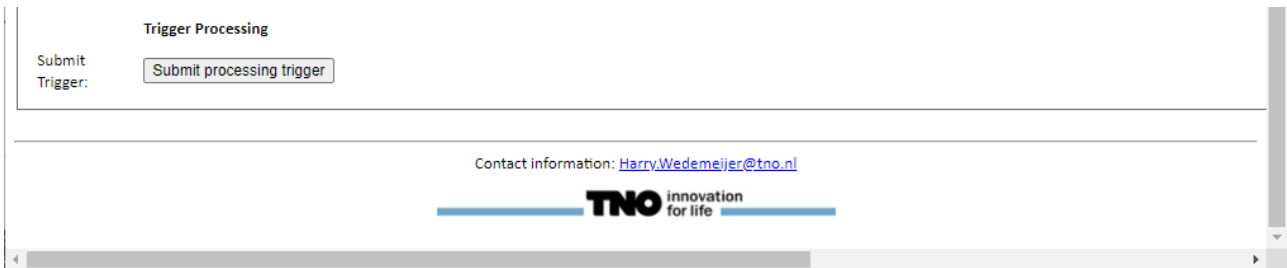


Figure 34: NL TS website to upload log data. (Top, middle and bottom sections of the web page)

4.6.4. NL Test data server architecture

Figure 35 shows the architecture for managing the data collection, analysis and management for the NLTS. The NLTS test data server consists of a file server for the log data, a PostgreSQL database server, a server for the Automated Data Analysis (ADA) and a web server to report the analysis results on the repository.

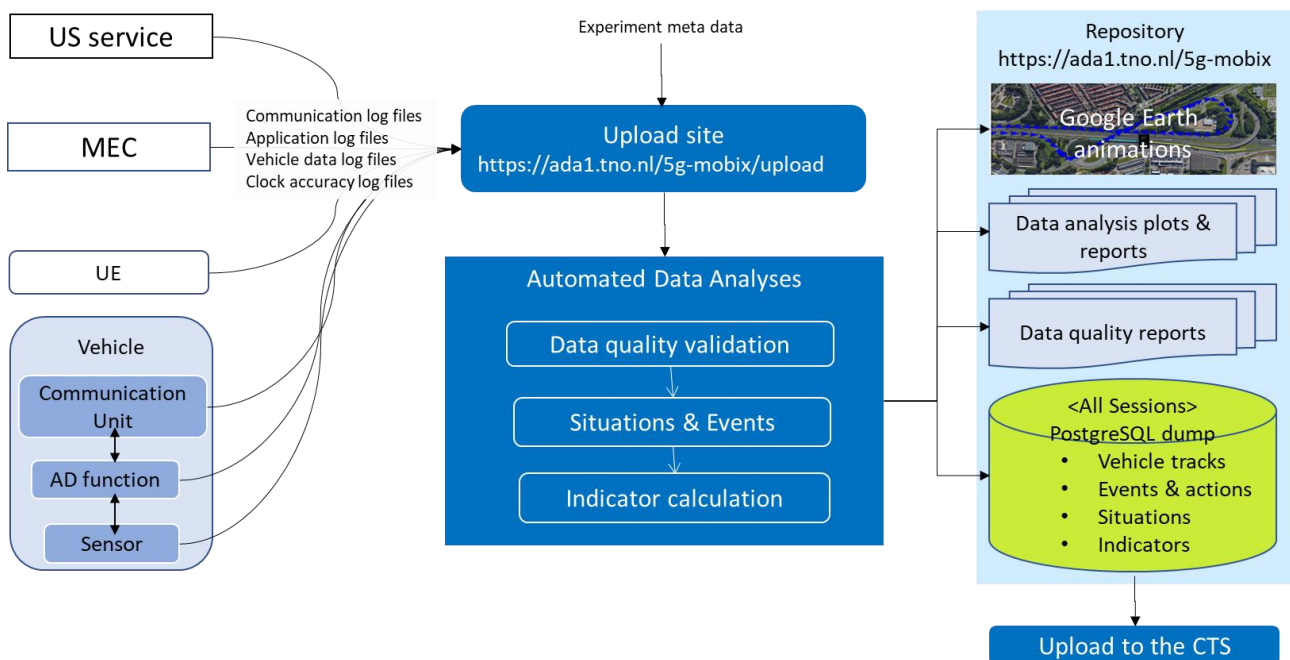


Figure 35: NLTS data management and server architecture

Every UE and vehicle unit, MEC and service for a user story log and upload the data to the upload site <https://ada1.tno.nl/5g-mobix/upload/> where the log data are stored on a file server. All log data from a single test run is uploaded to a separate folder, called an experiment. The metadata for the test run is added manually to the experiment. Once all data are uploaded, the test leader can trigger the ADA analysis from the upload site.

The output from the ADA analysis is stored in a PostgreSQL database. This includes filtered and aggregated log data, detected events and situations, and the calculated indicators. The results can be accessed in the form of a dump of the database, various visualisations, and reports on the repository web site.

The required aggregated data can be extracted from the database and exported as csv files to the CTS.

4.6.5. NL Partners involved in data collection and management

The following tables identifies the partners that collect and upload log data in the NL TS.

TNO is the data manager and responsible for managing and storing the log and analysed data, verifying and quality checking the log data, and for transferring the data to the CTS.

Table 26: NL TS data collection

Name	UC Extended Sensors	UC Remote Driving	UC CoCA
Vehicle with devices	TNO, Siemens	TUE, Siemens	VTT, TUE
MEC	TNO, KPN	KPN, Siemens	KPN
Core network	TNO, KPN	KPN, TUE	KPN
UC services	TNO, Siemens	TUE, Siemens, KPN	VTT
Remote Station		TUE, Siemens	

4.6.6. GDPR Process/Implementation

TNO is appointed as the Data Controller for the NL TS. All partners from Table 26 are data processors.

A data protection impact assessment will be executed.

The current point of view is that no personal data in the sense of the GDPR will be generated during the 5G-MOBIX tests, nor collected and stored. Only technical tests will be conducted at the NL TS; no user acceptance test or impact assessment tests will be conducted. All trials will be executed by partner employees only, such as qualified test drivers. No other users, such as from the general public, will be involved in the trials. All test equipment is owned by the partner companies, no personal devices of personnel will be used in the trials. Only test networks will be used for the 5G core networks to which external users have no access during the 5G-MOBIX trials. The test equipment, such as test vehicles and devices, will be operated by different test drivers during the 5G-MOBIX trials so no vehicle data can be correlated to any specific test driver.

4.7. Chinese (CN) Trial Site specificities

4.7.1. CN Data logging requirements

This section summarizes the specific requirements applicable to data logging at the trial site level.

Table 27: CN data logging requirements

ID	Title	Description
LRQ-CN-01	Timestamp Uniformity	For all data logging, Universal Time Coordinated (UTC) is adopted as single time reference with acceptable resolution, about 20 ms.
LRQ-CN-02	Log Verification	Local data from TS must be validated before uploading to the central cloud database. In a sample test, log data and meta data will be checked and analysed, then be uploaded to database.
LRQ-CN-03	Data Ownership	The local data processing and storage facilities should be able to ingest data from devices and services. The data will have different ownership.
LRQ-CN-04	Test Data	All data will be uploaded to the database in common log format.
LRQ-CN-05	Test Run	Test run is a single execution of a test case. Log data and meta data are uploaded to the database per test run and does not contain data of other test runs. Meta data includes a report on the executed test, including a list of participating vehicles, users, and any deviations to the planned test session such as system failures or deviations in vehicle trajectories or events.

4.7.2. CN Measurement tools

This section describes the tools selected for the measurements. Different tools can be selected to support different measurements e.g., measuring network capabilities (UCC/US agnostic) against measuring UCC/US specific logging information. Different tools can be even selected by different UCC/US within each trial site.

1) Fiddler ¹⁶

Fiddler is a web debugging proxy and packet capture tool for macOS, Windows, and Linux. It can intercept, retransmit, edit and transmit data packets sent and received through network transmission, and can also be used to detect network security. And we use fiddler to capture, inspect, monitor all HTTP(S) traffic between computers and the Internet, mock requests, and diagnose network issues.

¹⁶ <https://www.telerik.com/fiddler>

2) DiagGrabPro ¹⁷

DiagGrabPro is a tool used to grab SIMCOM communication module logs. In the CN TS, it is mainly used to capture RSU and OBU data records. The tool acts as a 5G module log capture tool in the measurement architecture.

3) WebRTC stats

Since the audio, video or data packets transmitted through the peer-to-peer connection may be lost and will experience various network delays, WebRTC stats is used to monitor the transmission performance between network components, view the current network delay, received and lost data, total number of packets, packet jitter, etc.

4.7.3. CN Data management tools

This section describes the data management tools defined to collect, process and provide evaluation data for technical evaluation.

The following are the database management tools used in the test, such as memory database Redis, relational database PostgreSQL, time series server influxDB, etc.

1) Redis

Redis is an open-source (BSD licensed), in-memory data structure store used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes with radius queries and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

2) PostgreSQL

PostgreSQL is an open-source object-relational database system. It provides two modes: one mode guarantees that if the operating system or hardware crashes, the data will be saved to the disk; the other mode does not provide a data guarantee, but it usually runs faster. It supports many modern features: complex queries, foreign keys, triggers, updatable views, transactional integrity.

3) influxDB

InfluxDB is the open-source time-series database that is part of the TICK (Telegraf, InfluxDB, Chronograf, Kapacitor) stack. It is designed to handle high write and query loads and provides a SQL-like query language called InfluxQL for interacting with data.

¹⁷ <http://www.pudn.com/Download/item/id/3970546.html>

4.7.4. CN Test data server architecture

This section contains the description and architecture of the local trial site test data server or management facility.

In CN TS, a test data server architecture is implemented for data collection, processing and storage. The edge cloud cooperates with the 5G gNB to collect data and transmit it to the central cloud. Three databases are deployed in the data centre for data storage. The first database, Redis, is an in-memory database based on the k-v mechanism, mainly used for real-time high-concurrency message queue services. The second database, InfluxDB, is a time-series database. InfluxDB is responsible for massive writes, load query tasks and stores a large amount of timestamped data. The third database, PostgreSQL, is a relational database, mainly used for persistence, addition, deletion, modification, etc.

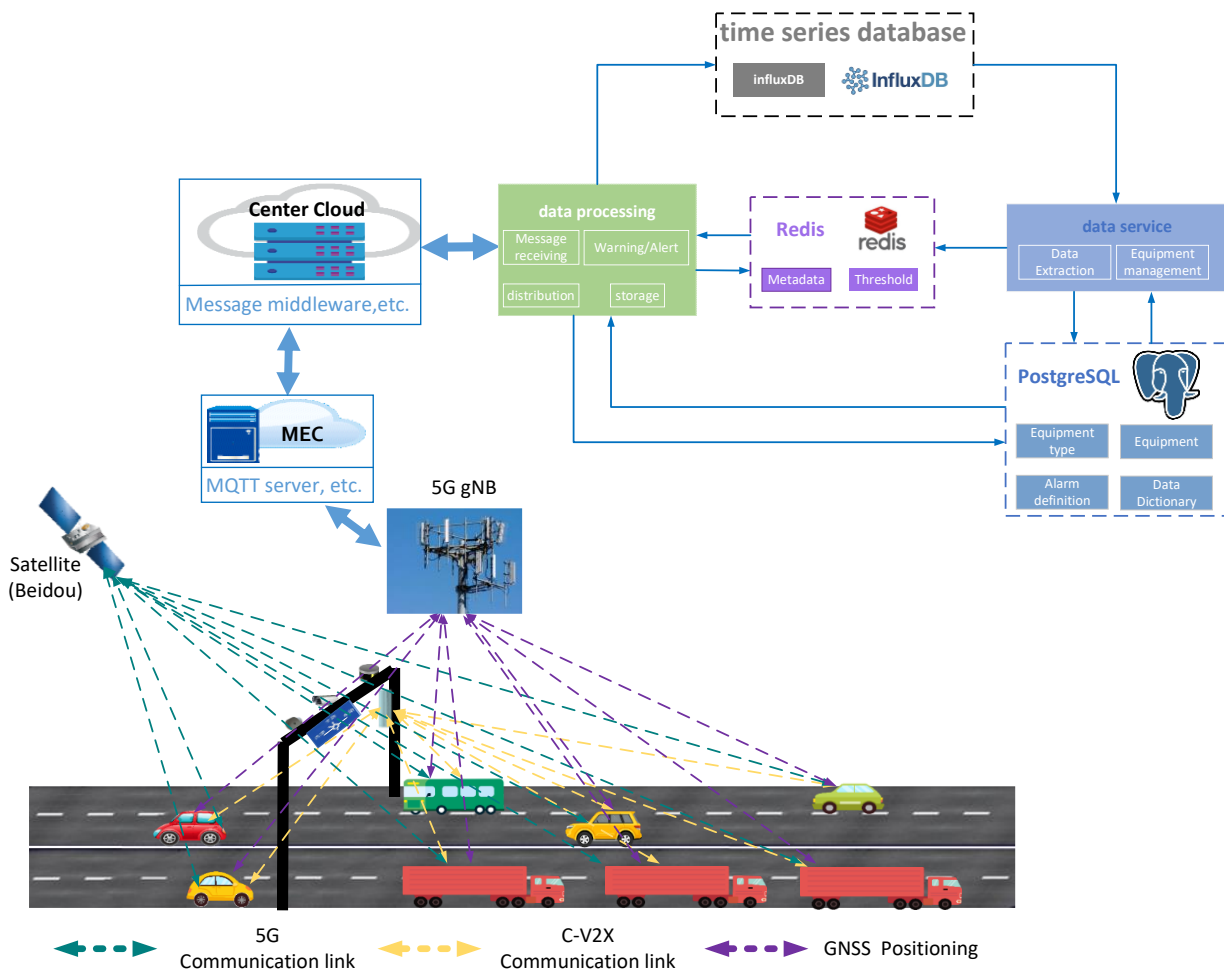


Figure 36: CN data flow and server architecture

4.7.5. CN Partners involved in data collection and management

Table 28 identifies the different persons/entities that will be responsible to carry out the following tasks:

Table 28: CN data collection responsibilities

CN data collection responsibilities	
Data manager	DUT
Data collection operation	Shandong Academy of Sciences (SDIA), Shangdong Hi-speed Company (SDHS), Dalian Dazzlee Technology Co.Ltd (DDET), China Datang Co.Ltd (DATANG), China National Heavy Duty Truck (CNHTC)
Data saving, centralisation and storage at TS level after each trial	SDIA, SDHS (QLTI)
Data quality check verification	DUT
Data transfer for evaluation to the CTS	CNHTC, SDIA

4.7.6. GDPR Process/Implementation

CN TS has developed and conducted tests for remote driving and advanced driving. The CN site has completed a 2km highway with three full-width gantries with a spacing of 500 meters and an eagle-eye camera. Mechanical and electrical equipment such as roadside LIDARs and V2X roadside units are installed with 5G, LTE-V, DSRC (Dedicated Short Range Communication), EUHT (Enhanced Ultra High Throughput), Wi-Fi, 10Gbe network and other heterogeneous networks 3D high-precision map and other vehicle-road collaborative applications. During testing, the test data from relevant vehicles and roads are extracted from eagle-eye cameras and LIDARs, etc., including location, speed, vehicle type, and weather conditions, etc. In addition, the data are transmitted to the edge cloud and then uploaded to the central cloud for further analysis and storage.

According to the data management plan (D1.6) CN will not process any data collected by European trial sites.

4.8. Korean (KR) Trial Site specificities

4.8.1. KR Data logging requirements

This section summarizes the specific requirements applicable to data logging at the trial site level.

Table 29: Summary of latency requirements

ID	Title	Description
LRQ-KR-01	data units	A ping test is measured in ms and downlink/uplink data rates are measured in Mbps.
LRQ-KR-02	data validity	An App installed in a Wi-Fi device checks connection to an App

		server. If the server cannot be reached by any reason, measured ping and data rates are not valid.
LRQ-KR-03	time reference	Universal Time Coordinated (UTC) is proposed as the time reference for data logging for the tethering via a vehicle and remote driving use case.
LRQ-KR-04	test run	Test run can be single or multiple execution of a test case. Since a test run is associated with a car, route and location, the test run must include these data simultaneously.

4.8.2. KR Measurement tools

For the KR TS different tools are chosen in order to produce the metrics needed to calculate the network capabilities and the KPIs. Benchbee¹⁸ is a tool designed to test mobile network speed. You can measure network performance such as ping, downlink data rate and uplink data rate. 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 in 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, UE, Wi-Fi AP and the end-user device.

4.8.3. KR Data management tools

KR TS will use the remote server to store all the measurement data with an appropriate file structure in remote control use case. The remote server is developed by KATECH based on visual studio (Microsoft) and measurement studio (National Instrument). Only KR 5G MOBIX partners have permission to access remote server database. The GUI for visualisation of the test data will also be developed and provided by KATECH based on visual studio and measurement studio. No management tools are used in the tethering use case.

4.8.4. KR Test data server architecture

KR TS uses simplified test data server architecture. All the test data of the test vehicle will be transmitted to the remote server via UDP protocol. The remote server stores all the collected data in the database through MySQL. No data server is used in the tethering use case.

4.8.5. KR Partners involved in data collection and management

Initial specification of Test Scenarios and planning Test Sessions for each of the two user stories are visible in Table 30.

¹⁸ <https://www.benchbee.co.kr>

Table 30: KR data collection responsibilities

KR data collection responsibilities	
Tethering	ETRI
Remote Driving	KATECH
Logging of test data	KATECH, ETRI
Upload Test data to Remote Server	KATECH, ETRI
Completion of test reporting	KATECH, ETRI
Data quality check	KATECH, ETRI

4.8.6. GDPR Process / Implementation

All the use cases of the KR TS are developed and tested in the private proving ground in KATECH. No personal data will be generated, collected and stored during the tests for all use cases since only technical tests will be executed by KR partners. Especially, all the test equipment in the test vehicle and remote server will only be used in the trials. All the test data will be communicated via private mmWAVE-based 5G network with secured communication link.

5. CENTRALISED TEST SERVER DATA MANAGEMENT

The Centralised Test Server (CTS) is a centralised platform which aims to collect and store the data from the trial sites and to allow the evaluators to browse, access and download data. All the data stored in the CTS has previously been anonymised by the data providers. The CTS will ease the identification of research data that will be shared under Open Research Data Pilot (ORDP [1]).

5.1. CTS data management requirements

This section presents the design rationale, i.e., it describes and justifies the main design decisions. The requirements and constraints define the expectations about the architecture. The design decisions are guided by the use cases of the CTS:

- Uploading data from trial sites;
- Browsing, fetching, querying, searching among data;
- Storage of various type of data;
- Monitoring upload tasks;
- User and profiles management;
- Downloading data manually (WEB API);
- Downloading data automatically (REST API).

Requirements in Table 31 are the most important non-functional requirements the CTS will comply in order to ensure an optimal interoperability with Trial Sites and integration in the 5G-MOBIX project.

Table 31: CTS non-functional requirements

ID	Title	Description
NFRQ-CTS-01	Data transfer	The CTS must provide an interface to upload and store test data. HTTPS is the preferred solution.
NFRQ-CTS-02	Data transfer	The CTS must provide an alternative SFTP interface to upload and store test data.
NFRQ-CTS-03	Data storage	The CTS must provide enough storage space for all test data from all trial sites.
NFRQ-CTS-04	Data browsing	The CTS must provide a web interface to search and browse uploaded test data.
NFRQ-CTS-05	Data upload monitoring	The CTS should provide a monitoring system that will enable the data uploading process.
NFRQ-CTS-06	REST API	The CTS must provide a REST API that gives access to search,

		browse, upload and download of test data.
NFRQ-CTS-07	Data validation	CTS must check and validate that uploaded test data complies with the predefined test description interface (test data XSD interface definition).
NFRQ-CTS-08	Data quality	CTS must check that test data complies with data quality requirements for evaluation.
NFRQ-CTS-09	Data	CTS will receive the test data in a zip or tar file that must be unzipped to retrieve the metadata information.
NFRQ-CTS-10	Data accessibility	CTS must provide access to all the evaluators.
NFRQ-CTS-11	Data access	CTS must provide an authentication and authorisation access to stored data.
NFRQ-CTS-12	Data processing	Test data must be stored without any transformation or conversion.
NFRQ-CTS-13	Data storage	Test data must be stored locally in the CTS environment.
NFRQ-CTS-14	Data storage	A PostgreSQL database is used to store the content of the description file associated to the test data.

5.2. CTS functional description

The CTS offers various APIs (with or without HMI) for trial sites to upload their data. It will allow data owners but also any other users connected to the CTS to follow the uploading progress and to see the final upload status of the data. The CTS provides an HMI to the evaluators in order to easily browse through the available data, search for specific data using metadata information, and download selected data.

Figure 37 represents the main use cases (UML description) developed within the CTS. In the schema, the Test Data Builder application is shown as the source providing test data to the CTS. Three categories of users can access the system (Data provider, Admin and Evaluator). Their capabilities to perform a specific operation are shown by the arrows

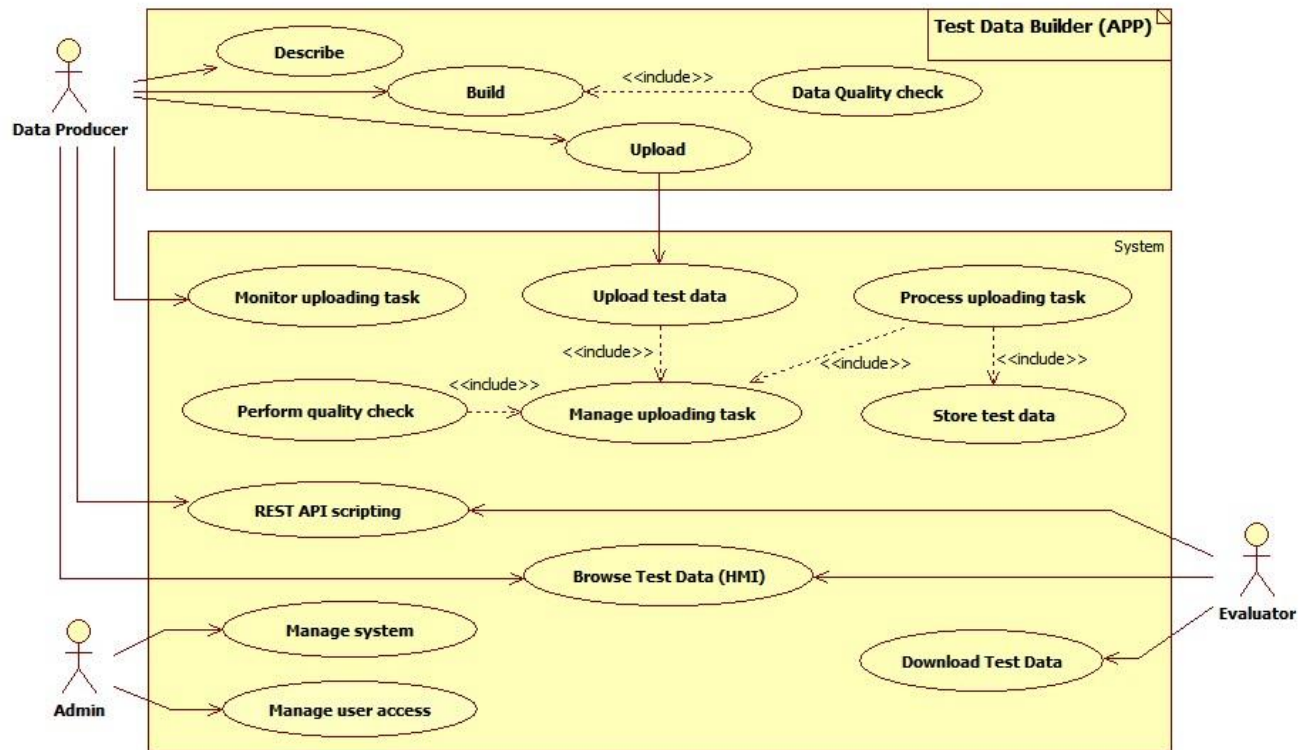


Figure 37: Centralised Test Server and Builder application use cases

From the main use cases description, a schematic view of the components of the CTS has been defined. Figure 38 shows the main functional components, those with API and the internal ones.



Figure 38: Centralised Test Server functional description

Upload test data collects data sent by trial sites and sends it to the storage component.

Test data are transferred as a package containing the data themselves, along with metadata (data description). Data are stored in a specific filesystem, and metadata are stored in a description database, also containing the path to the corresponding dataset in the filesystem. These metadata are used by the data browser to fetch available data using any criteria contained in metadata.

This component provides the functionality to upload data and unzip the archives. It checks the completeness of the description file and its consistency with the uploaded files. Then files are handed over to the metadata processing component.

Manage metadata provides the functionality to process and store metadata information describing the test data files. These metadata are stored in the description database.

Store test data provides the functionality to create repositories for each upload, to store files in their dedicated repository, and to store the original uploaded archive in a backup repository. Information about the path to retrieve data (repository) is added in the description database.

Browse test data offers an interface for browsing available test data, using various criteria (user story, trial site, date, etc....), retrieve and download test data.

Download test data builds an archive according to the user data request and makes it available for download. It can provide one or several test data archives at once, or an individual data file selected inside a test data archive.

Task manager, monitor upload will provide a way to monitor the upload progress and status, and to check the status of previous operations.

Quality check performs metadata quality check and produces a report on a test data archive.

5.3. CTS system deployment

The core of the CTS is an application server running the back-end (CTS main application), and exposing the front-end (CTS web interface). The CTS also has a REST API in order to facilitate the automation of the upload and download processes.

The system architecture, including interfaces of services and functionalities, is shown in Figure 39. It is mainly composed as follows:

- Application server: hosts the front-end (web interface and REST API), and the back-end (CTS main application);
- File system server: stores the test data files;
- Database server: stores the test data description;
- SFTP server: receives test data uploaded by trial sites.

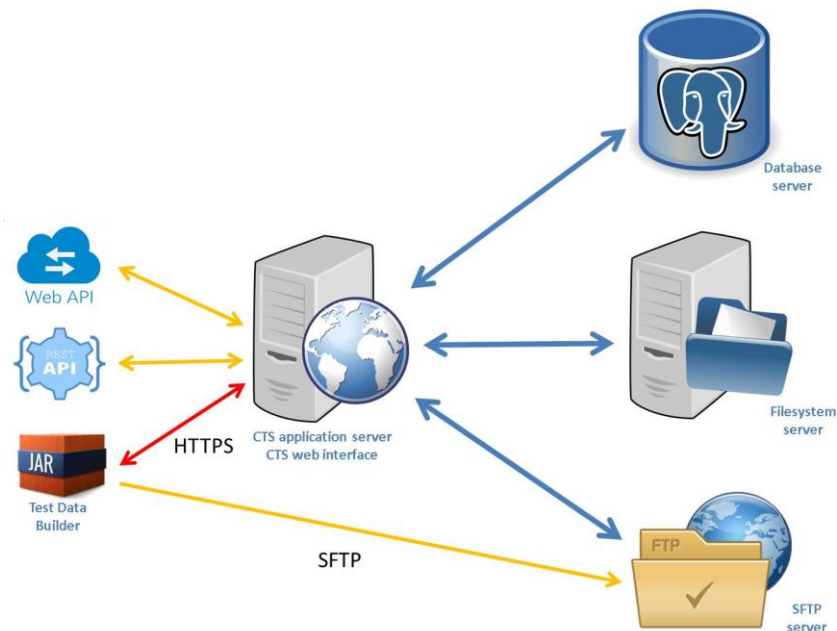


Figure 39: Centralised Test Server system architecture

Web API and REST API are used to browse and download test data from CTS. The Test Data Builder is used to upload test data. The red link represents the authentication process, the uploading task creation and the uploading information acquisition. Once uploading information is retrieved, the Test Data Builder connects and uploads test data files to the HTTPS or the SFTP server. The Python scripts behave the same way.

The Centralised Test Server is deployed in the Microsoft Azure cloud. It is composed of several virtual machines, each dedicated to a specific task. The main server is accessible across an AKKA reverse proxy, ensuring a high level of security.

5.4. CTS interfaces and access to data

There are several ways to list, browse, search and **download** test data files: web interface, REST API, Swagger. Those interfaces and the authentication process used to access them are described in the following sections.

There are two ways to **upload** test data archives to the CTS, the Test Data Builder application or the Python script. In each case, two methods are provided: SFTP, and HTTPS.

Whichever upload method is used, first step is to authenticate at the CTS using user credential. A unique identifier (UUID) is then generated and transmitted to the application. This time limited validity UUID is used for the HTTPS communication between application and server. HTTPS communications and commands trigger the CTS's REST API.

SFTP is a secured files transfer protocol over SSH. In our case, we use application internal credentials for data transfer.

5.4.1. Web interface

The CTS can be accessed at the following address — currently a test, demonstration and integration-version: <https://cts.westeurope.cloudapp.azure.com>.

CTS production's server and its URL will be shared with each TS.

CTS website allows to browse and download uploaded test data, read data details and quality check status, and follow data uploading and processing status.

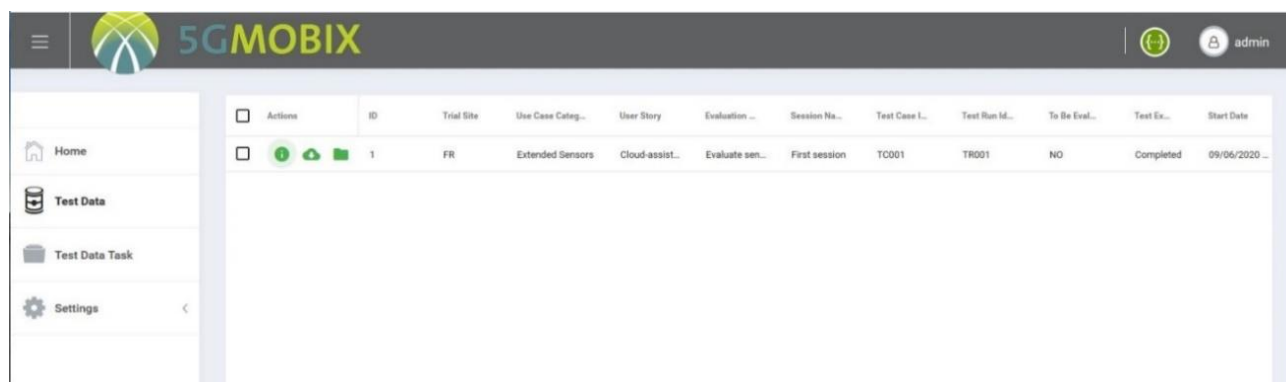
Users need credentials to login. In order to get access to the CTS, users must request by email the CTS administrator. The administrator creates the login name and the CTS generates and sends automatically the credentials to the user.

Following figures present parts of the CTS website pages. Figure 40 presents the login page of the CTS. The authentication process between the CTS front-end displayed in user's web browser and CTS back-end running on the cloud servers is described in section 5.3.

The image shows the login page of the 5GMOBIX CTS. At the top, there is the 5GMOBIX logo, which consists of a stylized green and blue circular icon followed by the text "5GMOBIX" in green and blue. Below the logo, the text "Sign in with your username" is displayed. There are two input fields: the first one contains the text "admin", and the second one contains a series of dots representing a password. Below the password field, there is a link that says "Forgot Password?". At the bottom, there is a green button with the text "SIGN IN" in white capital letters.

Figure 40: CTS login

Figure 41 presents a part of the CTS main page displaying the list of test data currently uploaded and stored in the CTS. From this page, it is possible to download whole test data archives, display the list of log files contained in an archive and download them separately, and display the details about an archive, as depicted in Figure 42.






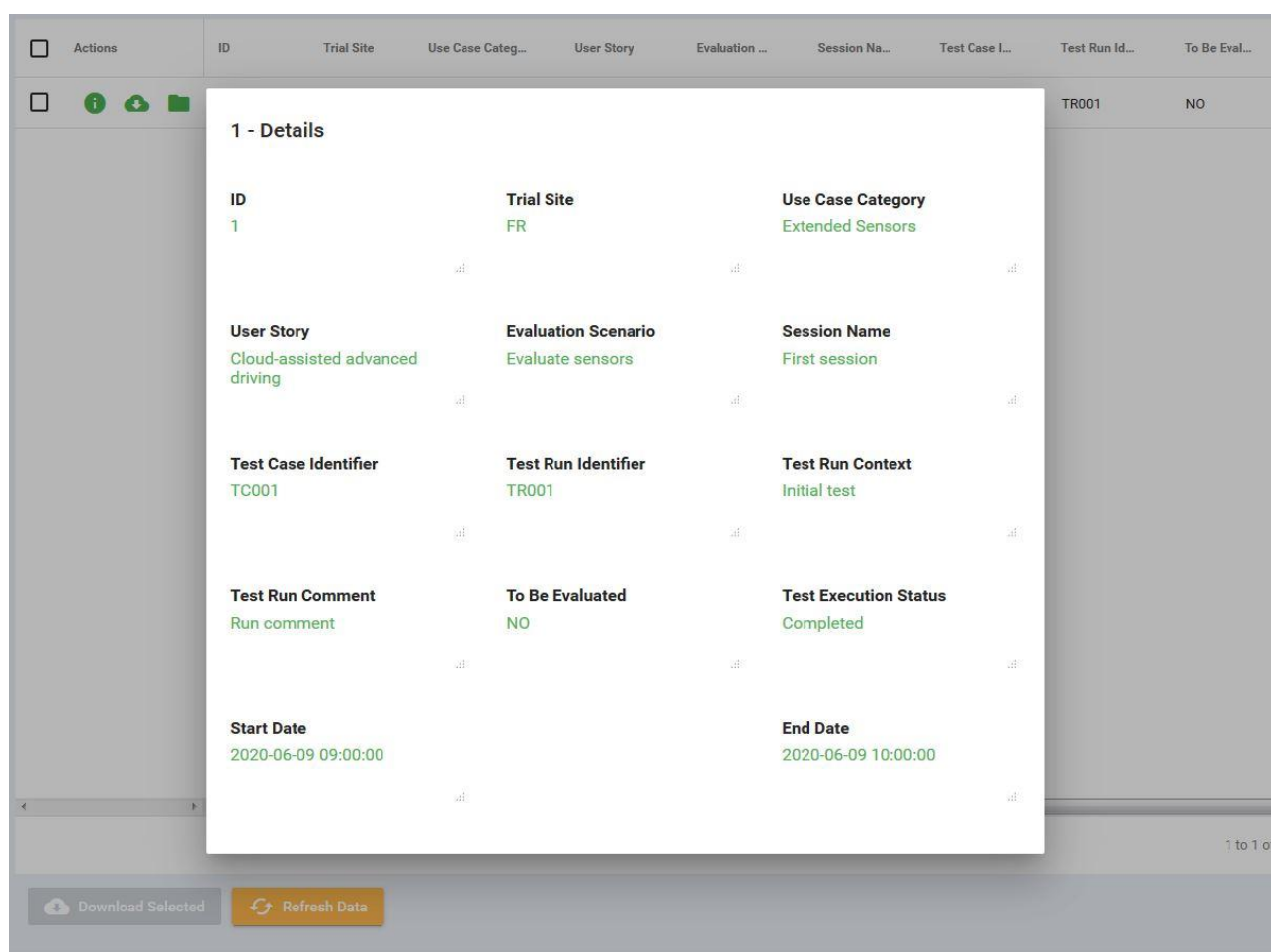
Actions	ID	Trial Site	Use Case Category	User Story	Evaluation	Session Name	Test Case Identifier	Test Run Identifier	To Be Evaluated	Test Execution Status	Start Date
  	1	FR	Extended Sensors	Cloud-assisted advanced driving	Evaluate sensors	First session	TC001	TR001	NO	Completed	09/06/2020

Figure 41: CTS test data list



1 - Details		
ID 1	Trial Site FR	Use Case Category Extended Sensors
User Story Cloud-assisted advanced driving	Evaluation Scenario Evaluate sensors	Session Name First session
Test Case Identifier TC001	Test Run Identifier TR001	Test Run Context Initial test
Test Run Comment Run comment	To Be Evaluated NO	Test Execution Status Completed
Start Date 2020-06-09 09:00:00	End Date 2020-06-09 10:00:00	

Figure 42: CTS test data details

5.4.2. REST API

The CTS RESTful API makes it possible to give access to all the public resources available in the CTS. This interface is based on a REST API standard and is documented by and can be discovered through the Swagger¹⁹ provided tools (See Section 5.4.3).

The CTS RESTful API is used by the CTS web front-end to communicate with the CTS back-end, but also by the Test Data Builder application and the Python scripts. Thus, through this API, one can browse and search data, and download or upload test data archives: data providers and evaluators can use this API in their own data management scripts.

Python scripts (see Section 5.5.2) and Test Data Builder (see Section 5.5.1) all use this API.

5.4.3. Swagger

Figure 43 screenshot shows the Swagger page accessible from the CTS website. This is a user friendly way of using directly the CTS RESTful API across a web page automatically built by Swagger¹⁸.

Using this interface, one can execute all the REST commands interfacing to CTS, and also, if needed, get the corresponding complete curl²⁰ command for executing it later in command line or in a script.

¹⁹ Swagger is used for API development, following OpenAPI Specification, the industry standard for RESTful API design. See <https://swagger.io/>

²⁰ Curl is a command line tool and library for transferring data with URLs. See <https://curl.se/>

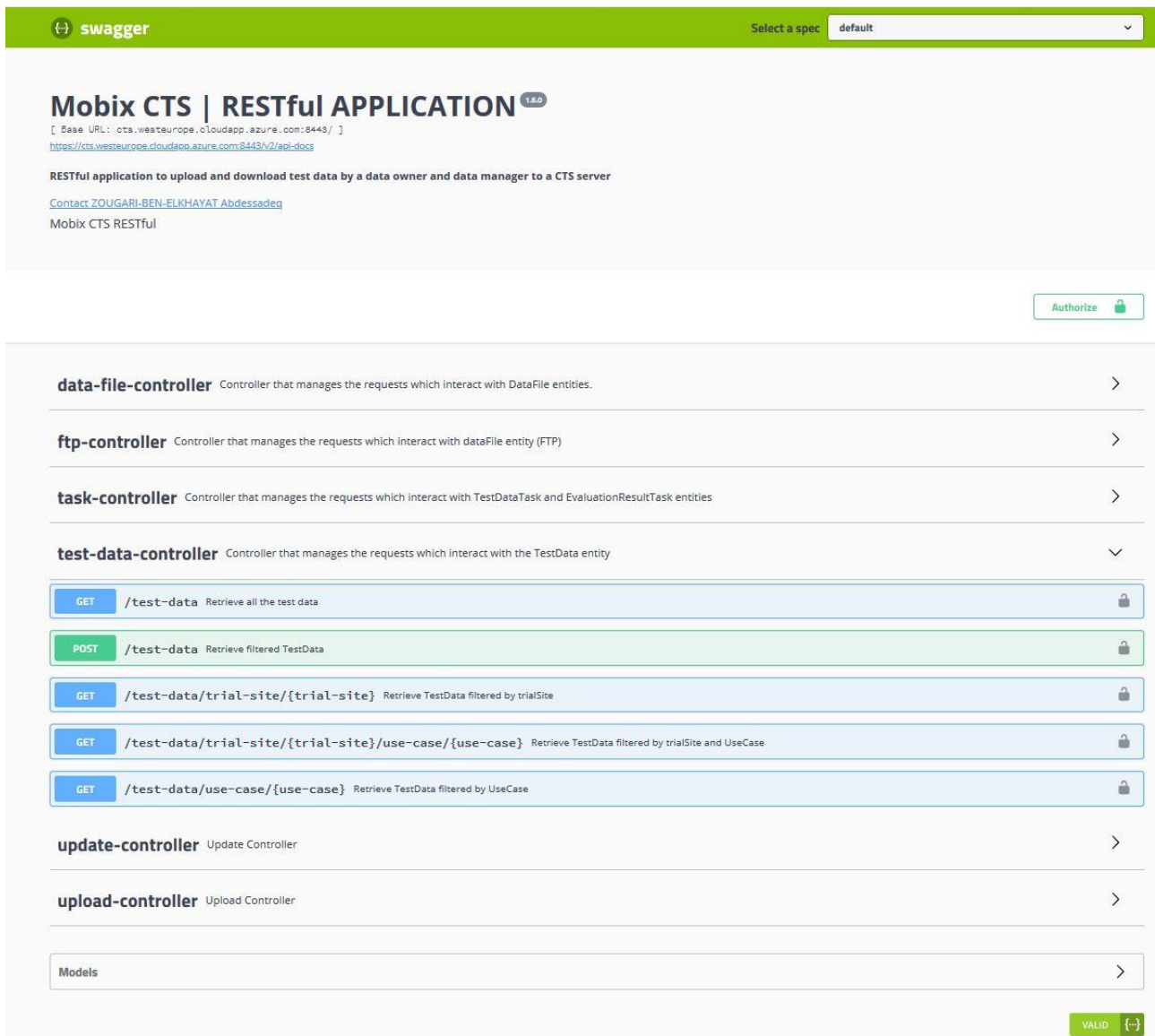


Figure 43: Swagger REST interface

5.4.3.1. Authentication and authorization

The REST API is secured with JSON WEB TOKEN (JWT)²¹. This concept allows users to enter their username and password to obtain a token to access a specific resource for a specific period.

²¹ JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties. See <https://jwt.io/>

The role of the server is to determine whether a user should or should not be authenticated, to return a response to the user to begin the authentication process. This authentication in our case is taken in this form: entering a login/password in a form encoded in base 64 as it is shown in Figure 44 and Figure 45.

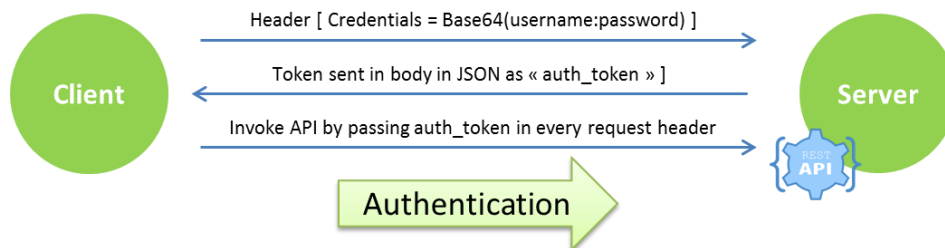


Figure 44: CTS authentication procedure

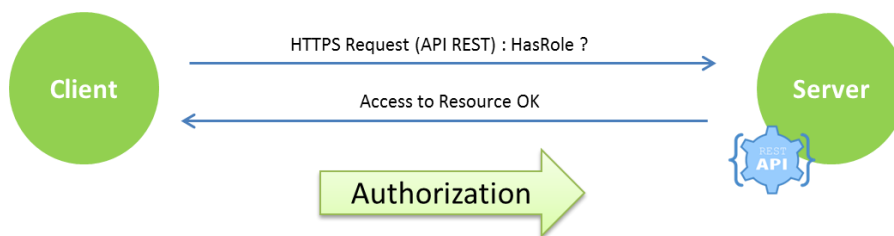


Figure 45: CTS authorization procedure

5.5. CTS tools

The upload and download processes may involve uploading or downloading a large number of files. Therefore, AKKA provides complementary tools for transferring data: the Test Data Builder, a Java application for building test data archives²² and uploading them to the CTS, the `upload_testdata_to_cts.py` and `download_td_from_cts.py` Python scripts for uploading or downloading test data archives to/from the CTS.

5.5.1. Test data builder application

This standalone Java application aims to help data managers to create test data archives with accurate and complete description and to upload these archives to the CTS. The main user interface form is mainly focused on collecting test data description, then the user can add as many data files as needed, each one with its description.

²² A "test data archive" is an archive (*i.e.* a zip file or a tar file) containing the log files (*i.e.* "data") of a test session, a description file and some complementary files.

The login page allows the user to set the configuration of the application, and to log using an authentication process, allowing only authorized users to upload data to the CTS.

Figure 46 presents the login page of the Test Data Builder application. The authentication process between the Test Data Builder and the CTS described in section 5.4.3.1.

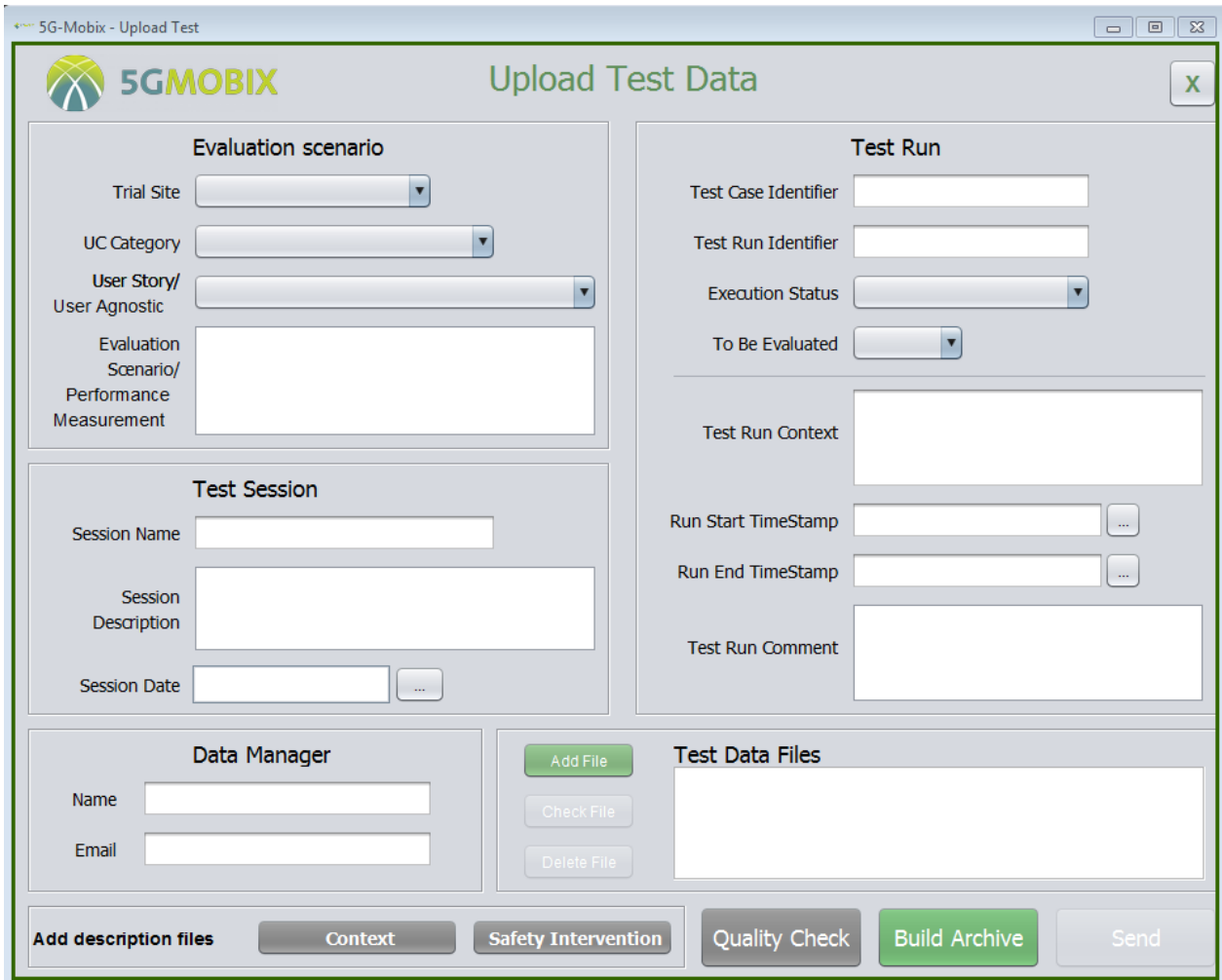


Figure 46: Test data builder secured login

From the login page it is possible to set the properties for using the Test Data Builder. The properties file is automatically generated by the application if not present, and can also be edited by hand See Annex 2,

Test data archive building, storage and upload:

The Test Data Builder main page Figure 47 allows filling information required for a complete description of the test run. This page presents also the list of log files currently added, and allows the user to check the quality of these files, either for one specific file, or for the complete archive at once.



The screenshot shows a web application window titled "5G-Mobix - Upload Test". The main content area is titled "Upload Test Data" and contains several sections for data entry:

- Evaluation scenario:** Includes dropdown menus for "Trial Site", "UC Category", and "User Story/ User Agnostic", and a text area for "Evaluation Scenario/ Performance Measurement".
- Test Session:** Includes text input fields for "Session Name", "Session Description", and "Session Date" (with a calendar icon).
- Test Run:** Includes text input fields for "Test Case Identifier", "Test Run Identifier", "Execution Status" (dropdown), "To Be Evaluated" (dropdown), "Test Run Context", "Run Start TimeStamp" (with a calendar icon), "Run End TimeStamp" (with a calendar icon), and "Test Run Comment".
- Data Manager:** Includes text input fields for "Name" and "Email".
- Test Data Files:** Includes buttons for "Add File", "Check File", and "Delete File", and a large text area for file details.

At the bottom of the form, there is a row of buttons: "Add description files", "Context", "Safety Intervention", "Quality Check", "Build Archive", and "Send".

Figure 47: Test data builder main form

The add file page, Figure 48, is used to add log files and fill required information to describe the log recording context and conditions, along with any comment useful for log analysis. A quality check button allows the user to verify that the log is compliant with expected quality requirements.

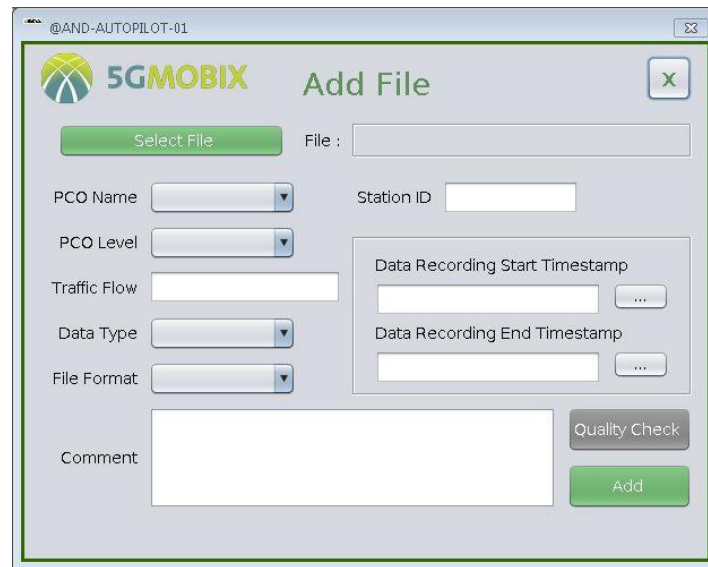


Figure 48: Test data builder adding test data file

Once ready, the user can generate the test data archive, which is stored locally. Then it is possible to upload it with one click to the CTS. The description file is automatically generated and DQCT is called by the Test Data Builder, therefore, a report on data quality is joined to the archive. The upload is done using HTTPS or SFTP, according to parameters setting.

Test data archive reading, validating and upload:

Test Data Builder can also be used to only upload to the CTS existing test data archives from local storage. These archives can have been created by the application, or with other means. In this case, the Test Data Builder first checks the archives. Only well-formed archives, compliant with requirements and with correct description are accepted and uploaded.

5.5.2. Python scripts

Test data archive upload

In order to offer a way to upload test data archives without using the GUI, a Python script has been developed, "**upload_testdata_to_cts.py**". This script provides only the upload service, therefore the description file inside the archive and the archive itself must be built before.

This script offers two uploading modes, as do the applications, either by SFTP or by HTTPS. It must be run in a command line terminal, on the server containing the archive to upload.

Note that Python 2.7 must be installed on the server running the upload script.

Example of use for uploading test data:

```
upload_testdata_to_cts.py -ip <CTS> -u <username> -p <password> -f /path_to/<test_data_archive> [-http]
```

The script needs credentials to check authorizations, thus, username and password, as for the java upload applications, are the ones used to connect to the CTS web-server.

Test data archive download

For the same reason as upload script, following the same schema, another script has been developed to ease evaluators archive downloading, "**download_td_from_cts.py**". This script allows downloading at once, all test data filtered (or not) by trial site and user story. Any test data previously downloaded is not downloaded again, and archives are unzipped for convenient and immediate use. For configuration, see Annex 2.

Extract of usage:

```
download_td_from_cts.py [-ps TRIALSITE] [-uc USECASE] [-ev EVAL] [-f]
```

5.6. Test data archive structure

Log files are built from raw data collected during trials, using a common format agreed by all CBC/TS. Once common format logs are ready, the trial site data manager must group all the logs from a test execution in a test data archive, and upload this archive to the CTS.

Each test data archive contains all the logs of a single test case execution ("Test Run"), a description file, two additional files, a test case run context description file and a safety intervention file, and the data quality check report (Figure 49).

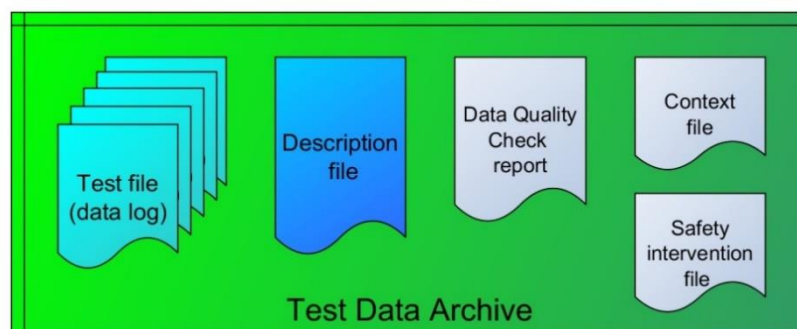


Figure 49: Test data archive content

The description file contains metadata related to the Trial Site, Use Case, User Story, Test Run, etc., and details about each log file (Timestamp, Level, PCO, etc.). The format of this file, defined in a XSD file, has been created by AKKA and agreed by and shared with partners. This XML description file and the data

quality check report are automatically generated and added to the archive when using Test Data Builder tool (see Section 5.5.1).

5.7. GDPR Process/Implementation

In the context of the 5G-MOBIX project, evaluation data produced during trialling activities will be collected and stored in the CTS and used for the technical evaluation of the project by work package 5. Each trial site is in charge of collecting, processing and managing its data in the local trial site data management system and the role of each partner is described in section 4.0.5 and in D1.6.

Regarding the CTS, ERTICO is DPO and AKKA is both data controller and data processor in charge of the collection and data sharing of evaluation data.

According to the objectives of the task 3.5, the collected data are adequate, relevant and limited to the purpose for which they are processed, namely the computation of key performance indicators defined in D5.1.

Data access to CTS is restricted to a list of users among TS partners who are identified by the following information: username, full name, password, email and role (ADMINISTRATOR, DATA PROVIDER, and EVALUATOR). AKKA is in charge of the creation of the user list provided by all TS partners. User's information is only used to allow the connection to the CTS. No additional processing is done using these data. Data subject needs to send an email to CTS admin if it does not need to access the CTS anymore. The list of the CTS users is not shared or transferred to any other partners of the project. Each data record stored at CTS can be traced by using the data provider responsible for the data aggregation of the data to be provided for the evaluation.

Regarding website security, the CTS implements an authentication and authorisation protocol, deployed in https and a backup process is applied.

At the end of the project and when the evaluation is completed, a subset of collected data will be published as Open Research Data by the data owner in a public repository and all data stored in the CTS will be deleted and the partner's user list removed.

A DPIA for the CTS has been performed and privacy and data protection risks analysed. Indeed, illegitimate access to the data, unwanted modification of the data and data disappearance risk's severity and likelihood are negligible or limited.

6. CONCLUSION

The work presented here has been elaborated within Task 3.5 of WP₃ of 5G-MOBIX, in close cooperation with WP₅. The main objective of the D_{3.5} is to define the methodology and deliver the tools to be used in 5G-MOBIX at centralized level by all trial sites and at distributed levels by each trial site.

The data collection methodology was created in both top-down and bottom-up approaches. In the bottom-up approach, pilot sites described what their capabilities and constraints were and in the top-down approach all mandatory requirements related to data measurements to be logged, data format, data quality check, time synchronisation, data storage and data sharing were defined, discussed and finally resulted in the definition of a common data management methodology.

This deliverable defines the distributed data management and the centralised data management. The distributed data management shows how trial sites are organised to manage data collection for evaluation in terms of architecture, data quality, data collection and data integration. The centralised data management describes how the collected data will be stored in a centralised environment, shared and made available to the evaluators.

This deliverable describes the measurement tools selected by each trial site. Different tools are selected to support different measurements measuring network capabilities (UCC/US agnostic) against measuring UCC/US specific logging measurements based on the CCAM application level observations reported by the CBC/TS. Proprietary data logging tools are also developed and integrated in all PCOs.

Finally, 5G-MOBIX proposes a unique platform called the Centralised Test Server (CTS) to upload, share, store and browse the data, allowing the evaluators to work with harmonized data. The CTS unifies the way the test data will be transferred and guarantees that all mandatory metadata will be provided in order to identify precisely any shared data created during trialling activities. The core of the CTS is an application server running the back-end (CTS main application), and exposing the front-end (CTS web interface). The CTS also has a REST API in order to facilitate the automation of the upload and download processes. The resulting Centralised Test Server platform and associated tools for transferring data enable an optimised management of test data collected during 5G-MOBIX test sessions and ease the identification of research data that will be shared under Open Research Data Pilot (ORDP [1]).

REFERENCES

- [1] OpenAIRE, "ORDP," [Online]. Available: <https://www.openaire.eu/what-is-the-open-research-data-pilot>.
- [2] EC, "Data management," Horizon 2020, [Online]. Available: https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-data-management/data-management_en.htm.
- [3] Zenodo. [Online]. Available: <https://zenodo.org/communities/5g-mobix>.
- [4] ntp.org, "How accurate will my Clock be?," [Online]. Available: <http://www.ntp.org/ntpfaq/NTP-s-algo.htm#Q-ACCURATE-CLOCK>.
- [5] H. Xu, "Application of GPS-RTK Technology in the Land Change," *International Workshop on Information and Electronics Engineering (IWIEE)*, pp. 3454-3459, 2012.
- [6] "Common log data formats from InterCor, C-Mobile and AUTOPILOT," TNO, [Online]. Available: <https://ada1.tno.nl/logformats/>.
- [7] "Common Format and MIME Type for Comma-Separated Values (CSV) File - RFC4180," [Online]. Available: <https://tools.ietf.org/html/rfc4180>.
- [8] EC, "AUTOMated driving Progressed by Internet Of Things," [Online]. Available: <https://cordis.europa.eu/project/id/731993>.
- [9] S. Vanneste, J. de Hoog, T. Huybrechts, S. Bosmans, R. Eyckerman, M. Sharif, S. Mercelis and P. Hellinckx, "Distributed uniform streaming framework: an elastic fog computing platform for event stream processing and platform transparency," *MDPI Future Internet*, vol. 11, no. 7, p. 158, 2019.
- [10] T. & K. C. & S. M. & B. C. Sturm, A Taxonomy of Optimization Factors for Platooning, *IEEE Transactions on Intelligent Transportation Systems*. PP. 1-18. 10.1109/TITS.2020.2994537, 2020.
- [11] J. G. A. a. B. G. Y. J. Liang, Analysis of packet loss for compressed video: effect of burst losses and correlation between error frames, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 861-874, 2008.

ANNEXES

Annex 1. Test data description file

This description is organised in an .xml file and filled by the data provider or use case responsible. This allows the evaluators to have more information about the test data. Table 32 describes the content of the test data description. All parameters marked with (*) are mandatory.

Table 32: Test data description file content

Test data description content			Description
Test Data Properties	Trial Site *		Name of the Trial Site
	Use Case Category*		Name of the use case category tested
	User Story *		User story tested or agnostic test
	Evaluation Scenario *		General description of the test in the evaluation perspective
	Session Name *		Name of the test session
	Session Description		Description of the test session
	Session Date *		Date of the test session
	TS Data Manager *	Name	Name of the data manager
		Email	Email of the data manager
	Test Run	Test Case Identifier *	Reference of the test case
		Test Run Identifier *	Identifier of the test run in the test case
		Execution Status *	The test status (complete, fail ...)
		To Be Evaluated *	Yes, if this test run is good for evaluation
		Test Run Context	Description of the test context. This contains the test conditions, the weather conditions... Any information helpful for evaluation.
		Start Time *	Starting time of the test in UTC reference time
		End Time *	Ending time of the test in UTC reference time
		Test Run Comment	Any relevant comment about the test run that could be useful to enrich the metadata or the content

		DQCT Status *	Data quality status of the test data archive
		DQCT Report *	Report generated by DQCT on the test data archive
Data Files	File Name *	Name of the test data file. It should be relevant to the file content	
	PCO Name *	Point of Control and Observation name	
	PCO Level *	Level of the communication layer measurement data source	
	Traffic Flow *	Name of the traffic flow	
	Type *	Reference of the data type (CAM, DENM, IVI...)	
	Format *	The data format used to store data	
	Log Station ID	Identifier of the logging device	
	Comment	Comment that could be useful to enrich the metadata or the content	
	File Time *	The time of the file generation in UTC reference time	

Annex 2. CTS configuration files

The Test Data Builder properties file is generated by the application if not present, can be edited using application GUI, and can also be edited by hand. It is structured as follows (address of the CTS is an example):

```
# Target upload server configuration
ip=cts.westeurope.cloudapp.azure.com
port=8443
http_upload=true
storage=/home/user/storage
dqctConfig=/home/user/dqct/formats
# If needed proxy configuration and credentials can be defined using following properties
# in order to reach the upload server across your proxy.
#proxyHost=
#proxyPort=
#proxyUser=
#proxyPassword=
```

Listing 3: Properties file of Test Data Builder

The data download Python script configuration file (config_download.ini) is structured as follows. Username and password are the credentials used to connect to the CTS.

```
[DEFAULT]
username =
password =
directory = /home/user/DOWNLOAD
```

Listing 4: Configuration file of test data download script