Grant Agreement Number: 731993

Project acronym: AUTOPILOT

Project full title: AUTOmated driving Progressed by Internet Of Things

D3.6

DATA COLLECTION AND INTEGRATION METHODOLOGY

Due delivery date: 31/03/2018

Actual delivery date: 30/03/2018

Organisation name of lead participant for this deliverable: AKKA

Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the GSA)	
RE	Restricted to a group specified by the consortium (including the GSA)	
CO	Confidential , only for members of the consortium (including the GSA)	

Project funded by the European Union's Horizon 2020 Research and Innovation Programme (2014 – 2020)

AUTOPILOT

Document Control Sheet

Deliverable number:	3.6
Deliverable responsible:	AKKA & VED
Work package:	3
Editor:	Abdessadeq ZOUGARI, Anne-Charlotte NICOUD

Author(s) – in alphabetical order		
Name	**Organisation**	**E-mail**
AITAAZIZI Amine	AKKA	Amine.ait-aazizi@akka.eu
ZOUGARI Sadeq	AKKA	Sadeq.zougari@akka.eu
KAHALE Elie	AKKA	Elie.KAHALE@akka.eu
DALET Benoit	AKKA	Benoit.dalet@akka.eu
PIERI Lorenzo	AVR	lorenzo.pieri@avrgroup.it
KALOGIROU Kostas	CERTH	kalogir@certh.gr
FALCITELLI Mariano	CNIT	mariano.falcitelli@cnit.it
PAGANO Paolo	CNIT	paolo.pagano@cnit.it
SIMEON Jean-François	CONTI	jean-francois.simeon@continental-corporation.com
YEUNG Michel	CONTI	michel.yeung@continental-corporation.com
GALLI Mauro	CRF	mauro.galli@crf.it
VISINTAINER Filippo	CRF	Filippo.visintainer@crf.it
MALDONADO Manuel	CTAG	manuel.maldonado@ctag.com
RIAL Moisés	CTAG	moises.rial@ctag.com
VELIZHEV Alexander	IBM RE	ave@zurich.ibm.com
PONT Jordi	IDI	Jordi.Pont@idiada.com
HERAS Ruben	IDI	Ruben.heras@idiada.com
SCALAMBRO Paolo	TIM	paolo.scalambro@telecomitalia.it
NETTEN Bart	TNO	Bart.netten@tno.nl
WEDEMEIJER Harry	TNO	harry.wedemeijer@tno.nl
CHEN Haibo	UNL	h.chen@its.leeds.ac.uk
BARNARD Yvonne	UNL	Y.Barnard@leeds.ac.uk
NICOUD Anne-Charlotte	VEDECOM	Anne-Charlotte.nicoud@vedecom.fr
SCHREINER Floriane	VEDECOM	Floriane.schreiner@vedecom.fr
SCHOLLIERS Johan	VTT	Johan.Scholliers@vtt.fi
KOSKINEN Sami	VTT	sami.koskinen@vtt.fi

Document Revision History			
Version	**Date**	**Modifications Introduced**	
		Modification Reason	**Modified by**
V0.1	30/06/2017	ToC	CHEN, Haibo
V0.5	31/08/2017	First Draft	CHEN, Haibo
V0.7	26/09/2017	Draft with responsibilities	CHEN, Haibo
V0.15	15/01/2018	Draft version (minor adjustments to ensure the quality of the deliverable)	SCHREINER, Floriane
V0.16	19/01/2018	Add 4.4.2 section Urban driving for Versailles pilot site	KALOGIROU, Kostas
V0.28	01/03/2018	Restructuring of the document	SCHREINER, Floriane

V0.32	07/03/2018	Inputs from partners after the telco	All
V0.32-35	16/03/2018	Final improvements of the deliverable	AITAAZIZI, Amine SCHREINER, Floriane
V1.0	19/03/2018	Version for peer review	AITAAZIZI, Amine SCHREINER, Floriane ZOUGARI, Abdessadeq
V2.0	30/03/2018	Version submitted to the EC	ZOUGARI, Abdessadeq

Abstract

This document describes the data collection and integration methodology together with a summary of the state of the art on tools needed for data acquisition, transmission, database structure, quality assurance, data storage.

The developed methodologies are highly impacted by the WP3 pilot site test specifications, the WP4 evaluation requirements and the FESTA methodology that is applied in AUTOPILOT.

Evaluators have developed spreadsheets in task 4.1 describing the required data for evaluation and the research questions that AUTOPILOT needs to answer. Pilot sites have provided their test specifications for the deployment of each use case.

The data collection methodology defines a common process that should be used by all the PS and for each use case to produce required data with the right quality to the evaluation process.

The deliverable presents the raw data to be collected with focus on the WP4 requirements and specificities per pilot sites. In addition, the document explains the data management methodologies and procedures in pilot sites, and defines the features of the centralised data management in AUTOPILOT which enables the storage and sharing of collected test data and produced evaluation results.

Legal Disclaimer

Abbreviations and Acronyms

Acronym	Definition
ADAS	Advanced Driver-Assistance Systems
AVP	Automated Valet Parking
CAM	Cooperative Awareness Basic Service Message
CAN	Controller Area Network
DENM	Decentralised Environmental Notification Basic Service Message
DMP	Data Management Plan
EC	European Commission
FAIR	Findable – Accessible – Interoperable-Reusable
FESTA	Field opErational teSt supporT Action
GA	Grant Agreement
GPS	Global Positioning System
INS	Inertial Navigation System
IoT	Internet of Things
ITS	Intelligent Transportation Systems
LIDAR	Light Detection And Ranging
LIN	Local Interconnected Network
MOST	Media Oriented System Transport
NDA	Non-Disclosure agreement
NMEA	National Marine Electronics Association
OBU	On Board Unit
OEM	Original Equipment Manufacturer
ORDP	Open Research Data Pilot
PO	Project officer
PS	Pilot site
RADAR	Radio Detection And Ranging
RPM	Revolutions Per Minute
RSU	Road Side Unit
SPSS	Statistical Package for the Social Sciences
UTC	Coordinated Universal Time
VRU	Vulnerable Road Users
WGS-84	World Geodetic System 1984
WP	Work Package

Table of contents

List of figures

List of tables

AUTOPILOT

Executive summary

This deliverable D3.6 “Data collection and integration methodology” is a public document and provides a complete picture of the data management methodology developed in AUTOPILOT, in particular for the data collection and integration. Task 3.4 has developed a common methodology for the local data management platform and the centralised on, which are used especially for data collection and D3.6 is the first deliverable of this task.

D3.6 gives a first global overview of the data to be collected from the pilot site (PS) tests with a focus on the specifications of these collected raw data: description, type and format, frequency, data quality etc. The raw data can be divided into two main groups: the vehicle data and the IoT. Both are needed for evaluation purposes and are described in chapter 2.

Each PS has developed a local data management methodology based on the common methodology. In other words, the PSs have all specifications on their local level regarding the methodology used for data collection, processing, quality and integration and these are presented in this deliverable.

On top of the local data management platform, there is a centralised data management server. All the principles and processes that will be applied in this centralised server are explained in chapter 5 of this document. More specifically, it gives a deep understanding of the interfaces provided to the evaluators and to the pilot sites in order to centralise both test data and evaluation results.

Finally, the data sharing methodology describes the data and metadata description recommendations to facilitate the understanding of the context in which the data was collected.

Commented [JB1]: On what?

Commented [JB2]: Added to Abbreviations and Acronyms

1 Introduction

1.1 Purpose of the document

This document represents the first deliverable of Task 3.4 - Test Data Management. Its main purpose is to ensure that, prior to the field tests (Task 3.3), all pilot sites have identified necessary tools for collecting and managing test data, and (wherever possible) to ensure comparability and consistency of collected data across pilot sites. Task 3.4 mainly draws input from T3.1 "Pilot tests specifications» and T4.1 "Evaluation methodology". Moreover, the task 3.4 outputs serves as an input for the task 4.2.

This deliverable (D3.6) presents a methodology and the tools for collecting and managing test data to be collected at the pilot sites as well as processing and integrating data needed for the impact evaluation in WP4. It builds upon the relevant results from existing field operational tests and automation projects to create an inventory of these tools for the whole procedure of pilot site tests including data acquisition, transmission, database structure, quality assurance, and data storage. D3.6 provides the requirements of the tools to be developed and implemented by the pilot sites.

It is important to note that the D3.6 is the first deliverable of the task 3.4. It will serve as an input for the next T3.4 deliverables.

D3.7 **"Test data management architecture platform"** describes the architecture of the Pilot Sites Test Servers and of the Centralised Test Server. This includes the software specifications and the technical details of their implementation.

D3.8 **"Implementation of test data management platform"** refers to the real implementation of the architectures defined in the D3.8.

D3.9 **"Test data"** is about providing report on piloting activities related to data collection process, data provisioning and collected test data during test sessions done at local pilot sites.

The D3.6 is structured in six chapters.

Chapter 1 **"Introduction"**

Chapter 2 **"Data Management Methodology"** presents the methodologies and processes applied for test data management across pilot sites. It includes the steps managed by Task 3.4 to come up with building the common methodology for local and centralised data management, in particular for data collection.

Chapter 3 **"Raw Data"** first provides a global overview of the data to be collected from the pilot site tests with a specific focus on the specifications of the collected data: description, type and format, frequency, data quality etc. The chapter starts with a global description of the data categories (vehicle data, IoT data) needed for the evaluation, and then describes the specificities found on each pilot site.

Chapter 4 **"Data Collection & Integration – distributed data management"** features all the pilot sites requirements of their local data management. Based on the spreadsheets filled by the pilot sites, they developed their data management depending on their previous facilities. In the chapter, every pilot site explains the methodologies used for data collection, data processing, data quality and data integration.

Chapter 5 **"Centralised Data Management"** explains all the principles and processes that will be applied in the centralised data management server. More specifically, the chapter gives a deep understanding of the interfaces provided to the evaluators and pilot sites in order to centralise test data.

Chapter 6 **"Data Sharing"** presents the methodologies and procedures applied by the pilot sites to share their data. Pilot sites are following the common approach described in the chapter.

This deliverable includes Annexes. The first one identifies an inventory of tools used for data management across pilot sites. The second and third appendix references the data collection templates filled by pilot sites. The fourth appendix references resources concerning the data sharing process.

1.2 Intended audience

This deliverable (D3.6) is a public document and therefore, the intended audience for this document is considered to be anyone that is interested in large-scale collection and management of test data which are highly diverse and spatially-and-temporally varying. Therefore, this document also covers tools needed for data acquisition, transmission, database design, quality assurance and data storage.

Within the AUTOPILOT project, the main intended audience for this deliverable is considered to be all the AUTOPILOT participants and in particular, the AUTOPILOT participants involved in the field tests at the pilot sites, as well as those involved in impact assessment (WP4).

2 Data management methodology in AUTOPILOT

This first deliverable of T3.4 presents the data collection and integration methodologies that will be applied in the AUTOPILOT project.

For the data collection, the AUTOPILOT methodology is based on a two-pronged approach. First, the deployed methodology needs to be compliant with the WP4 evaluation directives. In order to meet these requirements, evaluators have developed a spreadsheet in task 4.1 to describe the required data for evaluation and the research questions that AUTOPILOT needs to answer. The evaluation methodology is explained in the D4.1 deliverable[1].

In parallel, pilot sites provided their initial specifications in the D3.1 deliverable[2]. The aim was to collect from all the pilot sites the existing and envisaged specifications for the deployment of each use case. The specifications handle all the pilot sites aspects: vehicles, infrastructures, IoT devices, IoT platforms, standards, data and test scenarios.

Consequently, the data collection and integration methodologies are completely impacted by these requirements.

One of the main objectives within T3.4 "Test Data management" is to ensure the comparability and consistency of collected data across pilot sites. The task defines the distributed data management of site specific test data from the local pilot sites and central data management. To reach these objectives we defined a methodology described hereafter.

The distributed data management methodology started with the development of data management spreadsheets. These templates helped pilot sites describing their data collection and management process by answering a list of questions related to these topics: architecture, collection, quality, storage, pre-processing, metadata, archiving and preservation, personal data, database, data storage, data uploading and ethics. Once the template was filled, each pilot site presented during a telco their data collection and data management system. During a Q&A session, we discussed how this process can be adapted to provide the expected data needed for the evaluation.

After this first phase, each pilot site describes the data collection and integration methodology they will deploy locally in their Distributed Data Management Systems in chapter 4.

Concerning the data requirement, WP4 presented in several workshops their deliverables, highlights and their needs both in terms of data to be collected with their properties (quality, accuracy, frequency, synchronisation means, baselines, etc.), but also in terms of where the data should be collected and by which system. Consequently, each pilot site has filled a datasheet and presented it during telcos or face to face meetings associating WP3 and WP4 partners.

Concerning the data provisioning to the centralised server, T3.4 defined a methodology that covers data sharing and provisioning. Indeed, all test data collected and pre-processed at the local level will be made available to the centralised server through agreed interfaces described in chapter 5.

This interface identifies precisely any provided test data to ease the access and analysis of all the pilot sites data, and to enable easier comparison of AD/IoT functions across sites. The centralised server enables direct access to the evaluator. Concerning data provisioning and access to evaluation result, T3.4 will adopt the same process as the one defined for test data.

[1] [D4.1]Methodology for evaluation, version 1.0, 31 January 2018

[2] [D3.1]Initial pilot test specifications, version 2.0, 31 December 2017

Some pilot sites will collect some confidential data. As a result, the central storage will not include these data and the concerned partners will provide the result of the analysis.

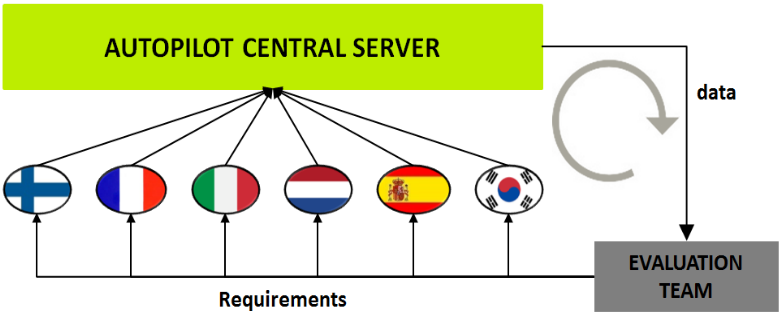


Figure 1 Data management methodology in AUTOPILOT

Concerning data sharing, The AUTOPILOT is compliant to the ORDP. In chapter 6, there are some recommendations that will be used to improve the data sharing methodologies during and after the project completion. These recommendations are more detailed in D6.7 “Data Management Plan” which describes the data management process for the AUTOPILOT project globally.

3 Raw data

Chapter 3 describes which data will be collected on each pilot site within AUTOPILOT project. More specifically, this chapter provides an overview of the data to be collected from the pilot site tests with a specific focus on the specifications of the collected data (e.g. types, structures and formats) which are commonly available at the sites; and the analysis of the quality, comparability and consistency of the collected data across the pilot tests.

3.1 Requirements for evaluation

This first section presents the basic data requirements from WP4, i.e. the minimum set of data which has to be collected by each PS. Based on a large spreadsheet developed by WP4, task 3.4 has built another, simplified, excel sheet listing the data required by WP4 for evaluation. The template of this sheet can be found in annex 2. The vehicles sources and data, IoT sources and data, survey data and tools as well as metadata standards will be described here.

Then in section 3.2, the specificities per pilot site will be presented. In particular, the available raw data on each pilot site.

After this, section 3.3 is going to refer to annex 3 where the excel sheet, mentioned above and filled in by each PS, will be shown. These tables give a first clear picture of the data available on each PS.

3.1.1 Vehicle sources and data

This section defines the common data format for vehicle data. Vehicle data provides information on the vehicle itself and its motion.

The FESTA matrix, a spreadsheet in Excel format containing three tables: *"Performance indicators"*, *"Measures"* and *"Sensors"*, has been used to create a spreadsheet with the data requirements. The spreadsheet has been defined in order to **list all the data requirements and to ensure that all the PSs are aligned with them**. In the spreadsheet all the sources of data are mentioned. In this section we focus on the vehicle sources and vehicle data.

In order to ensure a correct evaluation of the new functionalities implemented within AUTOPILOT it is important to collect most of the data generated by the vehicles and their sensors. Therefore, the list of sensors which might be available on the vehicles will be described first and, after that, the list of data extracted from them.

3.1.1.1 Vehicle sources

The spreadsheet in the appendix defines for each sensor some information to classify it referring to the columns in the tables:

- **Sensor group:** Group of sensors to indicate what type of information this sensor primarily provides, e.g. vehicle dynamics sensors
- **Sensor class:** Class of sensors used to do a kind of measurement, e.g. acceleration sensors
- **Sensor type**: The specific type of sensor, e.g. lateral acceleration sensor
- **Sensor key:** A unique ID to identify the sensor
- **Measures:** Measurable parameters obtained from the sensor

Vehicle

This includes the in-vehicle data which will be collected by the vehicle-bus (CAN/LIN/MOST/FlexRay). The vehicle bus is considered a data source with many possible outputs. For each test scenario, the capabilities of the intended vehicle should be investigated. From this source we get data such as:

- Vehicle speed
- External temperature
- State of the battery load
- Range estimated with the actual battery load state
- Fuel consumption
- Odometer data

Positioning system

This includes the GPS devices which the vehicle has. It can be a normal GPS, a differential GPS or an Inertial Navigation System (INS). From the GPS (or the differential GPS) we will obtain the standard consumer global satellite positioning system, with an increase of accuracy using reference stations on the ground in the case of the differential GPS. The INS, used in combination with the GPS, will provide information to “fill the blanks” at GPS dropouts. From this source we will get NMEA sentences which will include data such as:

- Speed
- Longitude and Longitude error
- Latitude and Latitude error
- Quality of GPS signal
- Heading
- Number of satellites in use

Vehicle dynamics

This group of sensors includes the ones involved with the dynamics of the vehicle. The yaw rate is the measurement with which the sensor obtains the angular velocity of the vehicle rotation, or rate of change of the heading angle when the vehicle is horizontal. The acceleration sensor provides the two types of acceleration, i.e., the lateral and the longitudinal. Finally, the speed sensor provides the speed measured with a sensor on free running wheel for an increased accuracy. From this source we will obtain:

- Yaw rate
- Lateral acceleration
- Longitudinal acceleration
- Speed (pulse sensor/counter at the wheels)

Driver/Vehicle interaction control

This group of sensors includes all the elements which allow the driver to interact with the vehicle, such as the pedals, the wipers or the steering wheel. The information we can obtain from this source is:

- Pedals status
- Brake force
- Wipers status
- Steering wheel position

Environment sensors

The environmental sensors are the ones that observe the environment of the vehicle and collect information from it. This includes the camera, which provides environment video from all the perspectives (forward, rearward, left and right), the radar, which also provides data from all the perspectives constantly sensing the distance between the vehicle in real-time and the LIDAR which provides the environment measurements from a laser scanner. From these sources we obtain following data:

- Video view
- Radar view
- LIDAR view

3.1.1.2 Vehicle data

From all the sources mentioned before, we expect a **list of data to be used by the evaluation**. In the spreadsheet this data is classified and also detailed in several columns. These columns give us information such as:

- **KeyPI:** Unique ID to identify the performance indicator
- **Parameter**: Name of the performance indicator
- **Measures:** Required measures to compute the value of the parameter, e.g. Speed_CAN and Speed_GPS and Speed_WheelUnitDistance can be used to compute the mean speed
- **Sensors**: Devices used to do the measurements

N.B. Additional information as the description of the parameter, the standard unit for measurement, the frequency or accuracy requirement for the measures can be found in the excel sheet developed by WP4 on which the one used in D3.6 relies. The WP4 sheet also contains the information on whether a parameter is mandatory or optional for a use case.

Next, we describe a list of vehicle data, as it is shown in the spreadsheet, in a common data format:

- **Speed:** The speed of the vehicle in m/s. This measure can be provided through different sources such as the vehicle-bus, the GPS or the wheel sensor. The frequency should be ideally 10Hz.
- **Acceleration:** The acceleration of the vehicle in m/s^2. This measure can be provided through the acceleration sensors (lateral or longitudinal acceleration). The frequency should be ideally 10Hz.
- **Yaw rate:** The vehicle's angular velocity around its vertical axis in degrees/second. This measure can be provided by the yaw-rate sensor. The frequency should ideally be 10Hz.
- **Brake force:** The brake force is the measure of braking power of the vehicle in g. This measure can be provided by the vehicle-bus or the driver/vehicle interaction group of sensors.
- **State of charge:** The state of charge of battery of vehicle in %. This measure can be provided by the vehicle-bus. The frequency should be ideally 1Hz.
- **Position and localisation data:** The position data of the vehicle in NMEA format. This measure could be provided by the GPS or the differential GPS of the vehicle. There are many sentences in the NMEA standard with different kind of information in each one. The ones mandatory for the AUTOPILOT Project will be the GGA (with the fix information), the GSA (with the overall satellite data) and the RMC (with the recommended minimum data for the GPS. More details of this format can be found in the NMEA webpage. The frequency should be ideally 10Hz.
- **Camera Log:** The log output of the camera of the vehicle could be: the entire test recorded by the camera or only recording when an event has been triggered. The format of the video file or the image should be any standard media format (.jpg, .png, .avi or .mpeg being the most commons.).
- **Radar Log:** The format of the radar output is open to each PS and device and should be indicated among the information when uploaded to the Central Data Server.
- **LIDAR Log:** The format of the LIDAR output is open to each PS and device and should be indicated among the information when uploaded to the Central Data Server.

Synchronisation
All the information mentioned before must be synchronised to establish consistency among data from all the sources and devices, and other PS measurements in order to ensure data consistency within each of the PS.

In order to achieve it, we should have a global timestamp for each PS and each time a message is sent and received, the device should log the timestamp when received and the timestamp when delivered.

The format of the timestamp must be: YYYY-MM-DDThh: mm:ss.sTZD (e.g. 1997-07-16T19:20:30.45+01:00)

YYYY	= four-digit year
MM	= two-digit month (01=January, etc.)
DD	= two-digit day of month (01 through 31)
hh	= two digits of hour (00 through 23) (am/pm NOT allowed)
mm	= two digits of minute (00 through 59)
ss	= two digits of second (00 through 59)
s	= one or more digits representing a decimal fraction of a second
TZD	= time zone designator (Z or +hh:mm or -hh:mm)

Frequency of logging
The frequency of logging shall be modified depending on the use case. We need higher frequency of logging on the use case Highway Pilot because it's a scenario done at a higher velocity, while we could need less frequency of logging on the Automated Valet Parking use case as it is done at lower velocity.

3.1.2 IoT sources and data

This section defines all the details related to the IoT data in the project. First, it will describe the IoT sources and devices available in pilot sites with special attention to the IoT platforms involved in pilot sites. Then, the section presents the common data model that will be use and implemented across pilot sites.

3.1.2.1 IoT sources & platforms

The AUTOPILOT IoT architecture builds on, and borrows building blocks from, relevant IoT architectures such as AIOTI[3] and IoT-ARM[4]. This architecture aims to provide a global IoT service coverage through features such as **openness**, **flexibility**, **interoperability** between IoT platforms, leveraging of **standards** for communication and interfacing, and **federation** of in-vehicle, road-side unit, and pilot site IoT platforms.

AUTOPILOT IoT sources include various IoT devices and sensors:

- **IoT vehicle devices**
- **Smartphones**
- **VRUs**

[3] Alliance for Internet of Things Innovation (AIOTI): https://aioti.eu
[4] ARM-IoT: https://developer.arm.com/products/architecture/system-architecture

- **RSUs**
- **Cameras**
- **Traffic Lights**
- **Drones**
- **Specific IoT devices: DATEX**

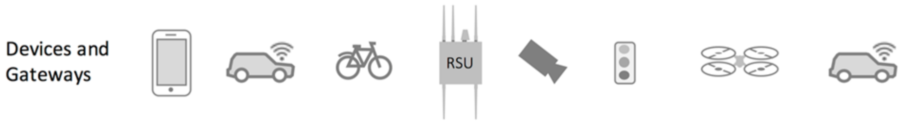


Figure 2: List of IoT devices in AUTOPILOT

Given the fact that AUTOPILOT has several large-scale pilot sites, the architectural components of the open IoT platform (infrastructure, IoT devices, services, etc.) are inherently physically distributed. AD functions themselves have varying requirements in terms of speed of access (latency), availability, and range (covered area). While some localised mission critical functions, such as warning other vehicles in the immediate proximity that a pedestrian is jaywalking, need to be accessible within very low latency. Other functions, such as notification about a parking spot being made available, need to cover wider areas but are less demanding in terms of latency. As a result, the AUTOPILOT IoT platform was designed and implemented as a federation of IoT platforms.

AUTOPILOT implements two types of IoT platforms across pilot sites:

- **Proprietary IoT Platforms:** These are used by some applications and use cases to exchange specific data with specific devices or vehicles. For example, the Brainport car sharing service and automated valet parking service use Watson IoT Platform™ to collect data from their vehicles. Several proprietary IoT platforms are used in AUTOPILOT for various purposes, use cases and pilot sites.

- **OneM2M Interoperability Platform:** This is the central IoT platform for exchanging IoT messages relevant to all autonomous driving (AD) vehicles. The proprietary IoT platforms are networked through the OneM2M interoperability platform and are connected to this through OneM2M interworking gateways. The interworking gateway of a given proprietary IoT platform may be configured to share selected data types with the interoperability platform. Such data will then become accessible to all the connected IoT platforms through the OneM2M interoperability platform. This is particularly useful for sharing data relevant to all the AD vehicles and applications, such as detected hazards, vulnerable road users, objects, etc.

3.1.2.2 IoT data

From all the IoT sources mentioned before, we expect to have a list of IoT data that will be generated as an output of the tests that will be executed in each pilot site.

AUTOPILOT launched a Data Modelling Activity Group (DMAG) that standardises all the IoT data across all pilot sites and pilot sites. Rather, the scope of the DMAG work covers only the IoT data used for exchanging **information** or **instructions** between IoT devices, services, and the AD vehicles. This includes, for example, messages notifying AD vehicles about the presence of a hazard, or object, or instructions for AD vehicle to avoid a given road lane.

Raw sensor data (example LiDAR, camera images, etc.) and service internal data models (e.g. parking data, user accounts) are beyond the scope of the data modelling activity and will be defined at each local pilot site.

The following figure provides an overview of the current AUOPILOT IoT data model packages and their dependencies. The AUTOPILOT common IoT data model is split into several packages, based on different standards (e.g., SENSORIS, DATEX II).

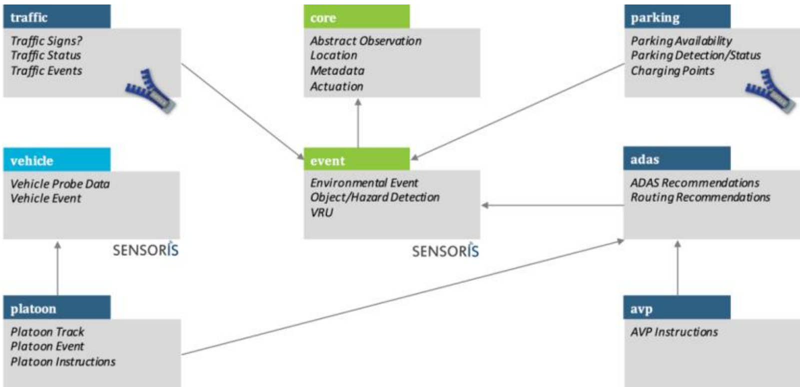


Figure 3: Common IoT data model

The data models will be specified after the initial feedback from all partners and they will be implemented. The data models are supposed to help easier development of interworking gateways which will make automatic translations from one interface to another (e.g., from OneM2M to Watson IoT). The data model should include metadata fields required for logging purposes.

The DMAG is still working on standardising g the IoT messages for the AUTOPILOT project. More details will be provided in the coming versions of the IR2.3[5].

3.1.3 Survey data and tools

Surveys are used to collect data during a specific task. Data collection is mandatory to create various types of analysis and reveal the research outcomes. Figures and charts may also illustrate a better view of the collective results.

3.1.3.1 Survey tools

CERTH/HIT uses the following survey tools for research data collection and analysis depending on the specific field of requirements. Each of the survey tools is described below with the features supports.

Google Forms[6]

Google Forms is a free software tool and a Google account is required for the user to create it. By Google Forms, surveys can be created in any language and sent out via email. It is also possible these surveys to be posted online through Google Drive. Any survey may have an unlimited number of questions and can be sent to an unlimited number of email addresses. A survey can be styled by any

[5] [IR2.3] Report on the implementation of the IoT Platform, version 1.1, 28 February 2018

[6] https://www.google.com/forms/about/

photo, colour or can be styled by using a series of template (themes). The types of questions are supported are the following:

- Free text; short answer with few lines to paragraph
- Multiple choice
- Checkboxes
- Dropdown
- Linear scale
- Multiple choice grid
- Checkbox grid

The responses to the surveys are automatically collected in forms, with real time response information and charts. Survey responses can be exported in .csv format. Finally, Google forms can be created/edited in collaboration similar to Google docs.

Google Forms does not offer the ability to post your question on Facebook like Survey Monkey, but you can post a link to your survey on Facebook, Google+ or Twitter directly from the forms web page.

Google Forms exports results in .csv format. The results can be viewed in excel or similar datasheet application.

SoSci Survey[7]
The software package SoSci Survey creates online questionnaires, sends invitation emails (including reminders) and downloads data (to SPSS, GNU R or Excel). For scientific purposes with no commercial background, the appropriate use of SoSci Survey is free – fully functional. Questionnaire length is unlimited and the standard packages allow up to 5,000 respondents (completed questionnaires). SoSci Survey offers the following features:

- Numerous question types (e.g. selection, balloon test, cloze test, ranking, scale, slider)
- Graphical and numerical scale anchors
- Integration of pictures, audio and video files
- Item-rotation and randomising
- SSL-secured data transmission for surveys
- Support with the questionnaire pre-test
- Access control with password, IP-address or with personalised links, including email distribution.
- Real-time response rate measurement, data acquisition to SPSS and GNU R (including labels from variables and values)
- Completely online-based software (no installation necessary)
- Ad-free (except free use for training purposes and non-academic private polls)

Additionally, there are some more features supported only for advanced users and professionals. A list of these features is:

- Freely programmable filters
- Freely programmable layout
- Implementation of HTML
- Dynamic content

[7] https://www.soscisurvey.de

- Uniform distributed randomisation
- Support for panel surveys
- Support for longitudinal survey designs
- Experience sampling with text message service
- Multi-language-surveys
- Support for international languages/characters (Japanese, Korean...)
- Valid HTML

SoSci Survey provides results in .csv and .sav format. The .csv files can be viewed in excel or similar datasheet application and the .sav can be opened by SPSS[8] application.

Survey Monkey[9]
SurveyMonkey is online survey development cloud-based software. It provides free, customisable surveys, as well as a suite of paid back-end programs that include data analysis, sample selection, bias elimination, and data representation tools. In addition to providing free and paid plans for individual users, SurveyMonkey offers more large-scale enterprise options for companies interested in data analysis, brand management, and consumer-focused marketing.

One of its limitations is that it supports 10 questions per survey surveys and only 100 responses per survey on the Basic (free) version. There are some limitations also on the survey analysis filters without data export. The best version is the Advantage, which costs 456€ per year. Advantage version provides maximum flexibility and eliminates all above limitations.

The features supported by Survey Monkey are listed below:

- Multiple ready to use surveys and templates
- Recommended questions for the selected survey category
- Previously used questions
- Pre-set questions
- Question builder
- Theme selector
- Logic support
- Print

Finally, the additional features are supported for advanced users and professionals are:

- Advanced survey analysis
- Data export (CSV, PDF, PPT, XLS)
- Advanced data exports (SPSS)
- Expedited 24/7 customer support via email
- Multilingual surveys

SurveyMonkey exports results in .pdf, .xls and .sav files similar to Soci tool.

3.1.3.2 Survey data

The contents of surveys will be defined later in work package 4 for all pilot sites and collected via a central service (as described above). A description of the intended survey for user requirements is provided in deliverable D4.1 and will be detailed in deliverable D4.7. This survey may be combined with a survey on legal issues that will be detailed in deliverable D4.9. An additional survey may be

[8] https://www.ibm.com/analytics/data-science/predictive-analytics/spss-statistical-software
[9] https://www.surveymonkey.com

considered for stakeholder input for the business impact assessments and will be detailed in deliverable D4.4.

Pilot sites can use one of the tools described above compliant with the WP4 requirements. The surveys data will be collected and stored in the centralised sever and made available to evaluators in their initial format.

3.1.4 Metadata standards

3.1.4.1 General principles

This section reviews the relevant metadata standards developed or used in the previous and ongoing field operational tests (FOT) and naturalistic driving studies (NDS) as a basis for the development of the metadata specifications of the pilot data. Such standards will help the analysis and re-use of the collected data within the AUTOPILOT project and beyond.

The text in this section is derived from the work done in the FOT-Net Data project [10] for sharing data from field operational tests. The results of this work are described in the Data Sharing Framework[11]. The CARTRE project [12] is currently updating this document to specifically addressing road automation pilots and FOTs.

As described in the previous sections, the pilots will generate and collect a large amount of raw and processed data from continuous data-logging, event-based data collection, and surveys. The collected data will be analysed and used for various purposes in the project including the impact assessment carried out by partners who are not involved in the pilots. This is a typical issue encountered in many FOT/NDS projects in which the data analyst (or re-user) needs to know how the raw data was collected and processed in order to perform data analysis, modelling and interpretation.

Therefore, good metadata is vital. The Data Sharing Framework defines meta-data as '**any information that is necessary in order to use or properly interpret data**'. The aim of this section is to address these issues and provide methods to efficiently describe a dataset and content of the associated metadata. It results in suggestions for good practices in documenting a data collection and datasets in a structured way. Following the definition of metadata by the data sharing framework, we divide the AUTOPILOT's Metadata into four different categories as follows.

- **AUTOPILOT pilot design and execution** documentation, which corresponds to a high level description of a data collection: its initial objectives and how they were met, description of the test site, etc.
- **Descriptive** metadata, which describes precisely each component of the dataset, including information about its origin and quality.
- **Structural** metadata, which describes how the data is being organised; and
- **Administrative** metadata, which sets the conditions for how the data can be accessed and how this is being implemented.

Full details of these metadata categories can be found in the Deliverables of the FOT-Net Data project such as D4.1 Data Catalogue and D4.3 Application of Data Sharing Framework in Selected-

[10] http://fot-net.eu
[11] http://fot-net.eu/Documents/d3-1-data-sharing-framework/
[12] http://connectedautomateddriving.eu

Cases which can be found at the project website[13].

Field Operational Tests (FOTs) have been carried out worldwide and adopted different metadata formats to manage the collected data. One of good examples is the ITS Public Data Hub hosted by the US Department of Transport. There are over 100 data sets created using ITS technologies. The data sets contain various types of information, such as highway detector data, travel times, traffic signal timing data, incident data, weather data, and connected vehicle data, many of them will also be collected in the AUTOPILOT data. The ITS Public Data Hub uses ASTM 2468-05 standard format for metadata to support archived data management systems. This standard would be a good start point to design metadata formats for various types of operational data collected by the IoT devices and connected vehicles in AUTOPILOT.

In a broader context of metadata standardisation, there are a large number of metadata standards available which address the needs of particular user communities. The Digital Curation Centre (DCC) provides a comprehensive list of metadata standards [14]for various disciplines such as general research data, physical science as well as social science & humanities. It also lists software tools that have been developed to capture or store metadata conforming to a specific standard.

3.1.4.2 IoT metadata

The metadata describing IoT data are specified in the context of OneM2M standard. In such context “Data” signifies digital representations of anything. In practice, that digital representation is associated to a “container” resource having specific attributes. Those attributes are both metadata describing the digital object itself, and the values of the variables of that object, which are called “content”.
Every time an IoT device publishes new data on the OneM2M platform a new “content instance” is generated, representing the actual status of that device.
All the “content instances” are stored in the internal database with a unique resource ID. They can be retrieved from other consumers, including Central Test Servers with simple GET, for e.g.

HTTP GET:
https://<ONE_M2M_SERVER>/onem2m/<APPLICATION_ENTITY>/<CONTAINER>

- The following URL is used for reading instances created in a specific period:

http://<ONE_M2M_SERVER>/onem2m/<APPLICATION_ENTITY>/<CONTAINER>?rcn=5&cra=<dateFrom>&crb=<dateTo>

(`cra` stands for `createdAfter`, `crb` stands for `createdBefore`)
The parameters `dateFrom` and `dateTo` use the date format `yyyyMMddThh`:mm:ss (e.g. 20170325T094522)

[13] http://fot-net.eu/Documents/fot-net-data-final-deliverables/
[14] http://www.dcc.ac.uk/resources/metadata-standards/list

Example of received data:
For single data (e.g. the latest or oldest data):


```
{
  "m2m:cin": {
    "pi": "Byf34qb0pg",
    "ty": 4,
    "ct": "20170505T092155",
    "ri": "ByfhVN6Y1b",
    "rn": "4-20170505092155986eBfi",
    "lt": "20170505T092155",
    "et": "20270505T092155",
    "acpi": [ "/onem2m/acp livorno" ],
    "lbl": [ "03" ],
    "st": 4186,
    "cs": 15,
    "cr": "livorno prod",
    "cnf": "application/json:0",
    "con": "{\"D TS\":\"2017-05-
05T09:21:59.4\",\"F VALORE1\":0,\"F VALORE2\":5.5,\"F VALORE3\":4294967295,\"F VALO
RE4\":444,\"F VALORE5\":555,\"F VALORE6\":0,\"S ID SENSORE\":\"3\"}"
  }
}
```


Looking at the above example, just the payload corresponding to the "con" attribute is the information about the values of the IoT device. The other fields are metadata describing the structure of the information, according to the OneM2M standard.

The IoT device metadata are described in the table below.

Table 1 – OneM2M Metadata for IoT data

Metadata Element	Extended name	Description
pi	parentID	ResourceID of the parent of this resource.
ty	resourceType	Resource Type attribute identifies the type of the resource as specified in clause [5]. For E.g. "4 (contentInstance)".
ct	creationTime	Time/date of creation of the resource. This attribute is mandatory for all resources and the value is assigned by the system at the time when the resource is locally created. Such an attribute cannot be changed.
ri	resourceID	This attribute is an identifier for the resource that is used for 'non-hierarchical addressing method', i.e. this attribute contains the 'Unstructured-CSE-relative-Resource-ID' format of a resource ID as defined in table 7.2-1 of [5]. This attribute is provided by the Hosting CSE when it accepts a resource creation procedure. The Hosting CSE assigns a resourceID which is unique in that CSE.
rn	resourceName	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources. See clause 7.2 in [5] for more details.
lt	lastModifiedTime	Last modification time/date of the resource. The lastModifiedTime value is updated when the resource is updated.
et	expirationTime	Time/date after which the resource will be deleted

		by the Hosting CSE.
acpi	accessControlPolicyIDs	The attribute contains a list of identifiers of an <accessControlPolicy> resource. The privileges defined in the <accessControlPolicy> resource that are referenced determine who is allowed to access the resource containing this attribute for a specific purpose (e.g. Retrieve, Update, Delete, etc.).
lbl	label	Tokens used to add meta-information to resources. This attribute is optional. The value of the labels attribute is a list of individual labels, that can be used for example for discovery purposes when looking for particular resources that one can "tag" using that label-key.
st	stateTag	An incremental counter of modification on the resource. When a resource is created, this counter is set to 0, and it will be incremented on every modification of the resource
cs	contentSize	Size in bytes of the content attribute.
cr	creator	The ID of the entity (Application Entity or Common Service Entity) which created the resource containing this attribute
cnf	contentinfo	Information on the content that is needed to understand the content. This attribute is a composite attribute. It is composed first of an Internet Media Type (as defined in the IETF RFC 6838) describing the type of the data, and second of an encoding information that specifies how to first decode the received content. Both elements of information are separated by a separator defined in OneM2M TS-0004 [3].
or	ontologyRef	This attribute is optional. A reference (URI) of the ontology used to represent the information that is stored in the contentInstances resources of the <container> resource. If this attribute is not present, the contentInstance resource inherits the ontologyRef from the parent <container> resource if present.

3.2 Specificities per pilot sites

This sub-chapter describes the specificities of collected data on each pilot site (e.g. data from VRU in Versailles, data from parking slots in Vigo etc.)

3.2.1 Raw-data in Tampere

3.2.1.1 Vehicle data

The prototype vehicles in Tampere use DDS (Data Distribution Service) to share information between the different modules in the vehicle. Measurement software, which logs all the information, including the raw data of the environmental sensors has been developed. For the AUTOPILOT project, data of relevance include:

- position data

- heading
- vehicle speed
- vehicle acceleration (from IMU)
- target speed
- trajectory
- The received data of the traffic lights.

Data formats/standards
Data will be available in csv file.

Frequency of logging
The logging frequency depends on the sensor. For GPS and IMU the logging frequency is 10 Hz.

Data quality and assurance
Due to the relatively small number of tests which will be performed, a simplified data quality assessment will be performed. The assessment will be mainly performed manually, and include e.g. checks for consistency and removal of outliers.

3.2.1.2 IoT data

At the Tampere test site, the video from a traffic camera is processed automatically to track objects at the parking place (for the automated valet parking use case) and VRUs in an intersection (for the urban driving use case). Information on the object or the VRU will be passed on to a self-driving car. Test scenarios at the test site are planned to include difficult-to-spot VRU movement, e.g. a bicyclist approaching an intersection with high speed, from an angle that may be difficult to see with vehicle sensors before the bicycle comes close.

The traffic camera is of the same type than the ones the City of Tampere has recently taken into use. It will be installed on a special trailer for an easy test setup in various intersections and to provide power for the camera and computers.

The video will be processed with a neural network classifier to mark boundary boxes around road users and to track their movement. Camera installation and on-side calibration will allow for estimation of objects' locations.

Video and recognised objects and their tracking data will be stored for the duration of the test session. Alternatively, the data could be saved only for periods of interest, e.g. when data is being communicated to the self-driving car.

The vehicle will additionally save the data that it receives, along with communication timestamps.

As the test will be carried out on a small area, the test site personnel will mark down the weather of a test day as part of test notes (word document). Alternatively, a SYNOP (surface synoptic observations) message of the closest weather station can be retrieved afterwards from public weather data sources, e.g. by using the OGIMET service[15].

[15] http://www.ogimet.com

3.2.2 Raw-data in Versailles

3.2.2.1 Vehicle data

The VEDECOM (VFLEX) prototypes can produce different types of data:

- **Configuration data**: these are fixed parameters which are related to the identification of the vehicle and its main technical characteristics
- **Localisation data**: these are current values of the position, speed and direction of the vehicle. These data are produced by the vehicle and are periodically sent the OEM's cloud server using the "vehicle-to-cloud" communication means set up by the OEM (cellular communication (4G) while driving, or Wi-Fi/Ethernet when the vehicle is parked). The data are also transmitted by the on board unit through the V2X messages (CAM).
- **Technical state of the vehicle data**: these data are collected by the vehicle bus CAN and sent to the OEMs cloud server using the "vehicle-to-cloud" communication means set up by the OEM. All the data regarding localisation and the technical state of the vehicle are sent to the VEDECOM cloud, and some of the data is then transferred to the OneM2M IoT platform. There is no direct link between the VFLEX vehicles and the OneM2M platform.
- **Operational status of the vehicle data**: these data are related to the operational use of the vehicle for providing services within the AUTOPILOT use cases. They are shared between the vehicle and the applications of the service platform. They are made available to the OneM2M platform via connectors in the cloud.

The vehicle data are defined by using ETSI TS 102 894-2 Applications and facilities layer common data dictionary. Regarding the interface with the OneM2M platform, it follows the work on common IoT data models done by the DMAG and uses the SENSORIS package elaborated by this working group. The main reference documents are ETSI EN 302 637-2 (CAM), ETSI EN 302 637-3 (DENM), ETSI EN 302 895 (LDM), AUTOPILOT – Common IoT data model (from DMAG).

The data which are coming from the VEDECOM cloud server and which are sent to the OneM2M IoT platform use the "JSON" format (vehicle/position, vehicle/status and vehicle/proprietary).

The frequency of logging depends on the sensors, but will be between 1 and 10Hz.

3.2.2.2 IoT data

At the Versailles test site, there will be two connected car sharing stations. These stations will be equipped with smart charging points, able to transmit the status of each charging point and any failure in the system, in real time. These stations will also be equipped with car park detectors, to be able to know, in real time, if the parking space is occupied, reserved or available and, when occupied by an AUTOPILOT car, the identification of this one (vehicle 1, 2 or 3 etc.). These detectors are connected in Lora WAN. The information is used by the fleet management system.

Smartphones are going to be used with two different apps: the first one will be used to connect to the car sharing service (user perspective) and the second for the fleet management operator to get information from the platoon and for the parking manoeuvres (operator perspective).

For the touristic use case, the French pilot site is setting up BLE (Bluetooth Low Energy) beacons in the city centre of Versailles. These beacons will detect every AUTOPILOT car when approaching a point of interest and disseminate touristic audio content through the car's loudspeakers.

For the collaborative perception use case, data from vulnerable road users such as connected bikes (802.11 OCB) or pedestrians wearing smart glasses, smartphones or smartwatches will be sent directly to the Sensinov OneM2M platform before being sent to the vehicle.

3.2.3 Raw-data in Livorno

3.2.3.1 Vehicle data

In Livorno, the data management scheme will be common for all three types of vehicles used on the Italian pilot site and foresees three sets of data.

1st set of data: delivered to the TIM OneM2M IoT platform

A first set of in-vehicle data (basic data) is delivered to the TIM IoT OneM2M platform during runtime.

These data include:

- **Vehicle position and motion**: latitude, longitude, height, heading and speed.
- **Vehicle acceleration**: longitudinal acceleration, lateral acceleration, vertical acceleration.
- **Vehicle status**: steering angle, yaw rate, engine speed, tire pressure, windscreen status, doors status, lights and beams status.
- **Other information:** external temperature, timestamp.

Frequency of logging

Sampling/recording and upload depends on the kind of data, as outlined at the end of this section. Sampling can be periodic, once every second, and the upload once every minute (or even less frequently) either as full sample set or as average value over the period; or sampling can be upon event, and trigger the related data upload to OneM2M.

Data formats/standards

Delivery is done directly via the mobile network using OneM2M compliant MQTT (JSON) protocol.

2nd set of data: stored locally on ISMB in-vehicle IoT platform

A second set of in-vehicle data (full data) is stored locally on ISMB in-vehicle IoT platform and delivered to AUTOPILOT after the tests, based on specific evaluation needs by WP4.

Frequency of logging

This dataset includes at least the first dataset for backup purposes, but potentially more data and potentially with higher frequency (1-10Hz), as available from the in-vehicle system through the ISMB-CRF CAN interface. The purpose of having more data, is to enable a more detailed analysis of use cases, in particular focusing on the in-vehicle IoT platform capabilities.

Data formats/standards

This second dataset could be enhanced by annotations/ observation data after the tests (e.g. textual files). The approach could be similar to DRIVE C2X Field Operational Tests: at consortium level a common data file organisation is defined, then folders are transferred to a central server (which includes a database) either from a terminal PC (e.g. via FTP) or via memory support (USB, DVD).

3rd set of data: stored locally in the vehicle AD computer or other devices

A third set of in-vehicle data (complementary data) is collected locally in the vehicle AD computer or on other devices constituting the prototype (e.g. specific data of connected inertial units, HMI tablets, etc.). These are complementary data are not directly shared to AUTOPILOT, but are needed for the delivery of specific results to AUTOPILOT. This dataset includes specific on board sensors (e.g. frontal sensor), AD control system data and actuation data. It is post-processed within CRF laboratory (or at partner/third party laboratory based on dedicated agreement) to deliver evaluation reports, analysis, or derived charts and figures needed for AUTOPILOT.

In addition, data could be available by specific test tools or parallel observations through systems that are not part of the vehicle (ad hoc videos, testing tools, complex annotations).

This approach is depicted in the figure below:

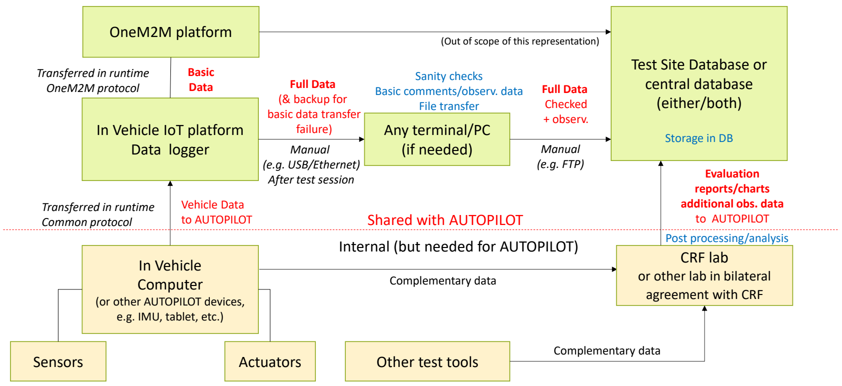


Figure 4 Data collection from vehicles in Livorno

Preliminary specifications are available in the current model for the basic dataset. It includes

- Every minute, the last 60 time-stamped vehicle data in the past minute (i.e. 1 Hz recording) of position, heading, yaw rate and steering wheel angle
- Every minute, speed, lateral, longitudinal and, for vehicles with dedicated IMU also vertical acceleration.
- Every 30 seconds, road surface status
- Every minute, external temperature
- Every minute, engine speed
- Every hour, tire pressure
- Upon change: Low beam status and High beam status, daytime running lights status (right and left), parking lights, parking lights status, fog lights status, reverse light, windscreen wiper status, door opened or closed for each of the four doors.
- With CAM frequency (ETSI-ITS G5 standard): CAM filled with CAN bus or IMU or GPS only depending on vehicle type
- Target speed (details in discussion, possibly sent upon change)
- IoT input used by the vehicle for data fusion, e.g. IoT VRU event, IoT suggested speed (details under discussion, possibly sent upon change)

Currently, the planned on board devices which are connected to AUTOPILOT are pothole detection sensors, inertial units, HMI tablets, connected eHorizon. Data derived from pothole detection inertial units, as processed by the in vehicle IoT platform, are already planned in both the first and the second dataset, and will therefore be transmitted to OneM2M.

Similarly, connected eHorizon data is in discussion, as part of the IoT input data used by the vehicle for data fusion. HMI data (e.g. event information shown on a tablet device) is in discussion with a similar approach. Raw data of these devices may be stored locally, as part of the third (private) dataset, if needed for evaluation. Additional on board devices as cameras, radar, etc. are treated within CRF and they are taken into consideration in the evaluation reports, charts, etc.

3.2.3.2 IoT data

The Livorno pilot site managed by AVR produces a wide array of data that is unprocessed and raw. Among these sets of data are real time traffic cameras with full HD images, Wi-Fi and Bluetooth MAC address loggers. The latter is, as of version 0.11 of this document, processed by a counting and timing algorithm to derive transit time from point A to B of the highway Fi-Pi-Li; in a later developed algorithm, the data containing the MAC addresses will be changed before any pushes to the OneM2M cloud platform are made (salting, hashing are possible answers). Video images are not processed and saved for only a small amount of time (72h) as per the Italian law.

The Livorno pilot site temperature and humidity data will aggregate the data from 12 sensors along the highway that send the above mentioned data every 15 minutes to a central collector device. The data will be stored in a MySQL database with regular data dumps and backup; the data format will be of string type and the quality of the sensors is assessed by a direct approach with a portable sensor by an AVR employee.

3.2.4 Raw-data in Brainport

3.2.4.1 Vehicle data

Different types of vehicles will be piloted for the different use cases (see D1.5 section 1.3 for example).

Toyota Prius / TASS

Toyota Prius vehicles from TASS provide the following vehicle data: Raw data is provided in ROSbags that will be converted to standard format information e.g. in csv files or as PostgreSQL database dumps:

- <u>Communication unit</u>:
  - Cellular network communication via LTE(-V2X).
  - ITS-G5 communication unit
- <u>Sensing devices</u>:
  - GPS module: It estimates the position of the vehicle in WGS'84 format (latitude, longitude and heading). RTK-GPS might be used to enhance the positioning precision. NMEA data may also be provided.
  - Additional sensor data from the radar(s), LIDAR, and camera(s) will not be provided in raw data formats. Instead, detections will be provided as IoT data.
- Vehicle CAN data:
  - Provides normal production vehicle data such as odometers, accelerometers, and information on the vehicle state.
- <u>AD/Units outputs</u>:
  - Actuators: components that act on the commands determined by the AD Unit to control basic vehicle functions such as steering, braking and acceleration.
  - HMI: shows data currently being processed in the on board units.

Toyota Prius / TU/e

The TU/Eindhoven Prius vehicle provides a similar set of vehicle data. In addition, VRU detections will be provided from the on-board camera systems. All raw data is provided in ROSbag formats and will be converted into standard format for uploading to the central data server.

The NEVS prototype provides the following vehicle data in standard output from the DSPace MABX box:

- <u>Vehicle state data</u>:
  - The platform can provide access to interior sensor readings regarding vehicle dynamic states, e.g. rotational or translational accelerations, steering angle, brake pressures, wheel speeds.

Valeo
The Valeo prototype will provide data via the Valeo cloud service:

- Vehicle Dynamics:
  - Actuators: components controlling steering, speed, braking
- Environment sensors:
  - Detections of potholes and other road (surface) hazards from on-board video and LIDAR systems.
  - VW system including lane detection, trajectory, pedestrian detection, mission planning, object detection, traffic signs detection
- Communication:
  - ITS-G5
  - 4G cellular communication

Some vehicles in the Brainport pilots will be equipped with on-board devices in addition to the devices needed for automated driving.

In the highway pilot use case, the vehicle form Valeo is equipped with additional camera sensors to detect road hazards such as potholes and speed bumps. This information will be communicated to the Valeo cloud service in a proprietary format, and considered as a closed system for AUTOPILOT. The road hazards will be provided to the IoT platform in the Brainport pilot site in standard formatted IoT messages.

In the urban driving use case, additional sensor systems will be installed to detect pedestrians. An on-board video camera system will be installed to detect pedestrians. A Wi-Fi sniffer will be installed to detect Wi-Fi enabled devices in the vicinity of the vehicle. Both types of vulnerable road users will be used on-board to enhance automated driving functions, and exchanged via one or more cloud IoT platforms as standard IoT messages.

3.2.4.2 IoT data

The following external devices are transformed into IoT devices, providing the data to the automated vehicles via IoT platforms:

- Drone with camera to detect parking spaces is an IoT device or IoT sensor. The raw camera footage is not shared, only the detections via the IoT platform.
- Road side cameras on the motorway, intersections and parking place are converted to IoT sensors and provide their detections via the IoT platform.
- Smartphones are used with apps to connect to car sharing, platoon formation and automated valet parking services.

The message formats for all IoT devices are being defined.

The Brainport pilot site is equipped with road side camera systems to monitor traffic on the motorway. Additional cameras will be installed to monitor the parking area for Automated Valet Parking. Video footage will not be provided in raw format. Instead the detections of traffic states and parking spaces will be provided via the IoT platforms and cloud services.

The following cloud services will be used via the IoT platforms:

- Traffic Management services provide traffic state data, and traffic control services for traffic light status and authorising the use of the hard shoulder for platooning.
- Parking, routing and motion planning services will be provided as part of the AVP use case.
  - The Valet Parking Service provides information on free parking spaces to automated

vehicles.

  - The routing service provides an obstacle free route from the current vehicle position to the target location, e.g. for platoon creation point or the drop-off or parking spot.
  - The motion planning service returns a trajectory for a requested route.
- TomTom HDmaps cloud service provides information on road hazards.
- A car sharing service provides information on car sharing, including pick-up location, pick-up time, and route to pick-up location.

3.2.5 Raw-data in Vigo

3.2.5.1 Vehicle data

In the Vigo test site the data from the vehicles is going to be recorded mainly through the different CAN networks inside the test vehicles. This includes the information coming from the vehicle directly, from the autonomous functions and sensors and from the V2X services.
The recorded information in the vehicle is going to be synchronised with a local server in the CTAG premises to be processed according to the requirements in a next step.
The following list represents a set of information that can be obtained from the vehicle:

- Vehicle Bus
- GPS
- DGPS
- Location System
- Vehicle Dynamics Sensors
- Sensors
- V2X CU
- IoT CU
- Speed
- Long Acceleration
- Lat Acceleration
- Timestamp

Data formats/standards:
The information in the vehicle is going to be recorded in raw CAN format to create logs in text format. Once this logs are located in the local server this files are going to be processed in order to translate all the data to a more readable format. The output of this data process can be a CSV file or a SQL database.

Frequency of logging:
The frequency of the data recorded is going to depend on the source in each case. As a regular basis most of the data will be recorded at 10Hz, except for the GPS, that will be recorded at 1Hz.

Vehicles in the Spanish pilot site will be equipped with an on-board device capable of detecting vulnerable road users (VRU). The information detected by this device will be considered for both Spanish use cases: urban driving and automated valet parking (AVP), where it will provide useful data about VRU, allowing the vehicle to act accordingly. This information will be published to the IoT platform following the standard IoT message and becoming available for any other user that might need it.

3.2.5.2 IoT data

Parking spots / AVP
The parking spot service operates with a static existing map of the parking spots. This allows us to build an effective spatial index for fast geographical queries. However, the service also supports real-time changes of the map which are coming through the IoT platform. This helps to handle actual changes in the parking map by updating the static part of the map. The service supports two different types of data modes. The first type consists of IoT data models which describe possible changes of the parking map and parking spot availability states. The second type of data models is used by the clients of the service to access parking information via a public API.

In the context of the AVP use case, the parking service will manage the previously mentioned parking spots, allowing the users to book a spot in advance, and publish the parking status information to the IoT platform, making it available to the parking spot service (provided by IBM) and described above. The parking service will also provide routing information to the vehicles performing the AVP use case, indicating the route that the vehicle must follow to reach its parking spot.

External devices
Within the Spanish pilot site, there will be three different external devices that will be connected to the IoT platform and will provide information to the vehicles:

- Camera device detecting vulnerable road users (VRU): The raw camera data will be internally processed and converted to IoT data that will be sent to the IoT platform following the corresponding IoT message.
- Hazards control centre detecting three kinds of hazard events: road work warnings, accidents and traffic jams. This control centre will act as an IoT device, publishing any new event into the IoT platform.
- Traffic lights that will publish their status periodically in the IoT platform following the defined IoT message. This will allow to the connected vehicles to follow the traffic light indications on the road.

These three kinds of devices will be connected and they will publish their information to the IBM Watson IoT platform. They will be available to the other IoT platforms through the OneM2M connector.

3.3 Synthesis of available data

Each pilot site has filled in an excel sheet showing which of the data required for evaluation will or will not be available on their site. This gives a first synthetic view of the data available on each PS. The spreadsheets are available in the annex 3.

4 Data collection and integration – Distributed data management

Chapter 4 is about how the data presented in chapter 3 are going to be collected on each PS.

4.1 Common methodology

The main objective of the D3.6 is to define the methodology to be applied in AUTOPILOT across all pilot sites. This includes the explanation of the common data collection and integration methodology.

One of the main objectives within T3.4 "Test Data management" is to ensure the comparability and consistency of collected data across pilot sites. In this context, the methodology is highly impacted by the pilot sites specifications of the Task 3.1 and compliant with the evaluation methodologies developed in the task 4.1. In particular, Technical evaluation primarily needs log data from the vehicles, IoT platforms, cloud services and situational data from pilot sites to detect situations and events, and to calculate indicators.

The log data parameters that are needed for technical evaluation are organised by data sources. The following list identifies the groups of sources:

- **Vehicle sources:** the list of in-vehicle data sources including the on-board sensors and systems connected to the vehicle network
- **Vehicle data:** the specific log parameters for the vehicle sources
- **Derived data:** parameters that will be derived from the log parameters
- **Positioning:** GPS information in standard NMEA format
- **V2X messages:** data elements from V2X messages, including the ITS-G5 messages for CAM, DENM, IVI, SPAT,MAP
- **IoT messages:** The IoT messages generated from the connected devices and infrastructures.
- **Events:** the relevant events from pilot sites. They should be defined for communication application logic, and user interactions.
- **Situations:** the functions of the parameters computed from the previous data sources
- **Surveys and questionnaires:** data resulting from the answers of surveys and questionnaires for user acceptance evaluation

For IoT data, Some Pilot sites use proprietary IoT platforms in order to collect specific produced IoT data with specific devices or vehicles (e.g. the Brainport car sharing service and automated valet parking service use Watson IoT Platform™ to collect data from their vehicles).

On the top of that, we have a OneM2M interoperability platform in each pilot site. This is the central IoT platform for exchanging IoT messages relevant to all autonomous driving (AD) vehicles at pilot site level. Then, the test data will be stored in pilot site test server storage that will contain mainly the vehicle data, IoT data and surveys data. Further, the test data will be packaged and sent to the AUTOPILOT central storage that will allow evaluators access all the pilot site data in a common format. This includes the input from all pilot sites and use cases and for all test scenarios and test runs.

Every PS has its own test central server storage for data collection (distributed data management). In addition, there is a central storage where data from all pilot sites will be stored for evaluation and analysis. The centralised data management process is deeply explained in chapter 5.

The following figure represents the data collection methodology used in AUTOPILOT across all pilot sites.

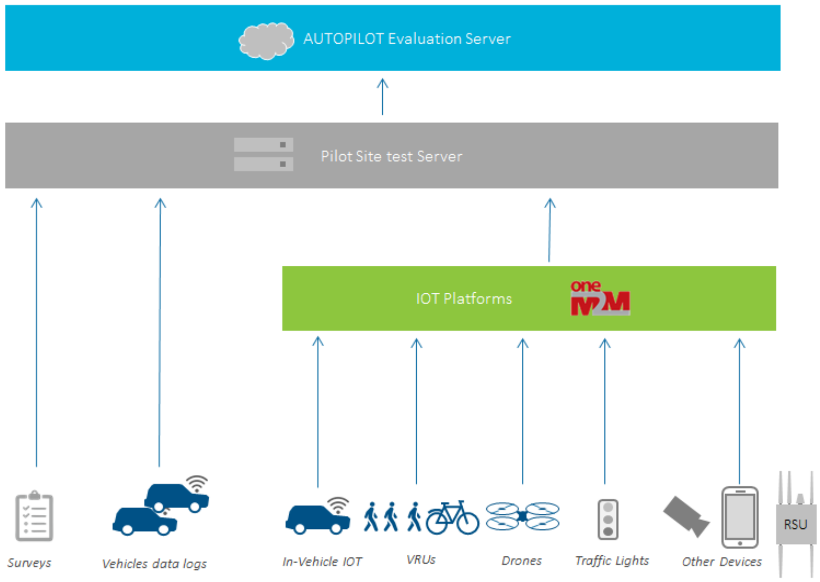


Figure 5 Generic scheme of data collection in AUTOPILOT

From the WP4 perspective, evaluators have provided some requirements for the log data provisioning in order to be analysed. As results, pilot sites should try to cover the following high level requirements:

- **Data format**: data should be provided in standard formats that can be accessed and processed with standard (preferably open source) tools, such as in SQL/NoSQL databases, CSV, XML,JSON or TXT files
- **Timestamp**: all data should be timestamped. UTC is proposed to be the single time reference for all data providers
- **Synchronisation**: All stations and applications that log data must be time synchronised within 100 msec accuracy and need to be verified (Task 2.5, Task 3.2) before piloting.
- **Processing Load**: The CPU and memory loads should be logged on all station processing units involved in automated driving and communication.
- **Communication**: to evaluate communication performances, the message content and also the locations and timestamps upon sending and receiving should be logged. In addition, Every message is logged with a communication action to identify whether the message is 'SENT' or 'RECEIVED" by the station
- **Measurement identification**: in order to uniquely identify the measurement, it's necessary to verify that every measurement is complemented with:
    - Unique id of a host station logging the measurement
    - Unique id of the application logging the measurement
    - Timestamp at which the application logs the measurement
- **Faulty Errors**: in order to be clean, faulty logger data should be removed to have good quality data

The next sections describe the data management methodologies applied in each pilot site. This includes the architecture description with high level components, data collection process, data quality process and deep dive into the data integration within use cases.

4.2 Data collection in Tampere

4.2.1 Architecture

All data from Tampere tests will be manually collected by VTT to a single storage, including also test metadata, such as related test plans and notes. This is to compile a ready dataset for evaluation purposes. The dataset will include vehicle data imported into a PostgreSQL database format, video data from periods of interest, video annotations and survey results. Essentially the collection will consist of data that can be widely used within the project consortium.

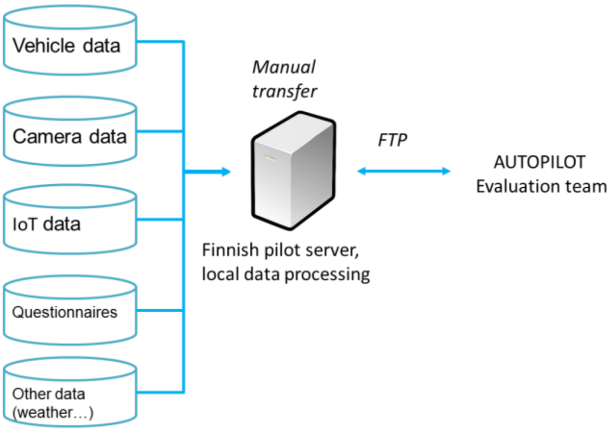


Figure 6 Architecture of the data management system for the Finnish pilot

Other partners can access the dataset with FTP. They will need a password for opening encrypted files.

4.2.2 Data collection

The data will be collected manually at the end of the test days, from the vehicle, road side unit and the IoT platform. Data to collect all data published on DDS, which is used both in the vehicle and the mobile road side unit, on which the traffic camera is installed, has been developed.

No data transformations are planned at test site level. Data will be logged as .csv files both in vehicles and roadside stations. Integration covers collecting log files from vehicles and roadside stations to a single server.
The data will be combined in a test data package, which includes also predefined test information. Both data collection and input of the test metadata will require manual actions by the person responsible for the tests.

The test site will later clarify the need to import csv files also to a PostgreSQL database.

4.2.3 Data quality

Once all data are uploaded for a test session, the data is analysed in several steps. The different signals are checked for their plausibility (verifying that the sensor data is within the expected range), and is checked for missing data. Remarks involving the quality of the data will be included with the test metadata.

4.3 Data collection in Versailles

This section describes the relevant elements related to the data management on the French pilot site. It explains the global architecture of the pilot site with a focus on the methodologies of local data management. More specifically, these methodologies handle data collection, data processing, data quality procedures and data storage means used in Versailles.

4.3.1 Architecture

The French pilot site will take into account the data flows produced by the following data sources: Vehicles, RSUs, VRUs and other IoT which is represented in the architecture below:

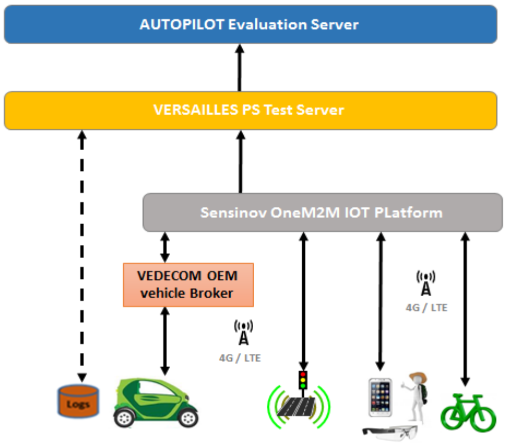


Figure 7 Data management architecture of the French pilot site

The concerned data flows in Versailles AUTOPILOT architecture are described below:

- Vehicle data is first collected by VEDECOM's service platform and then transferred to Sensinov OneM2M IoT platform
- VRU data, RSU data and other IoT data go directly to Sensinov OneM2M platform
- Vehicle logs are stored in hard drives

The pilot site data management features the following components:

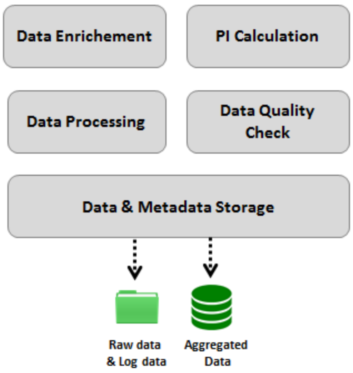


Figure 8 Components of the French data management

After the data collection process, data is stored in the PS data platform, and then the PS checks the data quality before processing it and sending it to the AUTOPILOT evaluation server.

4.3.2 Data collection

The data collection process is done in two distinct ways:

- Data read from Sensinov OneM2M IoT platform (vehicle, VRU, RSU, IoT data)
- Data manually grabbed using physical support (vehicle logs)

Results from questionnaires and interviews will be collected directly by the PS data platform.

Whatever data source, test data is collected in a test data package containing predefined test information, data description fields (test metadata), test data and the appropriate data model. The data preparation phase will require a manual intervention of the data provider/owner in order to provide all useful and required information about data about to be sent.

It is important to mention that all the data are using standard formats (CSV, TXT, XML …) in order to fit with evaluation requirements.

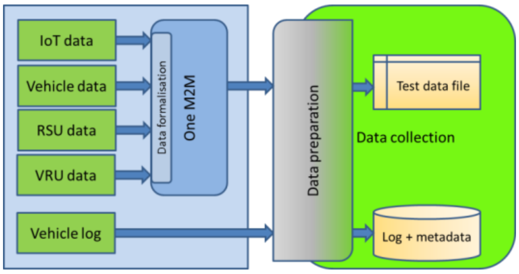


Figure 9 Data collection process in Versailles PS

The following table summarises data sources, descriptions, data providers and data model providers.

Table 2 – Data sources in Versailles PS

Data Sources	Description	Data providers
CAN Vehicle Data	All Vehicle data generated from CAN bus and stored in hard drives and transferred manually to the PS data platform.	VEDECOM
Vehicle Sensors Data	All the parameters or measurements from vehicle based sensors	VEDECOM
ITS-G5 / 3G-4G Data Log (CAM, DENM)	Data logs generated from every sent/received message using the communication protocols (ITS-G5, 3G/4G …)	VEDECOM
IoT Data (including RSU & VRU)	All the data coming from connected devices (VRUs, RSU, smartphones, wearables …)	CONTI / VED / SENSI/ AKKA / CERTH
Questionnaire & Surveys	The data collected from questionnaires and surveys with users and stakeholders.	VEDECOM

Data processing

Data processing will cover the following steps:

- Filtering data to prepare more accurate data for data analysis
- Calculating derived variables and performance indicators: the derived variables and performance indicators are calculated using agreed formulas
- Linking events with other information sources e.g. getting traffic data and linking them with vehicle logic for example:
  - Situations or application states,
  - Events that are detected within these situations,
  - Actions or decisions made by the application or service logic
- Building a ready to use data set that will be:
  - Stored in a FOT database (optional)
  - Shared with the centralised test server

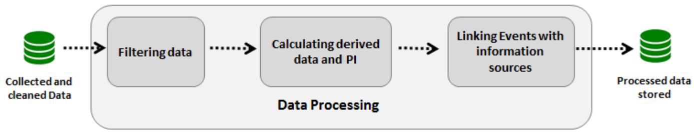


Figure 10 Data processing on the French PS

Data storage

The figure below represents the data storage means on the French pilot site:

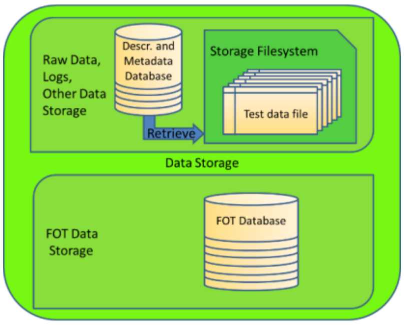


Figure 11 Data storage means on the French PS

The PS test server will contain two distinct types of storage:

- A database and a file system for raw data, logs and derived data
- A database (FOT optional) for test data

The "Descr. and Metadata Database" shown in the figure above contains all the useful information about data files stored in the dedicated filesystem, and is used to retrieve this data in the filesystem.

Metadata will be stored with their appropriate design in addition to data in databases.

Data sharing

Once the data are cleaned and processed, database dumps and/or aggregated data will be transferred to the AUTOPILOT evaluation server. The transfer can be done through network transfer protocol, available APIs or database transfer.

4.3.3 Data quality

The aim of data quality is to check that the data can be useful and ready for processing. This is done by comparing the values to typical signal characteristics (frequency, missing data, accuracy, unit standard for measurement). As an example, it is necessary to check that the signal is within the sensors' range or the natural range (physical characteristics ….) and to create a quality measure/value that assesses and identifies the data quality.

4.4 Data collection in Livorno

The main collection tool for the piloting data is the OneM2M platform, since it receives all the data produced by the devices with a role in the use case and make them available for both the implemented services and the evaluation activities of the project.

4.4.1 Architecture

The TIM IoT platform, based on OneM2M standard [16], is provided as PaaS: therefore TIM doesn't provide the source code. Figure 12 describes the high level architecture of TIM OneM2M platform, based on Ocean project [17].

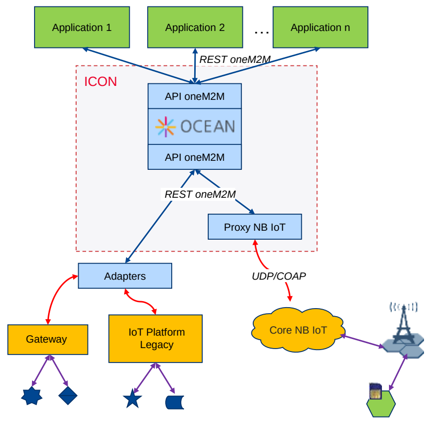
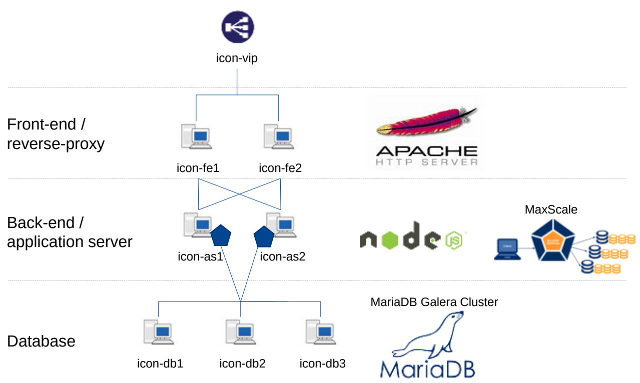


Figure 12: High level functional architecture of TIM OneM2M platform (Source TIM Internal documentation)

The platform has been implemented basing on the three-tier architecture (refer to

Figure 13).



---

16 http://www.onem2m.org/

17 http://www.iotocean.org/

Figure 13: Three-tier architecture and technologies (source TIM internal documentation)

4.4.2 Data collection

A set of in-vehicle data is delivered to the TIM IoT OneM2M platform during runtime in order to perform the analysis for answering the research questions based on the OneM2M platform indicators. Delivery is done directly via the mobile network using OneM2M compliant MQTT (JSON) protocol. An alternative flow towards the TIM IoT OneM2M for vehicle data is provided by the RSUs. Actually, the AUTOPILOT vehicles periodically broadcast over the ETSI G5 radio CAM messages which are received from the RSUs in communication range. The RSUs in their turn forward the timestamped payload of these messages via mobile network (LTE) to the TIM OneM2M platform, to be consumed by traffic control services.

Thus the TIM OneM2M collects and stores two kinds of data from the vehicles: the dataset transmitted at runtime by the in-vehicle platform and the dataset collected indirectly via short range communication from local roadside units. All data and metadata stored in the TIM OneM2M IoT platform are available for uploading to the AUTOPILOT central test server.

Data are sent to the OneM2M platform from sensors, gateways and clouds; in order to identify the type and the structure of the data being collected, Italian Pilot site partners defined an excel file to collect this information. In table 3 are reported the parameters of interest identified.

Table 3 – Data collection parameters

Parameters	Instruction
Data type	Description of data
Unit of measurement	Unit of measurement
Direction (to or from OneM2M platform)	Data could be "sent to" / "received from" OneM2M platform
Periodic data (yes/no)	A data could be periodic (and in this case specify the period) or "event basis" (give an idea of an average data quantity per hour/day)
Interface from/to OneM2M platform	Specify if it is available an interface compliant with OneM2M standard (HTTP, CoAP, MQTT...) or not compliant and in this case give more details about the interface
"PUSH or PULL model (ONLY for data from OneM2M platform to TCC/CC)	Specify if the data should be notified by OneM2M platform to the external node (SUBSCRIBE) or the external node reads the data on OneM2M platform

In table 4 is provided an example regarding a water detection sensor.

Table 4 – Water detection parameters

Parameters	Water detection sensor
Data type	This field is represented by an Integer [0, 1]. If water is present on the road surface above a fixed threshold, its value is 1, otherwise is 0
Unit of measurement	Integer [0, 1]
Direction (to or from OneM2M	SENT TO OneM2M platform

platform)	
Periodic data (yes/no)	Periodic with a period of 1 hour. If a state transition occurs (water present -> not present and vice versa) the first 10 message transmissions have a period of 30 seconds
Interface from/to OneM2M platform	UDP (JSON payload in hex format) /COAP (protocol not yet available, payload in hex format)
"PUSH or PULL model (ONLY for data from OneM2M platform to TCC/CC)	NA

4.4.3 Data quality

The security aspects of oneM2M platform are based on the following features:

- Services and APIs OneM2M are exposed with SSL (HTTPS)
- Authorisation mechanism based on credentials (username/password) of a specific user (tenant)
- Creation of Access Control Policy (ACP) for each Application Entity (AE)

All the data produced/consumed by the devices during the piloting activity will be collected by the OneM2M platform and made available to the AUTOPILOT central database.
If other kind of data will be produced during the piloting operation, such as those coming from survey to users or stakeholders, other data collected by on-board equipment during road testing, they will be digitalised, stored in a data server and made them available to the AUTOPILOT central server.
The overall data management architecture for the Italian pilot site is shown in Figure 14, where the ancillary optional database is called FOT database.

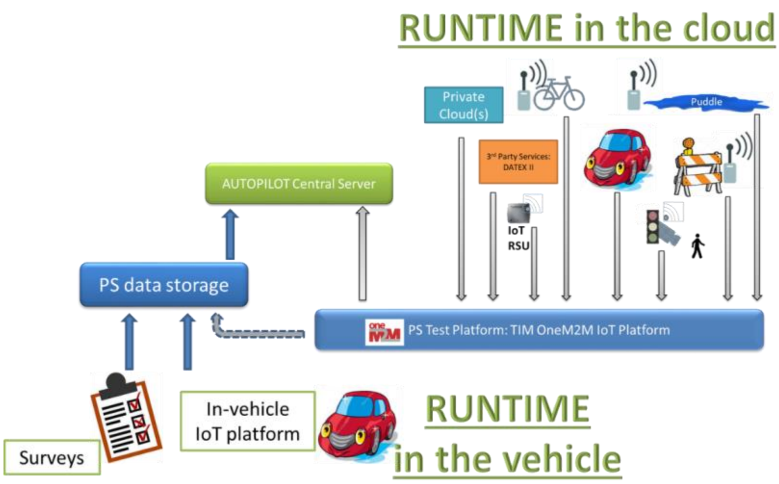


Figure 14: Data management architecture of the Italian PS.

4.5 Data collection in Brainport

4.5.1 Architecture

The Brainport implements a distributed approach to data management. The use cases will use different IoT platforms, IoT devices, cloud services and automated vehicles. Figure 15 shows some of the types of platforms, devices and services as an example. Data will be collected and stored in multiple data management systems.

At the time of writing the data management solutions are not completely defined for all use cases. This section describes a data management solution used in the Brainport pilot site.

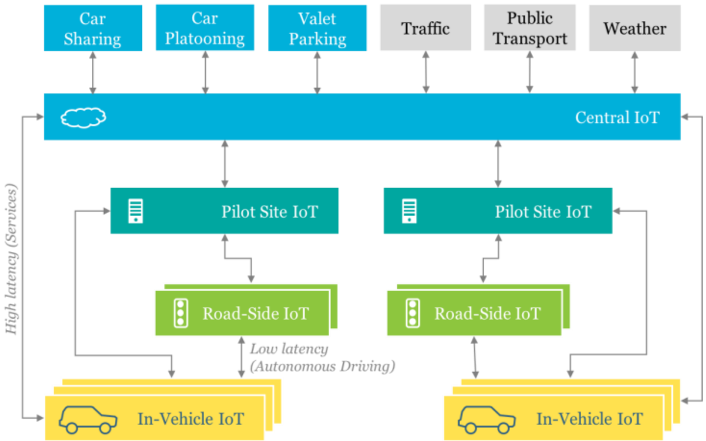


Figure 15: IoT platforms and services in the Brainport pilot site

4.5.2 Data collection

The rationale for data collection and management is sketched in Figure 16. The data collected from vehicles is primarily the detections from the vehicle sensors, external communication, and the relevant input and output to automated driving functions. Similar data is also collected from other stations involved in V2X communication, such as ITS-G5 road side units. The log data is collected remotely and stored in the data storage called the "central repository" – this is not the centralised test server described in section 6.

Data from IoT devices is collected at the IoT platforms where possible. This avoids the necessity to collect data from the IoT devices. This also avoids the necessity to collect data from cloud services. Data received and sent from the IoT platforms will also be stored in the same data storage with the vehicle data.

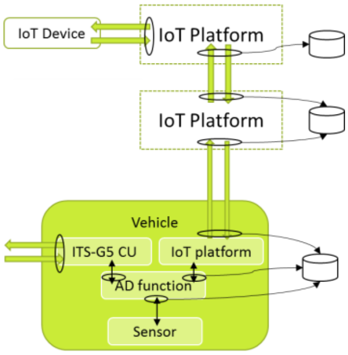


Figure 16: Collection of data from IoT platforms and vehicle platforms

Log data formats will be defined to harmonise the logging. All log data will be appended with the mandatory data elements for each log items defined in table 5. The Meta data from table 1**Error! Reference source not found.** allows evaluating the data flows, communication delays, and other communication performance indicators. The messages being communicated will be logged exactly as defined by the standardised or proprietary message formats, especially the data element names and message structure. Data elements that are irrelevant for evaluation or data analysis purposes may be omitted.

Table 5 – Mandatory data elements for data logging

Data element	Description
log_stationid	Identifier of the log station that logs the log item. The log_stationid should be unique within the AUTOPILOT project.
log_applicationid	Identifier of the log application (in the log station) that logs the log item. The log_applicationid is at least unique within the log station (log_stationid).
log_timestamp	Timestamp at which the log application logs the log item.
log_action	Enumerated value identifying the action in the data flow to, in or from the application: {SENT, RECEIVED, …}

Relevant information will be logged from the automated driving functions in the vehicles, such as the input and output, events, actions and decisions. Most important is to log decisions on the timeliness, geographical relevance, accuracy and contents of received information. This logged information will be used to evaluate and assess the relevant of alternative data sources for the functionality and performance of automated driving functions.

Data storage

Figure 17 show the process for collecting, storing and analysing data. The data collected from vehicles, road side and central ITS-Stations and IoT platforms will be stored in a central repository. All data from all devices and platforms are stored per test session in a single PostgreSQL data base. A test session is the execution of a single test and scenario in a test plan. All data from ITS-Stations, IoT devices, IoT platforms and services that interact during a single test session will be collected and analysed collectively.

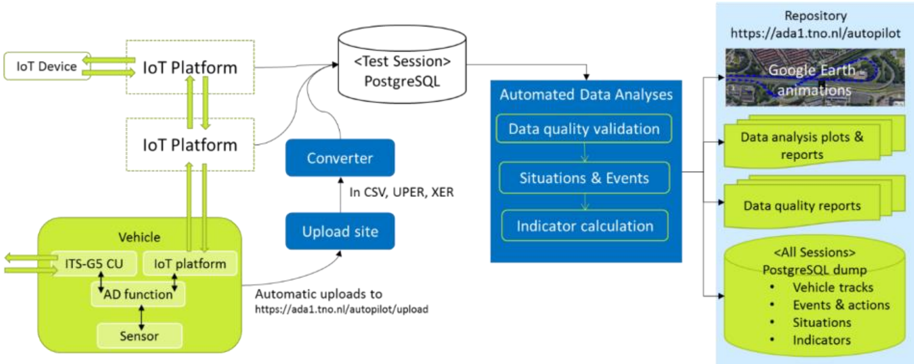


Figure 17: Data collection and management for Brainport platooning use case

The system presented in Figure 17 was originally developed for technical evaluation in DRIVE-C2X, and further developed for Day1 C-ITS services and the latest standards in INTERCOR and C_ROADS. The structure of the central repository and log formats are specified on the INTERCOR project site and in the Brainport pilot site section on AUTOPILOT Project Place.

4.5.3 Data quality

Once all data are uploaded for a test session, the data is analysed automatically in several steps.

First the quality of the log data is verified on syntax, missing data that is mandatory for specific evaluation criteria, frequency, and value ranges. Additionally, the plausibility of logged is assessed, for example on time offsets and synchronisation, kinematics, and correlations of events in messages, application triggers and actions.

Second step is to detect situations and evaluation events in the scenario, based on received messages and application events. In the third step, indicators are calculated for the situations and events. The consistency of these results and correspondence to test plans and expectations provide a final step in the quality assessment of the logged data and to determine the success or failure of the executed test session.

4.6 Data collection in Vigo

4.6.1 Architecture

The main structure for the data management is represented in the following figure.

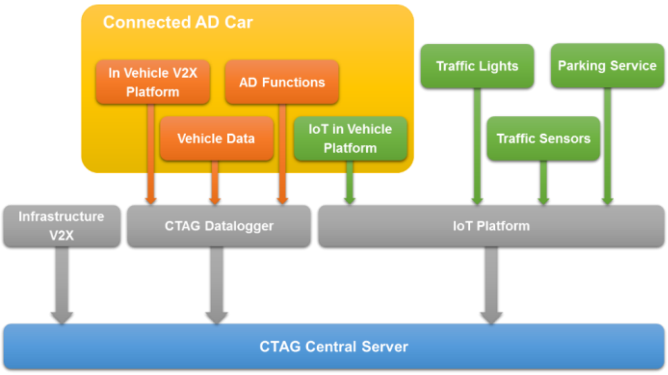


Figure 18 Data management structure on the Spanish PS

There are three different data sources identified in the Vigo PS:

- IoT platform
- Connected AD Car
- V2X infrastructure data

4.6.2 Data collection

All the data related to the IoT platform will be recorded directly in the platform according to the specifications of the project.
In the vehicle there is installed a data logger that will record the following data:

- Vehicle data (including positioning data)
- In vehicle V2X data
- AD functions data

The information recorded in the data logger will be synchronised with a server located in CTAG's premises devoted for data management. This server will receive also the data recorded in the IoT platform and the data generated by the V2X infrastructure.

Once the data is located in the pilot server it will follow the scheme described in the Figure 19.

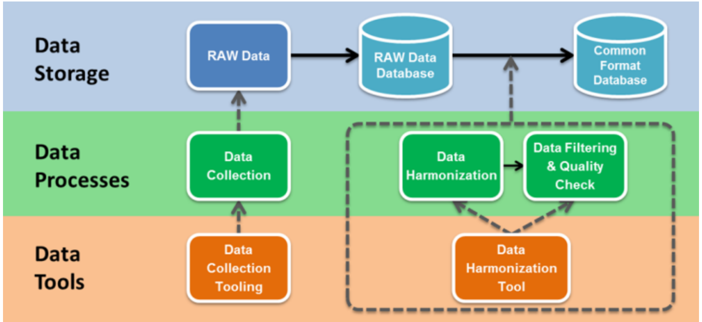


Figure 19 Data processes integration on Spanish PS

4.6.3 Data quality

In a first step the data is received in raw format provided by the data collection tooling. There is a second tool in the server responsible for executing two main processes:

- Data harmonisation: In this process the different data sources in raw format are processed for synchronise and adapt the data to the project requirements.
- Data Filtering and Quality Check: This process is checking the different data inputs in order analyse formats, data ranges, naming etc. This will avoid generating wrong information. This process generates a report for every analysed input.

5 Centralised Data Management

This chapter explains how the data is centralised after being collected.

5.1 General Description

The centralised test server (CTS) is a centralised platform which aims to collect and store the data from the pilot sites and allows the evaluators to browse, access or download data, but also to enrich the collected data and store own results. All the data stored in the CTS has previously been anonymised by the data providers.

The CTS offers various APIs (with or without HMI) for pilot 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 is going to offer a HMI to the evaluators in order to easily browse through the available data, search for specific data using metadata information, and download selected data.

A direct access to the databases will also be opened for the evaluators' usages, in order to make possible any query relevant operation for evaluation: selects, updates, inserts, exports.

5.1.1 Data ingestion system

The uploading of the pilot sites test data to the CTS is done at the initiative of the pilot sites and possible in two different ways:

- Direct ftp uploading
- Upload client (application or web service)

Note: other uploading options might be feasible according to the pilot sites specificities or willingness and will be discussed with partners.

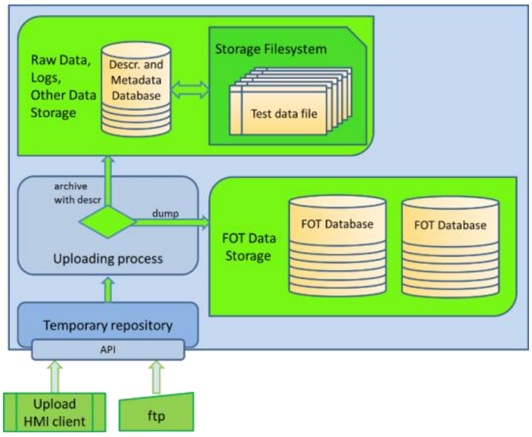


Figure 20 Data upload process for Pilot Sites

The temporary repository in which the data is received is watched by an "uploading process" in charge of launching packet analysis and storage. As mentioned before, the upload-HMI will allow the data owner to follow the uploading status in real-time. The uploading and ingestion status are also accessible via other ways (evaluator point of view for instance) allowing any users connected to the CTS to track all the tasks the system is working on.

5.1.2 Data access by the evaluators

The browser HMI allows the evaluators to browse the content of the CTS, using filtering criteria, either simple fields (PS, date, use case, etc.) or metadata. Retrieved data will be downloadable in one click.

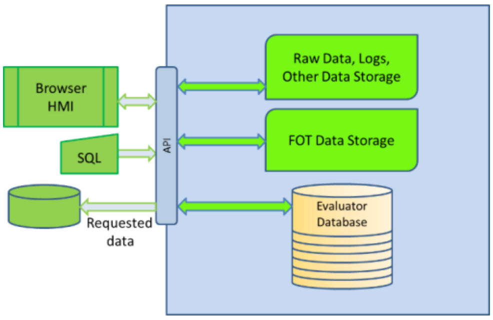


Figure 21 Data access process to evaluators

When accessing to the data stored in the CTS, the evaluators have the possibility to store any other information related to the collected data and which they would like to add. All the information (collected and added later) is then available in one place and accessible to others.

5.2 Functional description

A schematic functional view of the main components of the centralised test server is presented below:

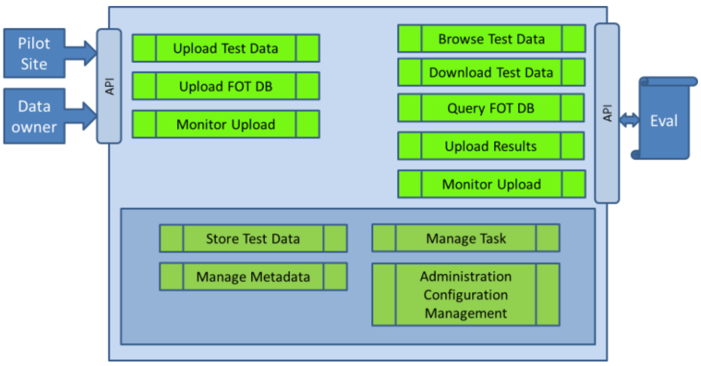


Figure 22 Main components of the CTS – Functional view

5.2.1 Components related to the 'pilot site API'

- Upload Test Data: Collects data sent by pilot sites and sends it to the storage component

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

- Upload FOT DB (optional): Import a FOT database in case of pilot site sending a database dump

If pilot sites have FOT databases, they might send database dumps, instead of raw test data. In this case, this dump will be used to restore a copy of this database in the CTS. These dumps must also be sent with metadata in order to be able to get a precise description of their content. This description will be accessible in the data browser, and the evaluators will have an access to any restored FOT database they are interested in.

- Monitor upload: Display the progress and status of uploading tasks

The monitor logs the activities related to the upload, so the data owner can check the status of uploading. In case of failure, the problem can be traced (e.g. lost connection, servers down, invalid data…). This component is also accessible to the evaluator.

5.2.2 Components related to the 'evaluator API'

- Browse Test Data: Available test data browser: display, search and access test data

The browser displays all available data and uses description data and metadata to filter and search among data. Test data displayed in the browser can be easily downloaded.

- Download Test Data: Enable the evaluator to download test data from the CTS

After browsing test data through the web interface, the evaluators can choose to download the desired datasets. The CTS will ensure an easy access to the browsed data, so that the evaluators can download the data manually or using specific tools (API, scripts...).

- Query FOT DB (optional): Enable the evaluator to perform queries (search, filter) on the FOT database through a generic SQL browser tool

A generic database browser will enable the evaluators to access and browse (read only) the database content for their use. The credentials for a direct access to the databases will also be provided to the evaluators to let them connect their own applications directly to the source databases.

- Upload results: Store the evaluation results in the CTS

The evaluators will have the possibility to upload their evaluation results, by providing a data result archive with its description file, in the evaluation results database.

- Monitor upload: Display the progress and status of uploading tasks

Refer to 'Monitor upload' in section 5.2.1.

5.2.3 Internal components

- Store Test Data: Stores test data in the CTS databases and dedicated filesystems

The CTS main feature/component is the test data storage which aim is to centralise all the test data from the pilot sites needed for the evaluation. This is done using a test description database and a file system for the test data logging, one or more databases for the FOT databases dumps, and also a database for the evaluators' results.

- Manage Metadata: Metadata manager related to various upload and storage components useful for browsing test data.

- Manage Task: Simple tasks management, creates, updates, stores uploading tasks and their status.

The CTS will offer an uploading task management feature within its interface that will show all task details (running and terminated tasks). The task management handles the task creation, update, progress and status. This way, uploading tasks can be monitored while running.

- Administration/Configuration/Management: Configuration and administration components only accessible to the administrator.

It is used to manage the configurations of database servers, file systems, APIs, host machines, web applications and other interfaces.

5.3 Test data description

As mentioned above, the PS leaders have to upload their test data to the CTS. These data are organised in archives (.tar, .zip, etc.) and contain test data files with their description. This description is organised in an .xml file and filled by the data provider or use case responsible. The test data description contains metadata related to the uploaded test data. This allows the evaluators to have more information about the test data. The following table describes the content of the test data description. All parameters identified with (*) are considered as mandatory.

Table 6 – Test data description file content

Test data Description Content			Description
Test Properties	Pilot Site *		Name of the Pilot Site
	Use Case*		Name of the use case tested
	Name *		Name of the test (Archive file name)
	Description *		General Description of the test
	Session Name		Name of the test session / run
	Text Context *		Description of the test context. This contains the test conditions, the weather conditions , number of cars ...
	PS Data Manager *	Name	Name of the Data Manager
		Email	Email of the data manager
	Test identification	Test Point Name	Name the references the test sub-session (Platoon formation / Platoon Run / ENUM or Not ENUM TBC)
		Identifier	The test identifier
		Test Status	The test status (complete, fail …)
	Test Date	Date	The test data time. It should be in YYYY – MM –DD (ISO 8601 format)
		Start Time	Starting time of the test in UTC reference time
		End Time	Ending time of the test in UTC reference time
	Publish Status		YES/NO (To be visible to anyone or not) by default = YES
Data Files	Name *		Name of the test data file. It should be relevant to the file content
	Type *		Reference of the data type (CAM, DENM, IVI …)
	Format		The data format used to store data
	Comment		Comment that could be useful to enrich the metadata or the content
	File Time		The time of the file generation. It should be YYYY – MM –DD (ISO 8601 format)

5.4 Uploading data from PS test platforms

The CTS will provide to all pilot sites a graphical HMI based on a web application. With this application, the PS data manager will be able to select in his/her filesystem the test data files he/she wants to upload to the CTS. The application will display a friendly interface to add mandatory information useful to describe each data file and the whole upload. These information will be compiled in the .xml description file. Once the preparation process completed, the upload button will become sensitive. The PS data manager will start the upload at his/her convenience. The application is based on the sftp protocol. The test data will be uploaded to the CTS using a secured communication channel.

5.5 Providing data to evaluators

The CTS collects and hosts data sent by all PS test servers. These data are described in the first chapters of this document.

The services offered to the evaluators consist in enabling the evaluators to:

- Browse through available data using metadata (description) filters
- Download data
- Upload their own test data
- Upload their evaluation results
- Browse available evaluation results
- Download evaluation results
- Optionally, in case FOT databases management, it is possible to query and dig in these FOT databases on the CTS.

If the evaluators produce new/enriched test data, they can upload it to the CTS, just as they can upload their evaluation results.

In the case CTS provides FOT databases, an API for direct access to these FOT databases is going to be opened so that evaluators can use their own software to access and pull wanted data.

The CTS does not offer any analysis tools or calculation means, nor graphical representation of data, parameters or evaluation results.

Evaluations and work on the data have to be done outside the CTS, on downloaded data.

5.6 Evaluation data description

As mentioned above, the evaluators have to upload their evaluation results to the CTS. These data are organised in archives (.tar, .zip, etc.) and contain evaluation data files with their description. This description is organised in an .xml file and filled by the data provider. The evaluation result description contains metadata related to the uploaded result. This allows the evaluators to have more information about the results. The following table describes the content of the file description. All parameters followed by the xsd schema representation. All parameters identified with (*) are considered as mandatory.

Table 7 – Evaluation result description content

Evaluation Result Description Content			Description
Evaluation Result Properties	Pilot Site *		Name of the Pilot Site
	Use Case*		Name of the use case tested
	Name *		Name of the test (Archive file name)
	Description *		General Description of the test
	Session Name		Name of the test session / run
	Comment		A open text field to add any comments
	Conclusion		A open text field to add any comments or conclusions
	Evaluator *	Name *	Name of the Evaluator
		Email *	Email of the Evaluator
	Date *		It should be in YYYY – MM –DD (ISO 8601 format)
	Evaluation Name *		Name of the evaluation
	Publish Status		YES/NO (To be visible to anyone or not) by default = YES
	Analysis Status		Completed, Incomplete, Failed, Planned, Postponed, tbc
Evaluation Result Files	Name *		Name of the data file. It should be relevant to the file content
	Type *		Reference of the data type (list to be defined)
	Format *		The data format used to store data
	Comment		Comment that could be useful to enrich the metadata or the content
	Date *		It should be in YYYY – MM –DD (ISO 8601 format)

6 Data sharing

The text in this section is derived from the work done in the FOT-Net Data project (fot-net.eu) for sharing data from field operational tests. The results of this work are described in the Data Sharing Framework [18].

AUTOPILOT is a large-scale field operational test project with a great focus on the use of IoT technology and devices for autonomous driving. It generates raw data similar to those from the previous FOT/NDS projects. Therefore, as far as data sharing is concerned, AUTOPILOT should refer to the guidelines developed in the aforementioned projects as a start point and also take into account the regulation for sharing data generated and possibly stored on the IoT devices or platforms[19,20].

As a summary, the FOT Data Sharing Framework (FOT-DSF) consists of seven elements as briefly described as follows.

Project agreements

- Project agreement content, including guidelines and checklists to incorporate the prerequisites for data sharing in the agreements, which together with legal and ethical constraints form the conditions for data sharing. There are several agreements in AUTOPILOT which can be used to facilitate data sharing. The project agreements including the grant agreement (together with the description of the work), the consortium agreement, the participant agreement and external data provider agreements.

Data & metadata descriptions

- To facilitate the understanding of the context in which the data was collected and the validity of the data. These include a suggested standard for the documentation of the data and metadata, divided into 5 categories: AUTOPILOT study design and execution documentation, descriptive metadata (e.g., how the data is calculated), data (e.g., sampling frequency), structural metadata (e.g., how the data is organised) and administrative metadata (e.g., access procedures).

Data protection

- This element consists of security procedures and requirements at both the data provider and analysis sites, including detailed implementation guidelines, to ensure that the personal and confidential data issues in AUTOPILOT are adequately considered and appropriate actions are taken.

Training on data protection

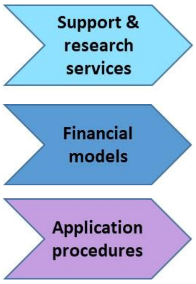
- This element is concerned with security and human subject protection training for all involved personnel. Issues include who should be trained and when, what content should be part of the training (including detailed suggestions), how to do the training, and how to document it.

---

18 http://fot-net.eu/Documents/data-sharing-framework/

19 Internet of Things: privacy and security in a connected world / US Federal Trade Commission, Staff report, 2015.

20 Regulating the Internet of Things: first steps toward managing discrimination, privacy, security and consent / S. Peppet, Texas Law Review, v. 93, n.1, p.85-176, 2014.



- To identify and select tools for data processing, analysis and interpretation and facilitate the performance of research tasks.
- To provide funding for the data to be maintained and available, and data access services. Eight financial models are discussed and a list of data management costs is provided.
- To provide detailed content lists to address when developing application procedures and data application forms.

In Annex 8.4 the text is given from the Data Sharing Framework on working with participants from the general public (including an example consent form) and with external data providers.

For more detailed information and recommendations we refer to the Data Sharing Framework[21].

[21] http://fot-net.eu/Documents/d3-1-data-sharing-framework/

7 Conclusion

This deliverable starts the work done in task 3.4 “Test Data Management”. This deliverable “Data collection and Integration methodology” describes the methodologies that will be followed by pilot sites and data providers. These methodologies ensure that data required for the evaluation are generated according to precise requirements and made easily available for technical evaluation, user acceptance, the impact on the quality of life, and the business impact.

This deliverable provides the starting point for how test data management will be defined by pilot sites and implemented in AUTOPILOT. The next deliverable D3.7 will provide the architecture and the solutions that will be implemented and which guarantee the data provisioning of the right data for the right user.

The developed methodology is based on requirements provided by the Task 4.1 which has adopted and implemented the FESTA methodology. The task 3.1 applies FESTA methodology by describing the use cases, the test scenarios and storyboards with and without Internet of things and the data collection means associated to AUTOPILOT subsystems (IoT devices and platforms, communication, automated driving functions and services).

The data collection methodology will be refined further from the practical knowledge and experience gained from the first pilot phases and evaluations in the next two project years.

This deliverable defines the distributed data management and the centralised data management. The distributed data management is described for each pilot site and shows how pilot sites are organised to data management 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 data sources that will be generated by pilot sites in section 3. The detailed specifications are provided in the form of living documents in appendix that will be updated throughout the data collection phase.

The data collection methodology was created in a top-down and bottom-up approach. In the bottom-up approach, pilot sites describe their existing assets, tools and facilities and their data collection environments. In the top-down approach, additional workshops and discussions with pilot 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 was to find the most common approach to data collection, data provisioning, metadata management and data sharing for all pilot sites.

Concerning the IoT aspect in AUTOPILOT, it is important to note that the IoT data model standardisation work carried out by the AUTOPILOT data modelling activity group (DMAG) is still in progress. The standardisation scopes handle only the IoT messages used for exchanging information and instructions between devices and platforms. This standardisation work is based on different standards (e.g., SENSORIS, DATEX II) and will allow using the same model across all pilot sites. The data models will be specified after the initial feedback from all partners and will then be implemented.

8 Annexes

8.1 Annex 1 – Inventory of tools used by pilot sites for data management

Table 8 – Inventory of tools for data management

Pilot Site Name	Name of Tools	Category	Description
VERSAILLES	**Sensinov IoT Platform**	Data collection	IoT platform based on OneM2M standard
	RTMaps	Data record	Toolkit for multimodal applications
	Tshark	Data logging	Open source tool for dump and analyse network traffic
BRAINPORT	**CommonLogFormat**	Data log formats	
	PostgreSQL	Data Storage	Opensource Database with PostGIS
	ADA	Data Storage and Data Analysis	Repository and analyses tools to automatically process log data in the common log data format
LIVORNO	**Ocean Platform**	Data collection	Iot platform based on OneM2M standard
	MariaDB	Data Storage	Opensource Database
	Protobuf	Data logging and serialisation	Open source language-neutral, platform-neutral toolkit for serialising structured data.
	Vsftpd	Data storage	Open source secure data transfer server
	Wireshark	Data logging	Open source packet analyser
VIGO	**CTAG Data logger**	Data acquisition	Software component used to record data
TAMPERE	**VTT Data Logger**	Data collection	Software component to record and collect data made available on DDS

8.2 Annex 2 – Data collection template for data collection

D3.6 Data Collection Template.xlsx

8.3 Annex 3 – Data collection templates filled by Pilot sites

Table 9 – Data collection templates per pilot site

Pilot Site Name	Data Collection Spreadsheet
VERSAILLES	D3.6 Data collection VERSAILLES.xlsx
TAMPERE	D3.6 data collection TAMPERE.xlsx
LIVORNO	D3.6 data collection LIVORNO.xlsx
VIGO	D3.6 data collection VIGO.xlsx
BRAINPORT	D3.6 Data Collection BRAINPORT.xlsx

8.4 Annex 4 – Data sharing: participants from the general public and external data provider agreements

In the FOT-Net Data Sharing Framework ((http://fot-net.eu/Documents/d3-1-data-sharing-framework/) the following recommendations are given for working with participants from the general public and with external data providers.

In the case of working with participants from the general public, it is necessary to get an informed consent from them, taking into account options for further sharing of data. From a data-sharing standpoint, it is especially important to describe:

- *what data are collected;*
- *where the data will be stored and who is responsible for the data;*
- *who (project partners/third parties) will have access to what data and on what conditions, during and potentially after the project;*
- *an overview of the access procedures;*
- *how anonymity will be ensured;*
- *the responses to the three YES/NO options below, directly related to data sharing.*

It is recommended that the participant actively consent to these vital aspects of data sharing. Example text, which needs to be adapted to adhere to specific national regulations, is provided here for European conditions:

I hereby agree to participate in the above-described research study. I consent to having the material transferred and shared with research partners in a third country (e.g., a country outside EES).

☐ *Yes* ☐ *No*

I also consent to video recordings or pictures being published or shown in public events (e.g., research reports or conferences).

☐ *Yes* ☐ *No*

I also consent to the collected data (including video recordings and pictures) being re-used in other research projects by research partners/third parties, focusing on factors regarding:

- *the driver; and/or*
- *the vehicle);and/or*
- *the traffic environment; and/or*
- *… (to be completed by the specific pilot site)*

☐ *Yes* ☐ *No*

For external data provider agreements:

External data providers could be companies providing sensor systems, map data, weather data or other services that the project needs to enhance the dataset. Contracts and NDAs should be signed. It is important to be aware of topics that can affect future research due to possible restrictions in data use. Attention from a data-sharing perspective should be given to answering the following questions:

- *What is regarded as confidential information and what can be shared?*
- *Can confidential data be anonymised/changed/aggregated, to allow for more open access?*
- *Can the data be accessed by another project partner/third party?*
- *Can the data be transferred to another project partner/third party?*
- *Are there restrictions on what the data can be used for?*
- *Are there special conditions for sharing and re-using the data after the project?*
- *What happens if the external data provider is bought by another company?*

For more detailed information and recommendations we refer to the Data Sharing Framework (http://fot-net.eu/Documents/d3-1-data-sharing-framework/).

References

[1] D4.1 Methodology for evaluation, version 1.0, 31 January 2018
[2] D3.1 Initial pilot test specifications, version 2.0, 31 December 2017
[3] Alliance for Internet of Things Innovation (AIOTI): https://aioti.eu
[4] ARM-IoT: https://developer.arm.com/products/architecture/system-architecture
[5] IR2.3 Report on the implementation of the IoT Platform, version 1.1, 28 February 2018
[6] https://www.google.com/forms/about/
[7] https://www.soscisurvey.de
[8] https://www.ibm.com/analytics/data-science/predictive-analytics/spss-statistical-software
[9] https://www.surveymonkey.com
[10] http://fot-net.eu
[11] http://fot-net.eu/Documents/d3-1-data-sharing-framework/
[12] http://connectedautomateddriving.eu
[13] http://fot-net.eu/Documents/fot-net-data-final-deliverables/
[14] http://www.dcc.ac.uk/resources/metadata-standards/list
[15] http://www.ogimet.com
[16] http://www.onem2m.org/
[17] http://www.iotocean.org/
[18] http://fot-net.eu/Documents/data-sharing-framework/
[19] Internet of Things: privacy and security in a connected world / US Federal Trade Commission, Staff report, 2015.
[20] Regulating the Internet of Things: first steps toward managing discrimination, privacy, security and consent / S. Peppet, Texas Law Review, v. 93, n.1, p.85-176, 2014.
[21] http://fot-net.eu/Documents/d3-1-data-sharing-framework/