Grant Agreement Number: 731993

Project acronym: AUTOPILOT

Project full title: Automated driving Progressed by Internet of Things

D2.3

Report on the Implementation of the IoT Platform

Due delivery date: 30 June 2018

Actual delivery date: 02 July 2018

Organisation name of lead participant for this deliverable: IBM Research Ireland

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:	D2.3
Deliverable responsible:	Yassine Lassoued, IBM Research, Ireland
Workpackage:	WP2
Editor:	Yassine Lassoued, IBM Research, Ireland

Author(s) – in alphabetical order		
Name	**Organisation**	**E-mail**
Alexander Velizhev	IBM RE	ave@zurich.ibm.com
Bram van den Ende	TNO	bram.vandenende@tno.nl
Daan Ravesteijn	TNO	daan.ravesteijn@tno.nl
Daniele Brevi	ISMB	brevi@ismb.it
David Martin	Gemalto	Martin.David@gemalto.com
Fabrizio Gatti	Telecom Italia	fabrizio1.gatti@telecomitalia.it
Georgios Karagiannis	Huawei	georgios.karagiannis@huawei.com
Gurkan Solmaz	NEC	gurkan.solmaz@neclab.eu
Jan Bosma	Technolution	jan.bosma@technolution.nl
Jean-Francois Simeon	Continental	Jean-Francois.Simeon@continental-corporation.com
Johan Scholliers	VTT	Johan.Scholliers@vtt.fi
Lorenzo Viola	Huawei	lorenzo.viola.ext@huawei.com
Louis Touko Tcheumadjeu	DLR	Louis.ToukoTcheumadjeu@dlr.de
Mahdi ben Alaya	Sensinov	benalaya@sensinov.com
Mariano Falcitelli	CNIT	Mariano.Falcitelli@cnit.it
Paolo Scalambro	Telecom Italia	paolo.scalambro@telecomitalia.it
Roberto Gavazzi	Telecom Italia	roberto.gavazzi@telecomitalia.it
Silvia Alén González	CTAG	silvia.alen@ctag.com
Xurxo Legaspi	CTAG	xurxo.legaspi@ctag.com
Yassine Lassoued	IBM IE	ylassoue@ie.ibm.com

Document Revision History			
Version	**Date**	**Modifications Introduced**	
		Modification Reason	**Modified by**
V0.1	27/04/2018	Initial draft based on IR2.3	All authors
V0.2	18/04/2018	Final draft before review	Mariano Falcitelli, Gurkan Solmaz, Bram van den Ende, Jan Bosma, Yassine Lassoued
V0.3	25/06/2018	Reviewed version	All authors

Abstract
This document represents D2.3, "Report on the implementation of the IoT platform". D2.3 describes the status of the implementation of the open IoT platform for autonomous driving of the AUTOPILOT project. The report covers all the project pilot sites and use cases and provides the target architecture under implementation for each of these.

Legal Disclaimer

Abbreviations and Acronyms

Acronym	Definition
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
ACP	Access Control Policy
AD	Autonomous Driving
ADN	Application Dedicated Node
AE	Application Entity
API	Application Programming Interface API
ASAR	serviceSubscribedAppRule
ASN	Application Service Node
ASN-CSE	Application Service Node – Common Services Entity
AVP	Automated Valet Parking
BLE	Bluetooth Low Energy
CAM	Cooperative Awareness Message
C-ITS	Cooperative Intelligent Transport Systems
CAN	Controller Area Network
CeH	Connected eHorizon
CEMA	Crowd Estimation and Mobility Analytics
CoAP	Constrained Application Protocol
CSE	Common Services Entity
CSEBase	Common Services Entity Base
DENM	Decentralized Environmental Notification Message
DMAG	Data Modelling Activity Group
DSRC	Dedicated Short-Range Communications
EC	European Commission
ETSI	European Telecommunications Standards Institute
GE	Generic Enabler
HMI	Human-Machine Interface
IEEE	Institute of Electrical and Electronics Engineers
IN-CSE	Infrastructure Node – Common Services Entity
IoT	Internet of Things
ITS	Intelligent Transport Systems
JSON	JavaScript Object Notation
LAN	Local Area Network

LTE	Long-Term Evolution
LWM2M	Lightweight Machine to Machine
M2M	Machine to Machine
MAP	Map Data
Mca	Reference Point for M2M Communication with AE
Mcc	Reference Point for M2M Communication with CSE
Mcc'	Reference Point for M2M Communication with CSE of different M2M Service Provider
MEC	Multi-Access Edge Computing
MICE	Mission Control Enabler
MMG	Morphing Mediation Gateway
MN-CSE	Infrastructure Node – Common Services Entity
MQTT	OASIS Message Queuing Telemetry Transport
NB-IoT	Narrowband IoT
NGSI	Next Generation Services Interface
NoSQL	Not only SQL
OBU	On-Board Unit
OEM	Original Equipment Manufacturer
OSGi	Open Service Gateway Initiative
OWL	Web Ontology Language
PaaS	Platform as a Service
PoA	Point of Access
PS	Pilot Site
RDF	Resource Description Framework
RemoteCSE	Remote Common Service Entity
REST	Representational State Transfer
RSU	Road Side Unit
SMG	Semantic Mediation Gateway ()" or "
SP	Service Provider
SPAT	Signal Phase and Time
SSL	Secure Sockets Layer
SQL	Structured query language
TCC	Traffic Control Centre
TCP	Transmission Control Protocol

TLS	Transport Layer Security
TMC	Traffic Management Centre
UD	Urban Driving
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
UWB	Ultra-Wide Band
VDH	Vehicle Data Hub
WP	Work Package
VRU	Vulnerable Road User
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network
XML	Extensible Markup Language

Table of Contents

EXECUTIVE SUMMARY 11

1 INTRODUCTION 12

1.1.1 Purpose of the document 12

1.1.2 Intended audience 12

2 OVERVIEW OF THE IOT ARCHITECTURE IMPLEMENTATION 13

2.1.1 Target Architecture 13

3 IMPLEMENTED IOT PLATFORMS 16

3.1 Watson IoT Platform™ 16

3.1.1 Overview of Watson IoT Platform™ 16

3.1.2 Devices 16

3.1.3 Gateways 17

3.1.4 Applications 17

3.1.5 Events 17

3.1.6 Commands 18

3.1.7 Communication Protocol 18

3.2 Watson IoT Platform™ Deployment and Access 18

3.2.1 Interfacing with Watson IoT Platform™ 19

3.3 Watson IoT for Automotive 19

3.3.1 Overview of Watson IoT for Automotive 19

3.3.2 Watson IoT for Automotive Deployment and Access 20

3.4 FIWARE 21

3.4.1 FIWARE Architecture 22

3.4.2 IoT Broker GE reference implementation: AERON 22

3.4.3 IoT Discovery GE alternative implementation: NEC ConfMan 23

3.4.4 Crowd Estimation Service & Mission Control 23

3.4.5 FIWARE Status of Deployment and Access 24

3.4.6 Future work 24
3.5 HUAWEI OceanConnect Platform 25
3.5.1 Huawei OceanConnect IoT Platform Architecture and Capabilities 25
3.5.2 Connecting Applications and Devices to OceanConnect 26
3.5.3 Interacting with OceanConnect: Open APIs 27
3.5.4 OceanConnect Deployment and Access 28
3.6 TIM IoT Platform 28
3.6.1 Overview of the TIM IoT Platform 28
3.6.2 TIM oneM2M Platform Deployment and Access 29
3.6.3 Interfacing with the TIM oneM2M Platform 30
3.7 SENSINOV oneM2M Platform 30
3.7.1 SENSINOV oneM2M Platform Deployment and Access 31
3.8 Technolution MobiMaestro Platform 31
3.8.1 Overview of the MobiMaestro Architecture 31
3.8.2 MobiMaestro Platform Deployment and Access 32
3.9 Future Work 33
4 PILOT SITE IOT ECOSYSTEMS 34
4.1 Finland (Tampere) 34
4.1.1 Overall IoT Architecture of the Finnish Pilot Site 34
4.1.2 IoT Ecosystem Implementation in the Finnish Pilot Site 35
4.2 France (Versailles) 36
4.2.1 Overall IoT Architecture of the French Pilot Site 36
4.2.2 IoT Ecosystem Implementation in the French Pilot Site 36
4.3 Italy (Florence-Livorno) 37
4.3.1 Overall IoT Architecture of the Italian Pilot Site 37
4.3.2 IoT Ecosystem Implementation in the Italian Pilot Site 38
4.3.3 IoT Platform and IoT Devices Integration 41
4.4 Korea 42

4.5 Netherlands (Brainport) 42
4.5.2 IoT Ecosystem Implementation in Brainport 43
4.6 Spain (Vigo) 44
4.6.1 Overall IoT Architecture of the Spanish Pilot Site 45
4.6.2 IoT Ecosystem Implementation in the Spanish Pilot Site 45
5 INTEROPERABILITY 48
5.1 oneM2M Interoperability Platform and Interworking Gateways 48
5.2 Standardised Data Models 49
5.3 Standardised Ontologies 49
6 COMMON IOT DATA MODEL 50
6.1 Scope of the Work 50
6.2 Use Case IoT Data Requirements 50
6.3 Common IoT Data Model 50
6.3.1 Vehicle Package 51
6.3.2 Parking Package 52
6.4 Future Work 53
7 ONTOLOGIES 54
8 REFERENCES 55

List of Figures

Figure 1 – AUTOPILOT IoT Architecture: Functional View 13
Figure 2 – Layers of the Federated IoT Architecture 14
Figure 3 – Autopilot Federated IoT Architecture 15
Figure 4 – Architecture of the Watson IoT Platform™ 16
Figure 5 – Architecture of the Watson IoT for Automotive 20
Figure 6 – Deployment Architecture for Watson IoT Platform™ and IoT for Automotive 21
Figure 7 – FIWARE IoT Architecture 22
Figure 8 – HUAWEI OceanConnect IoT Platform Architecture 26
Figure 9 – Overview of Huawei OceanConnect IoT Platform open APIs 27
Figure 10 – High-Level Architecture of the TIM oneM2M Platform 29
Figure 11 – High-Level Architecture of the Technolution MobiMaestro Platform 32
Figure 12 –Technolution MobiMaestro Target Architecture Platform 32
Figure 13 – Finnish Pilot Site Overall Architecture 34
Figure 14 – French Pilot Site Overall Architecture 36
Figure 15 – Livorno Pilot Site Overall Architecture 38
Figure 16 – Integration of IoT Devices into the oneM2M IoT Platform in the Italian Pilot Site 41
Figure 17 – Integration of IoT Devices into the oneM2M IoT Platform in the Dutch Pilot Site 43
Figure 18 – Spanish Pilot Site Overall Architecture 45
Figure 19 – Autopilot Federated IoT Architecture (Copied from Chapter 3). 48
Figure 20 – Overview of the AUTOPILOT IoT Data Model Packages 51
Figure 21 – Overview of the SENSORIS Message Elements 52
Figure 22 – DATEX II Parking Extension – High-Level Data Elements 53

List of Tables

Table 1 – Watson IoT Platform – Deployment Status and Access 18
Table 2 – Watson IoT for Automotive – Deployment Status and Access 21
Table 3 – Web Links to IoT Broker Artifacts 23
Table 4 – FIWARE – Deployment Status and Access 24
Table 5 – HUAWEI OceanConnect Platform – Deployment Status and Access 28
Table 6 – TIM oneM2M Platform – Deployment Status and Access 29
Table 7 – Interfacing with the TIM oneM2M Platform 30
Table 8 – SENSINOV oneM2M Platform – Deployment Status and Access 31
Table 9 – MobiMaestro – Deployment Status and Access 33
Table 10 – AUTOPILOT Use Case IoT Messages Selected for Standardisation 50
Table 11 – IoT Data Model Package Lead Partners 51

Executive Summary

This deliverable, D2.3, reports the work being undertaken as part of the AUTOPILOT task T2.3, "Development of the Open IoT Service Platform".

The AUTOPILOT IoT service platform is a federation of several IoT platforms, allowing it to be **open** and **flexible**. An open oneM2M standard IoT platform, referred to as the *oneM2M interoperability platform*, interconnects these *proprietary* IoT platforms provided by the project partners. The proprietary IoT platforms collect data from connected devices. They exchange IoT data and events with the interoperability platform through oneM2M interworking proxies.

Currently, several IoT platforms have been deployed for the pilot sites:

- FIWARE IoT platform, used in the Dutch pilot site
- Watson IoT Platform, used in the Dutch and Spanish pilot sites
- HUAWEI OceanConnect IoT platform, used in the Dutch pilot site
- TIM oneM2M IoT platform, used in the Italian pilot site
- SENSINOV oneM2M platform, used in the Finnish, French and Dutch pilot sites

oneM2M Interworking proxies are currently being developed for the non-oneM2M-compliant platforms.

Across the pilot sites, devices are currently being connected to the IoT platforms. The AD functionality is being adapted to support IoT data for all the project use cases.

To facilitate interoperability between the IoT platforms and to make the use cases pilot-site-independent, a Data Modelling Activity Group (DMAG) is specifying a common data model for the whole project, based on existing standards: SENSORIS for vehicle messages and detection events and DATEX II for parking and traffic information.

Ontologies may be used in AUTOPILOT to define controlled vocabularies and semantic mappings for some of the data model field values. This would allow for flexibility and openness, while facilitating interoperability. Several ontologies have been reviewed so far, but the actual work on developing the ontologies has not started yet. It is planned to start in the third quarter of 2018.

1 Introduction

1.1.1 Purpose of the document

This deliverable, D2.3, reports the work being undertaken as part of the AUTOPILOT task T2.3, "Development of the Open IoT Service Platform".

The document provides an overview of the AUTOPILOT open IoT architecture for autonomous driving, currently under development in six pilot sites in Finland, France, Italy, South Korea, the Netherlands and Spain.

An overview of the deployed IoT platforms, their capabilities and status of deployment, purposes and access instructions are provided.

The document also provides an overview of the implementation of the IoT ecosystem in the pilot sites.

The last two chapters of the document provide an overview of the work being carried out to specify common data models and ontologies across the pilot sites and use cases.

1.1.2 Intended audience

This deliverable is a public report intended for both internal use by the AUTOPILOT partners and external use.

2 Overview of the IoT Architecture Implementation

This chapter provides an overview on the initial target IoT architecture of AUTOPILOT, which has been specified as part of task 1.2 “IoT Architecture and Specification” and which is currently under development.

2.1.1 Target Architecture

The AUTOPILOT IoT architecture builds on and borrows building blocks from, relevant IoT architectures such as AIOTI[1] and IoT-ARM[2]. The development of the IoT architecture follows an incremental approach.

The AUTOPILOT target IoT 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.

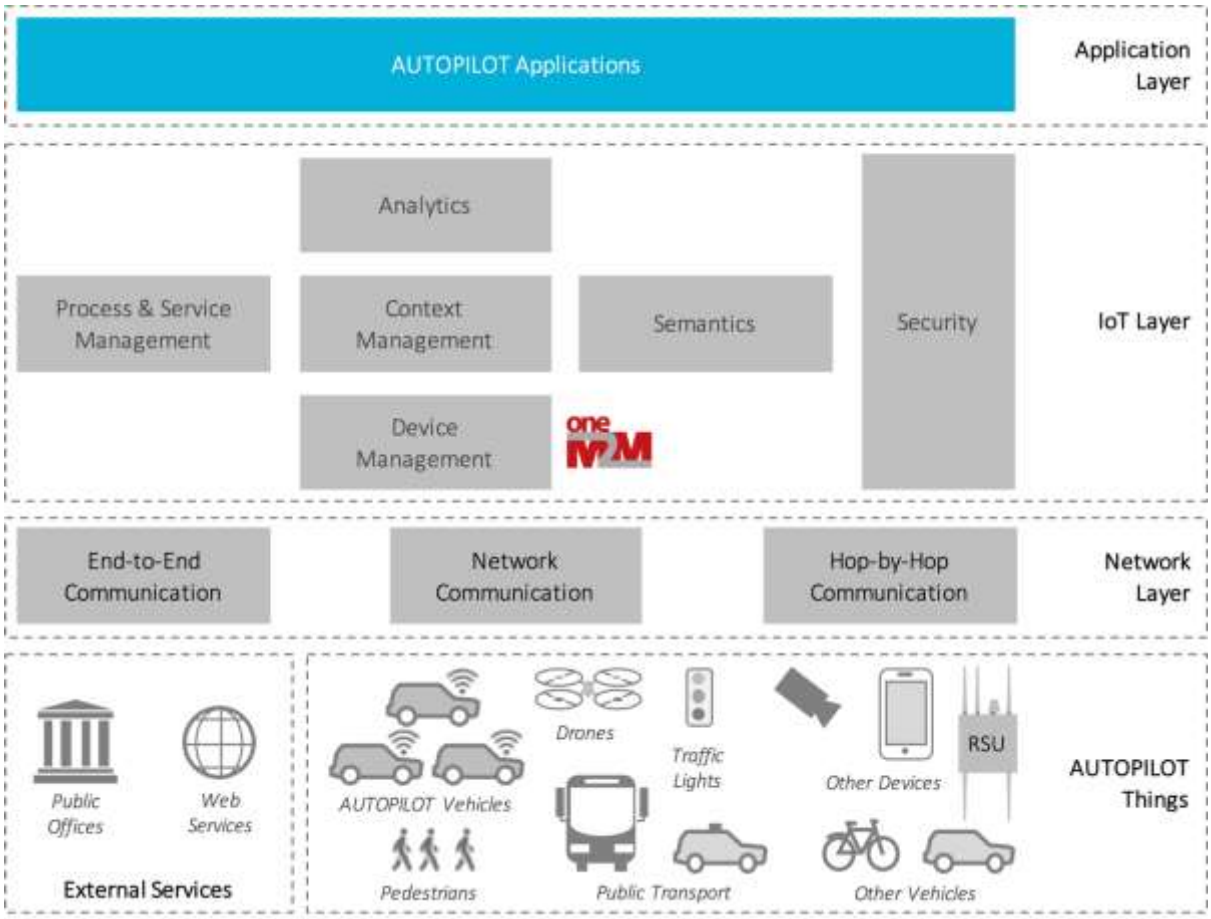

Figure 1 – AUTOPILOT IoT Architecture: Functional View

As shown in Figure 1, the AUTOPILOT IoT architecture has four main functional layers:

1. **Things Layer:** Includes the AUTOPILOT "things" (vehicles, cameras, drones, road side units, etc.) and external services provided by public offices or private web services.
2. **Network Layer:** Enables communication throughout the IoT ecosystem.

[1] Alliance for Internet of Things Innovation (AIOTI): https://aioti.eu

[2] ARM-IoT: https://developer.arm.com/products/architecture/system-architecture

3. **IoT Layer:** Enables the IoT functionality through a set of IoT building blocks: *device management, context management, process & service management, semantics, analytics and security.* Each of these functional building blocks is specified in detail in the AUTOPILOT deliverable D1.3.
4. **Application Layer:** Contains services that leverage the AUTOPILOT IoT. In AUTOPILOT, this includes services provided by the use cases to the AD vehicles and users (e.g. drivers, car sharing customers, etc.).

Given 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.

Figure 2 illustrates the AUTOPILOT **federated** IoT platform architecture. The term "federated", in this context, means that there are several layers of IoT platforms deployed on a variety of physical infrastructures starting from the in-vehicle IoT layer to the top-level internet cloud-based IoT platform.

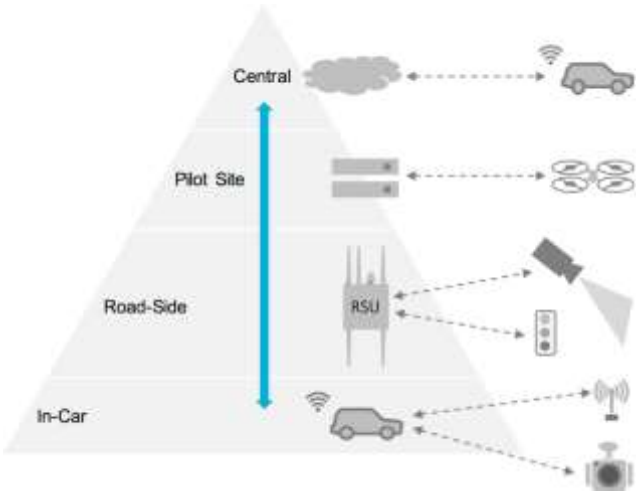

Figure 2 – Layers of the Federated IoT Architecture

In this architecture, data may flow from any level to any level as required by the use cases. At a given level, data may be processed to generate new information that may be published to an IoT platform at another layer. For example, image data submitted by a dash camera to the in-vehicle IoT platform, may be processed inside the vehicle and upon detection of an object or hazard on the road, a message may be generated and submitted to the road-side and/or cloud IoT platforms.

Figure 3 shows the AUTOPILOT target IoT architecture. As shown in this diagram, devices, gateways and in-vehicle and road-side IoT platforms exchange information (e.g. about detected objects, hazards, vulnerable road users, traffic lights, vehicle updates, etc.) with several distributed cloud IoT platforms. We distinguish the following two types of cloud IoT platforms:

1. **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. These are introduced in chapter 3.

2. **oneM2M Interoperability Platform:** This is the central IoT platform for exchanging IoT messages relevant to all autonomous driving (AD) vehicles.

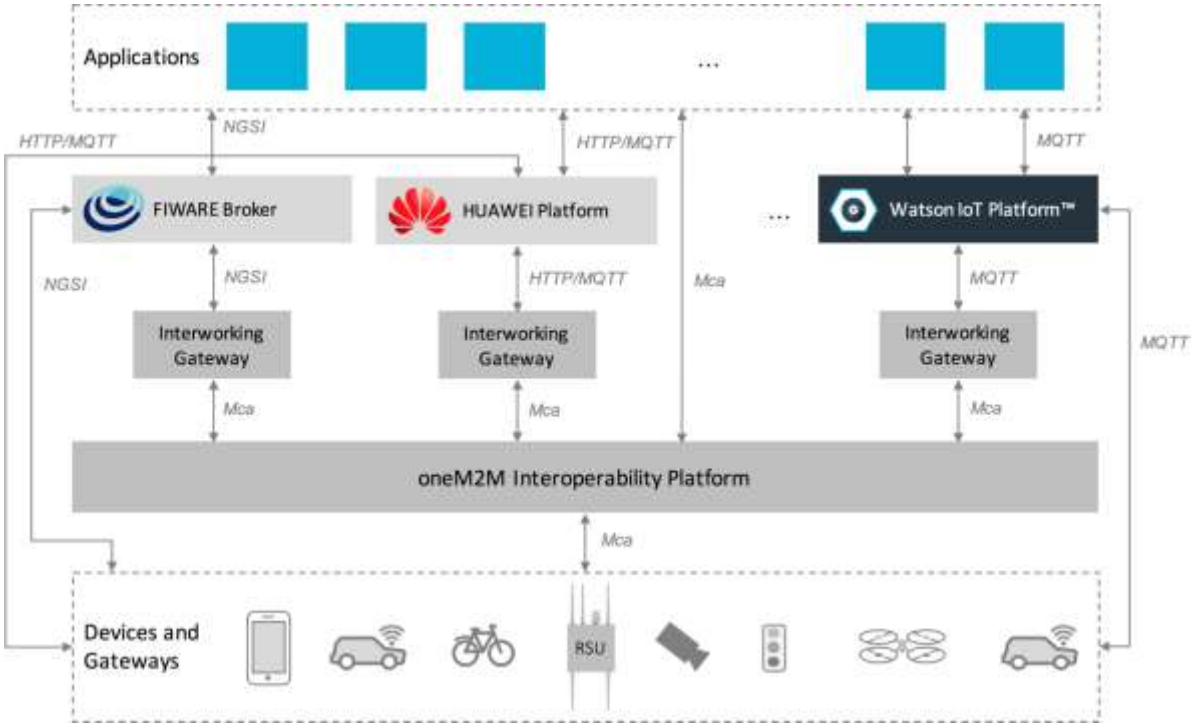

Figure 3 – Autopilot Federated IoT Architecture

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 Implemented IoT Platforms

This chapter provides an overview of the deployed IoT platforms, their components, features and intended use. Links to the deployed platform instances, their documentation, source code (if applicable), client-side code and instructions on how to access the IoT platforms are also provided when applicable.

This chapter reports on the following types of activities:

- Deploying, customising and configuring the project IoT platforms;
- Connecting devices to the IoT platforms and testing their connectivity;
- Developing and deploying the oneM2M interworking proxies to connect the proprietary pilot site IoT platforms to the central oneM2M interoperability platform.

Other work related to designing and implementing the common IoT exchange data models is reported in Chapter 6.

3.1 Watson IoT Platform™

This section provides an overview on the IBM Watson IoT Platform and its deployment in AUTOPILOT.

3.1.1 Overview of Watson IoT Platform™

This section provides an overview of the IBM Watson IoT Platform™ and its key concepts. Further details are provided in the official documentation of the platform [IBM18a].

Watson IoT Platform™ (cf. Figure 4) is an IoT broker and device manager, which can be connected to a wide variety of devices, gateways and applications. It allows devices and applications to publish and subscribe to data. The platform provides secure communication to and from any devices, using MQTT and TLS. It also provides a dashboard for monitoring the devices.

Device events published on Watson IoT platform™ may be pushed to a database for storage. By default, Watson IoT Platform™ supports the IBM Cloudant database [IBM18b].

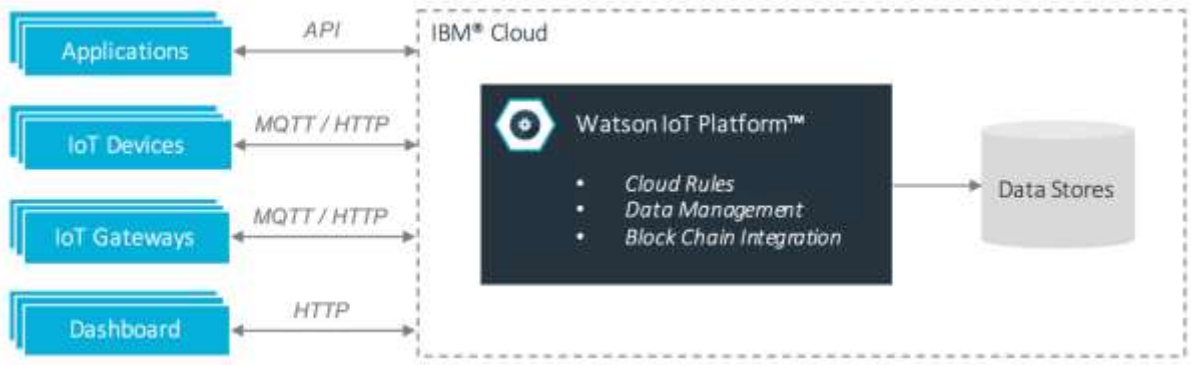

Figure 4 – Architecture of the Watson IoT Platform™

The following subsections introduce the key concepts of Watson IoT Platform™.

3.1.2 Devices

A *device* can be anything that has a connection to the Internet and that can push data into the cloud. However, devices cannot communicate directly with other devices, instead devices accept commands from applications and send events to applications. Devices in the Watson IoT Platform™ are identified by a unique authentication token. They **must be registered** before they can connect to

the Watson IoT Platform™. The Watson IoT Platform™ recognises two classes of device: *managed devices* and *unmanaged devices*.

Managed devices are defined as devices that contain a device management agent. A device management agent is a set of logics that allows the device to interact with the Watson IoT Platform™ Device Management service by using the Device Management Protocol. Managed devices can perform device management operations, including location updates, firmware downloads and updates, reboots and factory resets. Management operation may be triggered through the main Watson IoT Platform™ dashboard or the device management API. Device management can also be extended to include custom device management actions.

Unmanaged devices are all devices without a device management agent. They may connect to the Watson IoT Platform™ and send and receive events and commands, but they cannot send device management requests or perform device management operations.

3.1.3 Gateways

Gateways are specialised devices that have the combined capabilities of an application and a device, which allows them to serve as access points for other devices. Devices that cannot connect directly to the Internet can access the Watson IoT Platform™ service by first connecting to the gateway device. Gateways have all the functions of a device, but can also publish and subscribe on behalf of the devices connected to them.

Examples of gateways are:

- A software in a vehicle that collects data from the vehicle sensors and publishes them to Watson IoT Platform™;
- A server that receives image data from car park cameras or drones detects parking spot status and publishes the results to Watson IoT Platform™.

3.1.4 Applications

An *application* is anything that has a connection to the Internet and interacts with data from devices and controls the behaviour of those devices. Applications identify themselves with the Watson IoT Platform™ by using an API key and a unique application ID. Unlike devices, individual applications do not need to register before they can connect to the Watson IoT Platform™. However, they **must use a valid API key** that has been previously registered.

Examples of applications are:

- The car sharing service, which listens to traffic and environmental events published by devices and gateway through Watson IoT Platform™ and schedules, accordingly, pick-ups, drop-offs and vehicle routes;
- The AVP service, which listens to events on the car park and to the positions and status of vehicles being parked to assign parking spots and routes to the parking spots;
- The autonomous driving (AD) functionality in an AD vehicle, which listens to external events shared through Watson IoT Platform™.

3.1.5 Events

Events are the mechanism by which devices publish data to the Watson IoT Platform™. Devices control the content of their messages and assign a name for each event that is sent. The Watson IoT Platform™ uses the credentials that are attached to each event received to determine which device sent the event. This architecture prevents devices from impersonating one another. Applications can process events in real time and see the source of the event and the data contained in the event. Applications must be configured to define which devices and events they subscribe to.

Examples of events are:

- Car probe data submitted by vehicle gateway;
- Identification of an object, obstacle, vulnerable road user (VRU), or hazard on the road;
- Parking spot and occupancy detection.

3.1.6 Commands

Commands are the mechanism by which applications communicate with devices. Only applications can send commands and the commands are sent to specific devices. The device must determine which action to take on receipt of any given command. Devices can be designed to listen for any command or to subscribe to a specified list of commands.

3.1.7 Communication Protocol

IBM Watson IoT Platform™ supports version 3.1.1 of the MQTT messaging protocol [MQTT]. It accepts any content that is permitted by the MQTT standard. MQTT is data-agnostic, so, in theory, it is possible to send images, texts that are in any encoding, encrypted data and virtually every type of data in binary format.

The IBM open source Watson Developer Cloud APIs (https://github.com/watson-developer-cloud) provide client APIs (https://github.com/ibm-watson-iot) for Watson IoT Platform™ in various programming languages (Java, NodeJS and Python). These APIs may be used to facilitate interaction with Watson IoT Platform™.

3.2 Watson IoT Platform™ Deployment and Access

IBM is providing two Watson IoT Platform™ instances for AUTOPILOT:

1. Development: This instance should be used as a starting point for learning, development, experimentation and testing purposes. You may want to use this instance to create virtual (fake) devices and test their connection to the platform.
2. Production: This instance should be used for official project demonstrations and for connecting real devices.

Information about the deployment status and access to the above instances is provided in Table 1 below.

The Watson IoT Platform™ instances will be connected to the oneM2M interoperability platform (see section 3.7) through oneM2M connectors, called interworking proxies, developed by SENSINOV and IBM IE. A oneM2M connector is already deployed on the TNO servers for connecting the development Watson IoT Platform to the oneM2M platform.

Table 1 – Watson IoT Platform – Deployment Status and Access

Platform	Watson IoT Platform™		
Hosting	IBM Cloud		
Deployment Status	***Development***	Deployed	Development instance, for learning, development, experimentation and testing purposes
	Production	Deployed	Production instance, for official project demonstrations and for connecting the real devices
	OneM2M Interworking Gateway	Deployed	Bidirectional communication between the oneM2M interoperability platform and Watson IoT Platform. One connector for the development instance is deployed on the TNO servers. This will be moved to the IBM Cloud when firewall issues are sorted out.
Purpose/Pilot Site	Currently used in the Brainport and Vigo pilot sites for car sharing, AVP and parking		

	spot detection	
Standard/Protocol	Watson IoT Platform™ oneM2M connector: deployed on TNO servers	
URL	***Development***	https://2j73n2.internetofthings.ibmcloud.com
	Production	https://btjftu.internetofthings.ibmcloud.com
Connected Devices and Applications	***Development***	Cars/devices/applications collecting and using parking spot availabilities (DLR), car sharing service, parking spot web service, cameras and corresponding applications for vulnerable road user detection (CTAG)
	Production	None
Access Process	The Development and Production Watson IoT Platform™ instances are available to all the pilot sites and use cases. Access to these instances may be requested from IBM IE, through one of the following points of contact: • Yassine Lassoued ylassoue@ie.ibm.com • Anton Dekusar adekusar@ie.ibm.com Any devices and device types to connect to the Watson IoT platforms must be registered in advance by the IBM IE point of contact. You will be requested to provide the device or device type data schemas. Once the devices are registered, you will receive credentials for each device. Device credentials must not be shared with other partners or across devices. Data submitted by one device must comply with the data schema provided for the registration of the device. Any changes to the data schemas must be validated by the IBM IE point of contact before being able to submit data from devices.	
Restrictions	***In addition to the above conditions, please note that personal information, images, or videos, must not be sent to the Watson IBM Platform™ instances.***	

3.2.1 Interfacing with Watson IoT Platform™

Example Java source code to interact with Watson IoT Platform™ is provided in the project GitLab repository **iot-central**, under the folder entitled "**watson**".

3.3 Watson IoT for Automotive

In addition to Watson IoT Platform™, IBM is providing services from Watson IoT for Automotive [IBM18c]: Vehicle Data Hub and Context Mapping.

3.3.1 Overview of Watson IoT for Automotive

Watson IoT for Automotive is an IoT solution for the automotive industry to enable the development of cognitive IoT solutions in areas such as advanced mobility, in-vehicle services, automated and autonomous driving and commercialising vehicle data.

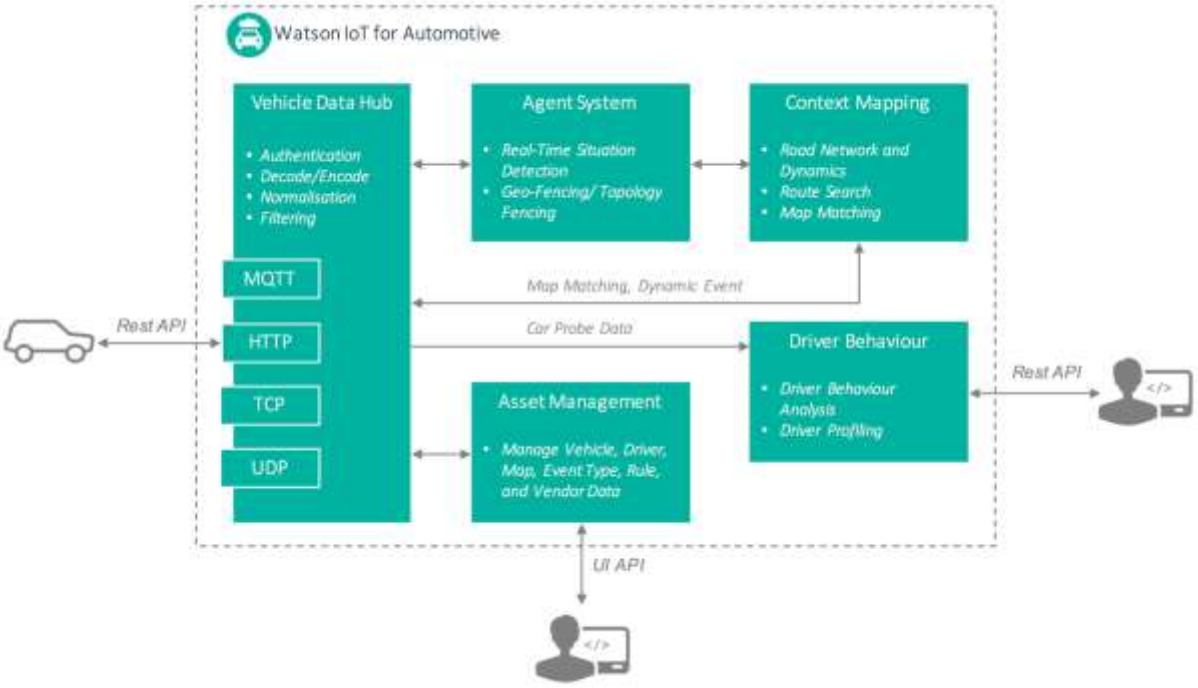

Figure 5 – Architecture of the Watson IoT for Automotive

The IoT for Automotive architecture is illustrated in Figure 5, which shows the following components:

- Vehicle Data Hub (VDH): Front-end component that collects and manages large volumes of vehicle data from connected vehicles and automotive devices by using a range of protocols (MQTT, HTTP, TCP and UDP) and formats.
- Agent System: Detects, stores and manages events that are related to vehicles, driving activity and movement.
- Asset Management System: Manages vehicles, drivers and environmental data about drivers, map sources, geographical areas, analytic rules, events and other things that are meaningful to the connected car ecosystem.
- Context Mapping: Provides geospatial functions, including map matching, road geometry data retrieval, shortest path search for global road networks and real-time traffic event manipulation.
- Driver Behaviour: Analyses driver behaviour data from a connected vehicle or automotive device together with geospatial contextual data.

Documentation of Watson IoT for Automotive is available at [IBM18c]. Developer documentation for Vehicle Data Hub and for Context Matching is available at [IBM18d] and [IBM18e].

3.3.2 Watson IoT for Automotive Deployment and Access

Two components of the Watson IoT for Automotive are currently deployed: Vehicle Data Hub and Context Mapping. The plan for integrating these with Watson IoT Platform™ is illustrated in the architectural diagram of Figure 6.

Figure 6 – Deployment Architecture for Watson IoT Platform™ and IoT for Automotive

As shown in this diagram, vehicle data and events will be pushed from Watson IoT Platform™ to VDH and their positions will be matched to the road network using Context Mapping.

Table 2 – Watson IoT for Automotive – Deployment Status and Access

Platform	Watson IoT for Automotive		
Hosting	IBM Cloud		
Deployment Status	***VDH***	Deployed	Development instance, for learning, development, experimentation and testing purposes
	Context Matching	Deployed	Development instance, for learning, development, experimentation and testing purposes
Purpose/Pilot Site	To be used primarily for the Brainport car sharing use case		
Standard/Protocol	HTTP		
URL	***VDH***	Internal (to be connected to Watson IoT Platform)	
	Context Matching	Internal (to be connected to Watson IoT Platform)	
Connected Devices and Applications	***VDH***	None	
	Map Matching	None	
Access Process	The VDH and Map Matching Instance may be accessed upon request. Access to these instances may be requested from IBM IE, through one of the following points of contact: • Yassine Lassoued ylassoue@ie.ibm.com • Anton Dekusar adekusar@ie.ibm.com Further instructions and conditions regarding access will be provided in due time.		
Restrictions	***In addition to the above conditions, please note that personal information, images, or videos, must not be sent to the Watson IBM for Automotive service instances.***		

3.4 FIWARE

This section provides an overview of FIWARE, its features and deployment status in AUTOPILOT. For detailed information about FIWARE, please refer to the FIWARE wiki [FW].

FIWARE IoT chapter provides the generic enablers (GE) to allow things to become available, searchable, accessible and usable. In this context, "things" mean any physical object, living organism, person or concept interesting from the perspective of an application and whose parameters are totally or partially tied to sensors, actuators or combinations of them.

IoT chapter uses the NGSI standards for data exchange.

- The NGSI-10 interface [NGSI10] is used for exchanging information about entities and their attribute, i.e. attribute values and metadata.
- The NGSI-9 interface [NGSI9] is used for availability information about entities and their attributes. Here, instead of exchanging attribute values, information about which provider can provide certain attribute values is exchanged.

3.4.1 FIWARE Architecture

IoT chapter architecture deployments vary from simple scenarios (e.g. connecting few devices using standard IoT communication protocols to the data chapter Context Broker) to more complex scenarios distributed across a large number IoT networks, connecting IoT gateways and IoT nodes and providing advanced composition and discovery functions.

IoT GEs are spread over two different domains as shown in Figure 7:

1. **IoT Backend:** Comprises the set of functions, logical resources and services hosted in a cloud data centre. Northward, it is connected to the data chapter Context Broker, so IoT resources are translated into NGSI context entities. Southward, the IoT backend is connected to the IoT edge elements, which is all the physical IoT infrastructure.
2. **IoT Edge**: Comprises all on-field IoT infrastructure elements needed to connect physical devices to FIWARE Applications. Typically, the IoT edge comprises the IoT end-nodes, IoT gateways and IoT networks (connectivity). The IoT Edge and its related APIs will facilitate the integration of new types of gateways and devices, which are under definition in many innovative research projects and warranty the openness of FIWARE IoT architectures.

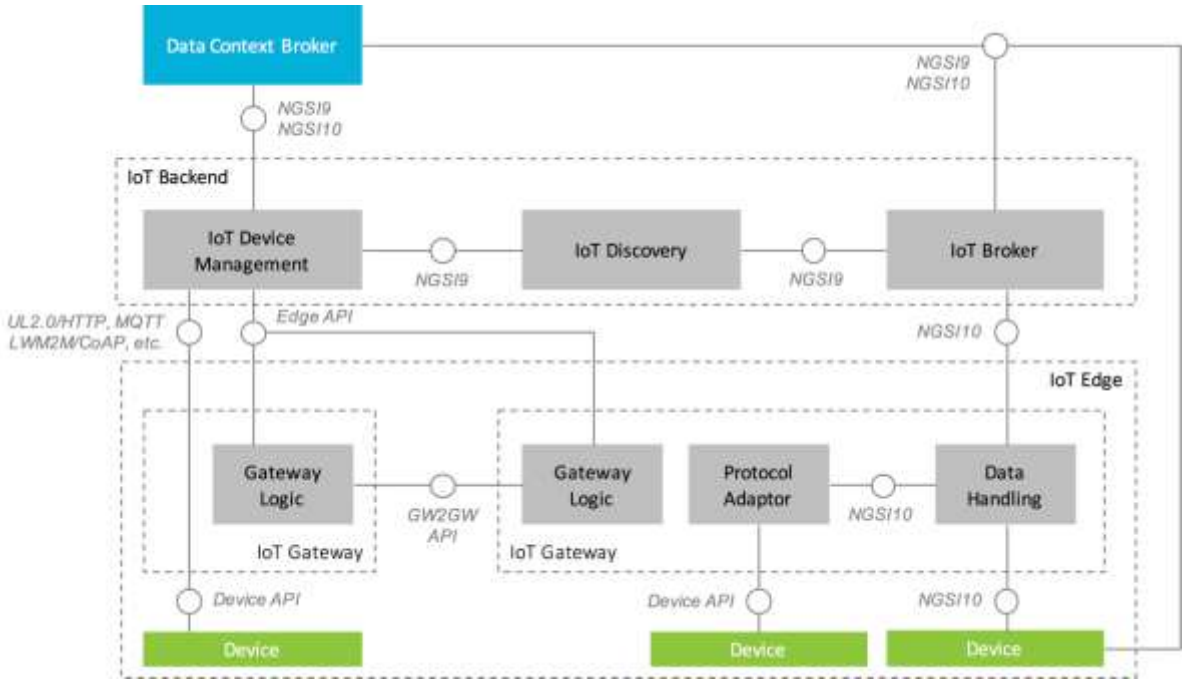

Figure 7 – FIWARE IoT Architecture

3.4.2 IoT Broker GE reference implementation: AERON

The IoT Broker is specified as a lightweight and scalable middleware component that separates IoT applications from the underlying device installations. The IoT Broker implementation available through the FIWARE Catalogue is the reference implementation of this Generic Enabler. This implementation satisfies all properties described in the specification of the Generic Enabler. More details about the IoT Broker GE can be found in the FIWARE wiki [FW].

IoT Broker is integrated with three other components: IoT Discovery (ConfMan), IoT Knowledge Server and FIWARE (Orion) Context Broker. Table 3 provides links to the IoT Broker source code, Docker images and manuals.

Table 3 – Web Links to IoT Broker Artifacts

Artifact	Link
Source Code	https://github.com/Aeronbroker/Aeron
Binaries from FIWARE Catalogue	https://catalogue.fiware.org/enablers/iot-broker
Docker Image	https://hub.docker.com/r/fiware/iotbroker/
Documentation	http://fiware-iot-broker.readthedocs.io/en/master/ https://forge.fiware.org/plugins/mediawiki/wiki/fiware
Online Course	https://edu.fiware.org/course/view.php?id=33

3.4.3 IoT Discovery GE alternative implementation: NEC ConfMan

The NEC Configuration Management or NEC ConfMan is an implementation of the FIWARE IoT Discovery Generic Enabler. This implementation is specifically designed to interwork with the IoT Broker GE FIWARE reference implementation, serving as the registry of FIWARE NGSI context providers. The detailed information about the IoT Discovery GE can be found in [FWa].

ConfMan is responsible for discovering the availability of context. The availability of context must be through the registrations made by IoT Agents (i.e. data providers). By registering, data providers make their access endpoints available to the data consumers. Data providers offer information about context entities (i.e. virtual entities) and their attributes (e.g. measurement values, metadata). The role of IoT Agents can be taken by either the Data Handling GE in IoT Gateways or the Back-end Device Management GE. Other IoT Backend systems may also provide context information.

Typically, context source availability information is forwarded to the FIWARE Context Broker GE. This allows context information (i.e. information generated by or coming from the "things") to be available and accessible to applications. ConfMan is integrated with two other components: IoT Broker and IoT Knowledge Server. The ConfMan source code is accessible at https://github.com/Aeronbroker/NEConfMan.

3.4.4 Crowd Estimation Service & Mission Control

The AUTOPILOT IoT architecture's IoT layer includes the "Analytics" building block. To fulfil the functionality required by this building block, NEC FIWARE-based IoT Platform includes an analytics component which is integrated with FIWARE IoT Broker and IoT Discovery. The name of this component is **Crowd Estimation and Mobility Analytics (CEMA).**

CEMA works on raw data which is received from IoT devices deployed in the field such as Wi-Fi sensors or stereoscopic cameras to count the people and estimate crowdedness in certain areas such as road sides. This service can be useful to understand pedestrian traffic and notify the autonomous vehicle beforehand about the expected crowdedness levels of nearby roads. This could be used for better route planning. Moreover, a Wi-Fi sniffing device with GPS sensors is placed in autonomous cars for mobile sensing.

The "Service management" building block of the IoT layer of the AUTOPILOT IoT architecture is provided by the **Mission Control Enabler (MICE)** Framework. The service management component allows configuration of a pre-deployed system for enhancing capabilities of IoT. For instance, CEMA service can be supported with the MICE framework. In this scenario, the wireless sensors can be placed in vehicle or in places such as where RSUs are located. The services of the sensors can be configured and managed by the service management in the IoT layer through. The service management can then provide the CEMA analytics output data to the AUTOPILOT applications

through the context management. The MICE framework enables more control and flexibility for dynamic missions such as operation of CEMA in a certain environment (e.g. highway, university campus).

Application developers can access CEMA analytics results through NGSI subscriptions or queries through the IoT broker.

3.4.5 FIWARE Status of Deployment and Access

The deployment status and access procedure of the FIWARE components are summarised in Table 4.

FIWARE Aeron Broker and NEConfMan components are currently deployed in the servers which are provided by NEC in Heidelberg, Germany. This server is dedicated for the Brainport pilot site and more generally to the central IoT platform. The server can connect to the oneM2M platform in Brainport pilot site.

Since the server is inside NEC, the components currently reside inside the internal NEC VPN. However, the access to the FIWARE IoT Broker and Discovery can be provided through whitelisting certain IP addresses. In the near-future, the physical location of the components may change.

Currently, CEMA service is available through the IoT platform service. CEMA and the IoT platform reside in the NEC server and CEMA outputs can be openly shared with all partners. CEMA is integrated with a lighter version of the IoT Broker which can be referred to as the Thin Broker. The CEMA service was tested in the first Brainport plug fest on 23-25 January 2018 and access is provided to the use case developers.

Table 4 – FIWARE – Deployment Status and Access

Platform	FIWARE		
Hosting	NEC Servers, NEC Laboratories Europe, Heidelberg, Germany		
Deployment Status	***AERON Broker***	Deployed	Dedicated to the Brainport pilot site
	NEConfMan	Deployed	Dedicated to the Brainport pilot site
	CEMA	Deployed	Dedicated to the Brainport pilot site
	oneM2M Interworking Proxy	Deployed	The FIWARE-oneM2M interworking proxy was successfully tested
Purpose/Pilot Site	To be used primarily for the Brainport pilot site		
Standard/Protocol	HTTP (Rest)		
URL	***AERON Broker***		
	NEConfMan		
	CEMA		
Connected Devices and Applications			
Access Process	Since the server is inside NEC, the components currently reside inside the internal NEC VPN. However, the access to the FIWARE IoT Broker and Discovery can be provided through whitelisting certain IP addresses. In the near-future the physical location of the components may change.		
Restrictions	***In addition to the above conditions, please note that personal information, images, or videos, must not be sent to the NEC FIWARE broker instances.***		

3.4.6 Future work

ETSI ISG CIM is currently standardising a new interface for FIWARE, which is called NGSI-LD. NEC is part of this standardisation group. The new interface will JSON-LD based and it will allow more expressiveness for easier application development.

Since the new JSON-LD based interface will enable features that are possible with more expressiveness with graph-based ontologies, NEC plans to implement a new version of the IoT Broker and IoT Discovery components. When the new version is ready, it will be deployed and enabled to AUTOPILOT with the new NGSI-LD interface.

3.5 HUAWEI OceanConnect Platform

This section provides an overview of the Huawei OceanConnect IoT platform based on the official Huawei documentation [HOC].

The Huawei OceanConnect platform is an open ecosystem built on IoT, cloud computing and Big Data technologies. It provides over 170 open APIs and serial agents that enable application integration, simplify and accelerate device access, guarantee network connection and realise seamless connection between upstream and downstream products for Huawei partners.

The Huawei OceanConnect IoT platform connects IoT devices in a secure way to the IoT cloud platform, enabling bidirectional communication to easily collect data and deliver commands between devices and the platform.

The key features provided by the Huawei OceanConnect IoT platform are:

- **Agile and Easy-to-Use Device Integration:** OceanConnect supports more than 20 mainstream cable and wireless IoT protocols. It is also pre-integrated with mainstream IoT chips and modules.
- **Complete and Highly Efficient Device Management**: OceanConnect's device management portal provides powerful functions and a user-friendly GUI for managing and configuring devices, visualising their status, identifying faults and performing firmware/software upgrade and maintenance.
- **Flexible and Open Applications**: The platform provides over 170 APIs and functions for device management, data and rules, allowing users to quickly create network.
- **Highly-Concurrent and Highly-Reliable Cloud Services:** Hundreds of millions of connections are supported with a service reliability of 99.9%. In addition, the Huawei OceanConnect IoT platform provides E2E security protection, device-level certification/authentication, low-power optimisation and complete application-level access control.

3.5.1 Huawei OceanConnect IoT Platform Architecture and Capabilities

Figure 8 depicts the Huawei OceanConnect IoT platform architecture. The supported functionalities are briefly described in this section. For further information, please refer to the platform official documentation [HOC].

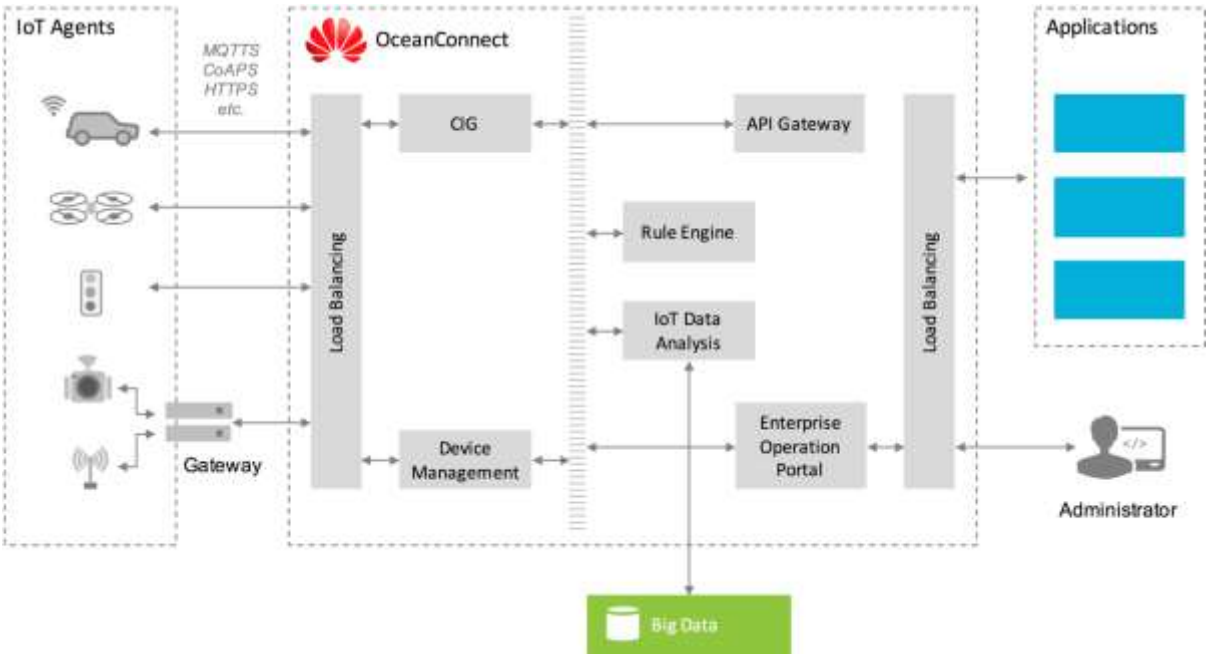

Figure 8 – HUAWEI OceanConnect IoT Platform Architecture

- **Cloud Inter-Networking Gateway (CIG):** The CIG is OSGi-based [OSGi] and comprises modules that transmit messages between IoT devices and the IoT platform and supports multiple communication protocols including TCP, UDP, MQTT, CoAP and LWM2M.
- **IoT Agent:** Serialised IoT Agents (such as Agent Rich, Agent Lite, Agent Tiny and Agent IPC) are deployed on different types of gateways and IoT devices. Multi-vendor devices can be quickly integrated with the cloud platform. The Agents are pre-integrated with near-field communication protocols such as Z-Wave, ZigBee, Wi-Fi and Bluetooth; they are used to manage data links.
- **Device Management:** Bidirectional data channels between devices and the IoT platform are used for device data reporting and remote control, including full-lifecycle management functions, such as getting devices online, maintenance, network connections, alarms, report analysis, upgrade and deregistration.
- **Rule Engine:** Simplified and flexible rules enable linkage and triggering of messages, notifications and alarms between devices.
- **IoT Data Analysis:** The Huawei OceanConnect IoT platform processes concurrent real-time data, stores mass data, computes data and exposes data APIs.
- **Service Operation and Management Portal:** The portal consists of the following modules: application management, device management, reports management, rule engine, software management, sub-account management and service status statistics.
- **API Gateway:** OceanConnect exposes over 170 types of functions, such as device management, rule engine and data analysis, helping developers to quickly create new applications.

3.5.2 Connecting Applications and Devices to OceanConnect

This section provides the high-level principles of connecting applications and devices to OceanConnect. For more details, please refer to section "Open Capabilities of OceanConnect" of the platform's official documentation [HOC].

Applications requiring access to OceanConnect need to be authenticated first. Once an application is successfully authenticated, it may perform the following actions:

- Collect device data from the IoT connection management platform using either an active query or data subscription.
- Issue commands to a specified sensor through the IoT connection management platform.
- Issue rules to the IoT connection management platform allowing response events and commands to be triggered based on the rules.
- Subscribe to device information (events) from the platform.

Devices may be connected to OceanConnect in two steps:

1. Connect the device to the IoT connection management platform either directly or indirectly through a gateway. Direct connection may rely on near-end integration (i.e. device integrates the IoT Agent Lite with the system and invokes the interface opened by the Agent Lite) or remote interrogation (i.e. device is interconnected to the CIG and a plug-in is compiled and sent to the CIG to enable the device to access the platform). Gateways may be connected through an Agent (Rich) or Agent Lite depending on the network capabilities and the communication protocol used between the device and the gateway.
2. Based on the selected access mode, determine the Agent to be used and then use the API provided by the Agent to complete integration development and implement data reporting and command receiving.

3.5.3 Interacting with OceanConnect: Open APIs

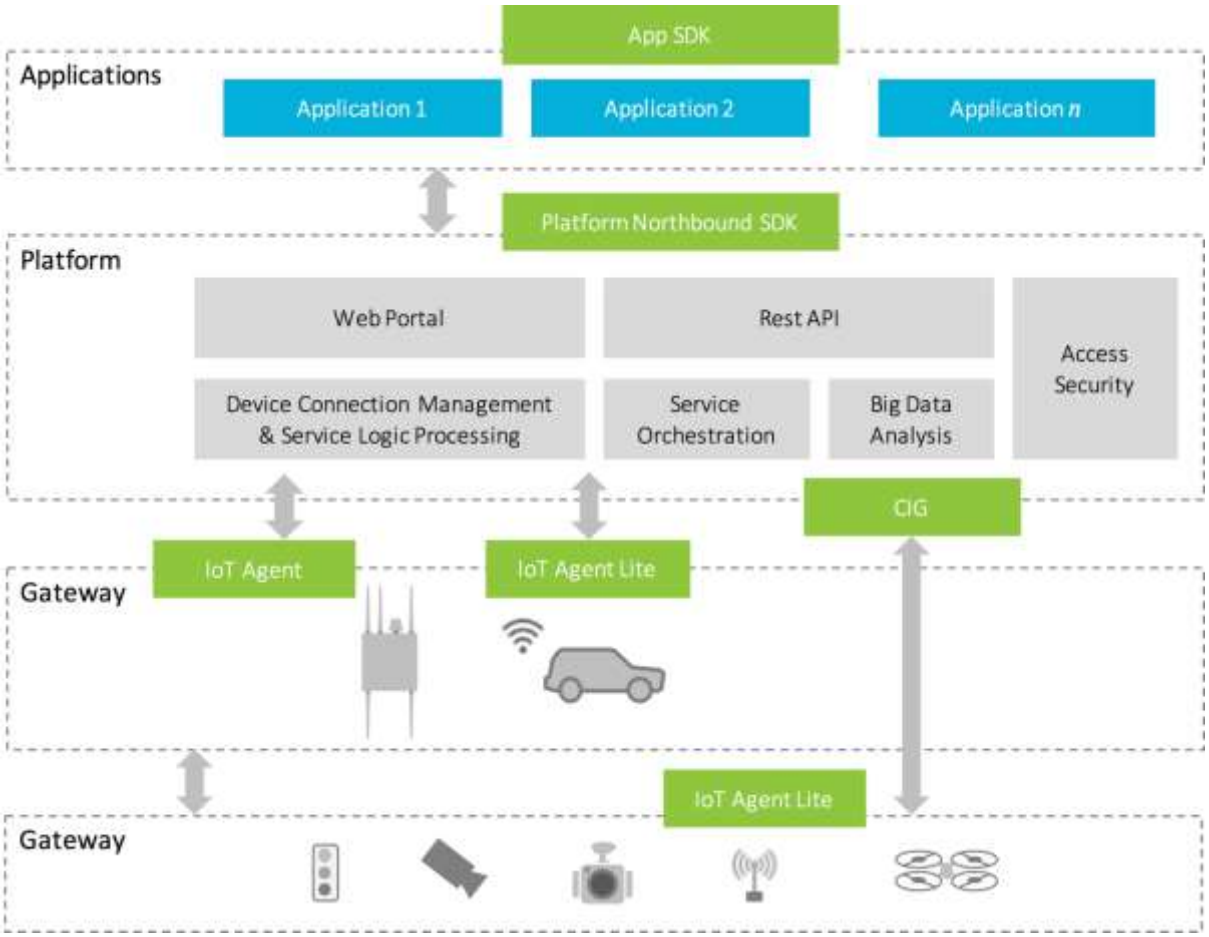

Figure 9 – Overview of Huawei OceanConnect IoT Platform open APIs

OceanConnect provides a set of APIs for developers to use for different purposes. The OceanConnect IoT Platform open APIs are depicted in Figure 9 and introduced below. For further details, please refer to section "Open APIs of OceanConnect" of platforms official documentation [HOC].

- **API Software Development Kit (SDK):** Provides developers with interfaces to integrate (or newly develop) open capabilities of the OceanConnect on a mobile APP.
- **Platform Northbound API:** Supports six capabilities opened to application developers, including authentication, device management, data collection, device service invocation (command issuing), rule issuing and message pushing.
- **Agent:** Provides a gateway integration API, a Z-wave sensor integration API and a ZigBee sensor integration API.
- **Agent Lite:** Currently, the Agent Lite is provided through the SDK and the supported development languages are Android and the C programming language. The Agent Lite is applicable to gateway integration and device integration.
- **CIG:** Provides an API for northbound sensor integration. On the CIG, Northbound transport protocol and integration have already been completed (currently, only interconnection with the Huawei Northbound module is supported). Developers only need to develop the data format conversion plug-in on the CIG. The CIG also provides APIs that can be invoked by developers. The CIG API document and development guide are not yet released on the Huawei Developer website.

3.5.4 OceanConnect Deployment and Access

HUAWEI provides an instance of OceanConnect for the Brainport pilot site. A summary of the deployment status and access to the platform is provided in Table 5.

Table 5 – HUAWEI OceanConnect Platform – Deployment Status and Access

Platform	HUAWEI OceanConnect IoT Platform	
Hosting	OceanConnect Platform	
Deployment Status	IoT Platform	One development instance is deployed and available to use, for learning, development, experimentation and testing purposes
	oneM2M Interworking Proxy	The oneM2M interworking proxy is not developed yet.
Purpose/Pilot Site	To be used primarily for the Brainport car rebalancing/relocation	
Standard/Protocol	HTTP/MQTT	
URL	http://developer.huawei.com	
Connected Devices and Applications	Solutions cover Smart Home, Internet of Vehicles, Public Utilities	
Access Process	The relevant to the Brainport car rebalancing/relocation development instances are available to the Brainport pilot site. Access to these instances may be requested from Huawei Technologies, through the following points of contact: • Liuxin Walle walle.liuxin@huawei.com Further instructions and conditions regarding access will be provided in due time.	
Restrictions	***In addition to the above conditions, please note that personal information, images, or videos, must not be sent to the Watson IBM for Automotive service instances.***	

3.6 TIM IoT Platform

3.6.1 Overview of the TIM IoT Platform

The TIM IoT platform is based on the oneM2M standard [ONE17]. TIM provides the oneM2M platform as PaaS (Platform as a Service). Figure 10 shows a high-level architecture of the platform.

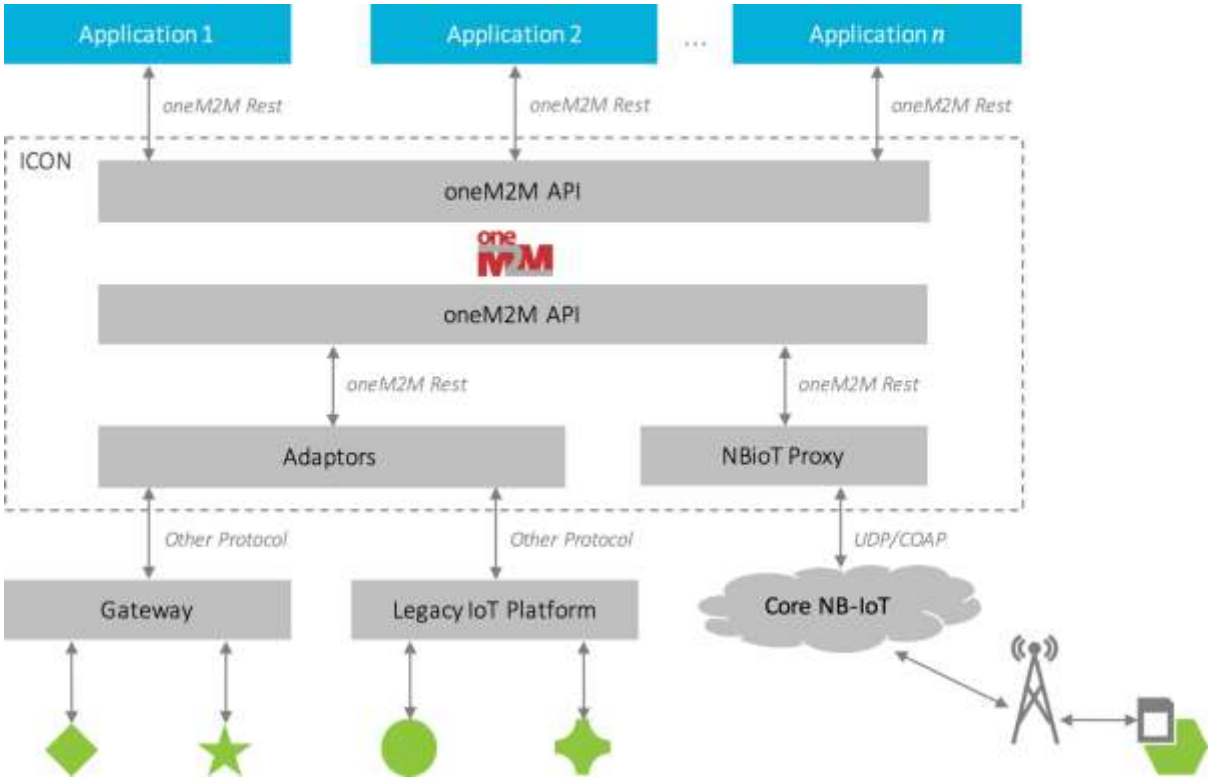

Figure 10 – High-Level Architecture of the TIM oneM2M Platform

The platform is based on the Ocean [OIP17] open source project. Its main features are:

- Compliance with the oneM2M standard
- Southbound and northbound Rest APIs for data storage and sharing
- Data sharing by means of pull and push (subscription/notification)
- URIs for identifying resources
- Web console for resource management and provisioning
- Web console for administrators

Security in the TIM oneM2M platform is based on the following features.

- oneM2M services and APIs are exposed through SSL (HTTPS).
- Authorisation is based on credentials (username/password) associated with a specific user (tenant).
- An Access Control Policy (ACP) needs to be created for each Application Entity. ACP is defined as a set of conditions that determine whether entities are permitted to access a protected resource.

3.6.2 TIM oneM2M Platform Deployment and Access

TIM provides an instance of its oneM2M IoT platform on TIM Self Data Center, a commercial platform for hosting, managed by TIM. The platform is exposed on public Internet at https://icon-lab.tim.it. The TIM oneM2M status of deployment and access is summarised in Table 6.

Table 6 – TIM oneM2M Platform – Deployment Status and Access

Platform	TIM oneM2M Platform
Hosting	TIM Self Data Center
Deployment Status	One instance is deployed and available to use
Purpose/Pilot Site	Collect data from all the Livorno pilot site use cases: • Highway use cases:

	○ Hazard (puddles) on the roadway ○ Roadway works with traffic control centre in the loop • Urban Driving use cases: ○ Pedestrian detection with camera ○ Connected bicycle ○ Potholes detection
Standard/Protocol	oneM2M
URL	https://icon-lab.tim.it
Connected Devices and Applications	Environment sensors (temperature, humidity, pollution, rain, sound level pressure), parking status, traffic flows, smart waste, smart green, WiFi scanner, metering (power consumption)
Access Process	The instances are available to all the pilot sites and use cases. Access to these instances may be requested from TIM; any devices and device types to connect to the oneM2M platform must be registered and provisioned in advance by TIM. Once the devices are registered, you will receive credentials for each device/application.
Restrictions	***Please note that personal information, images, videos, documents must not be sent to the oneM2M platform***

3.6.3 Interfacing with the TIM oneM2M Platform

The oneM2M platform is accessible through HTTP(S) APIs, using GET and POST to read/write data. Table 7 provides the main HTTP methods needed to interact with the platform.

Table 7 – Interfacing with the TIM oneM2M Platform

Posting Data to the TIM oneM2M Platform
The body of the message contains the data that to post to the TIM oneM2M platform. **Method:** HTTP POST **URL Pattern:** https://icon-lab.tim.it/onem2m/<APPLICATION_ENTITY>/<CONTAINER>
Getting Dara from the TIM oneM2M Platform
The body of the response message contains the data retrieved from the TIM oneM2M platform. **Method:** HTTP GET: **URL Pattern:** https://icon-lab.tim.it/onem2m/<APPLICATION_ENTITY>/<CONTAINER>
Subscribing to Notifications from the TIM oneM2M Platform
It is possible to subscribe to notifications of data being received by a container. The mechanism involves creating a Subscription resource within the container in question, indicating the target endpoint where you will notification should be posted (HTTP POST). **Method:** HTTP GET: **URL Template:** https://icon-lab.tim.it/onem2m/<APPLICATION_ENTITY>/<CONTAINER>

HTTP is the most used protocol, but the platform also supports MQTT [TS10] and CoAP [TS08] oneM2M bindings.

3.7 SENSINOV oneM2M Platform

The SENSINOV oneM2M platform is used as the interoperability bridge for linking the AUTOPILOT IoT platforms (cf. Figure 3). Its main features are listed below:

- **Supported Nodes:** IN-CSE, MN-CSE, ASN-CSE and ADN
- **Reference Points:** MCA and MCC interfaces
- **Resource Types:** CSEBase, RemoteCSE, ACP, ASAR, AE, Container, ContentInstance, Subscription, Group, Node, Request, PoA, Discovery, Notification, etc.
- **Request Primitives:** Retrieve, Create, Update, Delete, Discovery and Notify

- **Interworking Proxies:** Watson IoT Platform, FIWARE Context Broker and Vedecom OEM Platform
- **Addressing Formats:** Structured and unstructured
- **Addressing Modes:** Absolute, SP-relative and CSE-Relative
- **Protocol Bindings:** HTTP, CoAP, MQTT and Websocket.
- **Content Formats:** XML and JSON
- **Communication Modes:** blocking, non-blocking synchronous and non-blocking asynchronous
- **Multi-Hop:** Retargeting via PoA.
- **Storage:** SQL and NoSQL (SQL H2 by default)
- **Security:** SSL/TLS
- **User Interface:** Web interface for browsing oneM2M Resources and command line OSGi console

3.7.1 SENSINOV oneM2M Platform Deployment and Access

The SENSINOV oneM2M status of deployment and access is summarised in Table 8.

Table 8 – SENSINOV oneM2M Platform – Deployment Status and Access

Platform	SENSINOV oneM2M Platform
Hosting	TNO Self Data Center SENSINOV Digital Ocean cloud
Deployment Status	Two instances are deployed and available to use for Brainport and Versailles pilot sites. One instance is provided to Tampere pilot site (Work in progress).
Purpose/Pilot Site	Interoperability between IoT devices and IoT backend applications Main IoT platform in the Tempere pilot site
Standard/Protocol	oneM2M standard
URL	Brainport: https://vmi137365.contaboserver.net:8443/web Versailles: http://46.101.225.249:8080/web
Connected Devices and Applications	Continental cloud applications (work in progress) VEDECOM OEM (Work in progress)
Access Process	Certificates or application keys to be obtained from SENSINOV for Versailles: • Mahdi Ben Alaya benalaya@sensinov.com Certificates or application keys to be obtained from TNO for Brainport: • Daan Ravesteijn daan.ravesteijn@tno.nl
Restrictions	

3.8 Technolution MobiMaestro Platform

MobiMaestro is a Technolution proprietary IoT platform, used by traffic management centres (TMC) in the Netherlands and Denmark.

3.8.1 Overview of the MobiMaestro Architecture

Figure 11 shows a high-level architecture of the MobiMaestro architecture.

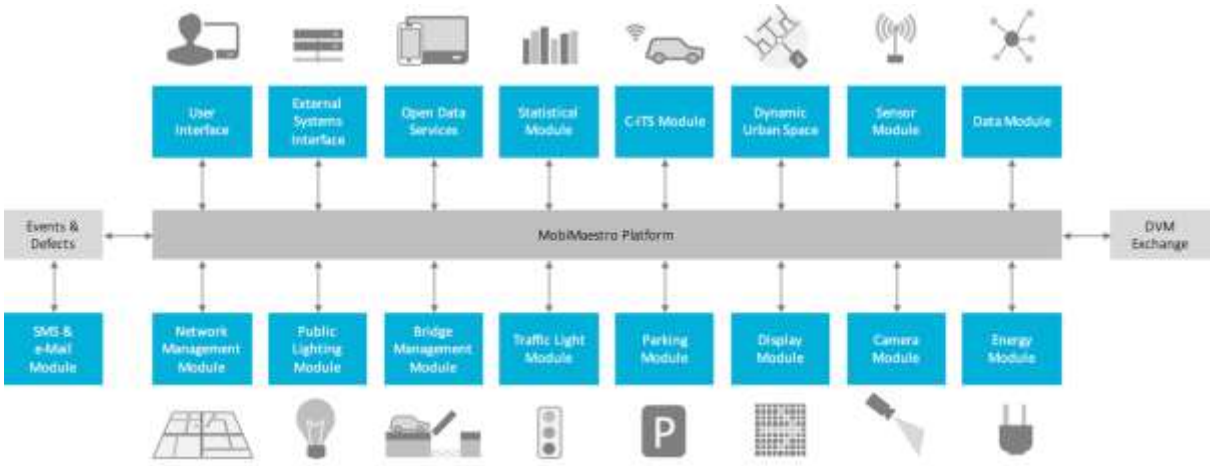

Figure 11 – High-Level Architecture of the Technolution MobiMaestro Platform

As part of AUTOPILOT, an effort is being carried out to upgrade the MobiMaestro platform towards a true IoT-compliant platform using the target architecture illustrated in Figure 12. With this IoT architecture, the Service Bus structure of our current MobiMaestro platform is reused as proprietary IoT platform and is connected to the oneM2M interoperability platform using oneM2M interworking gateway.

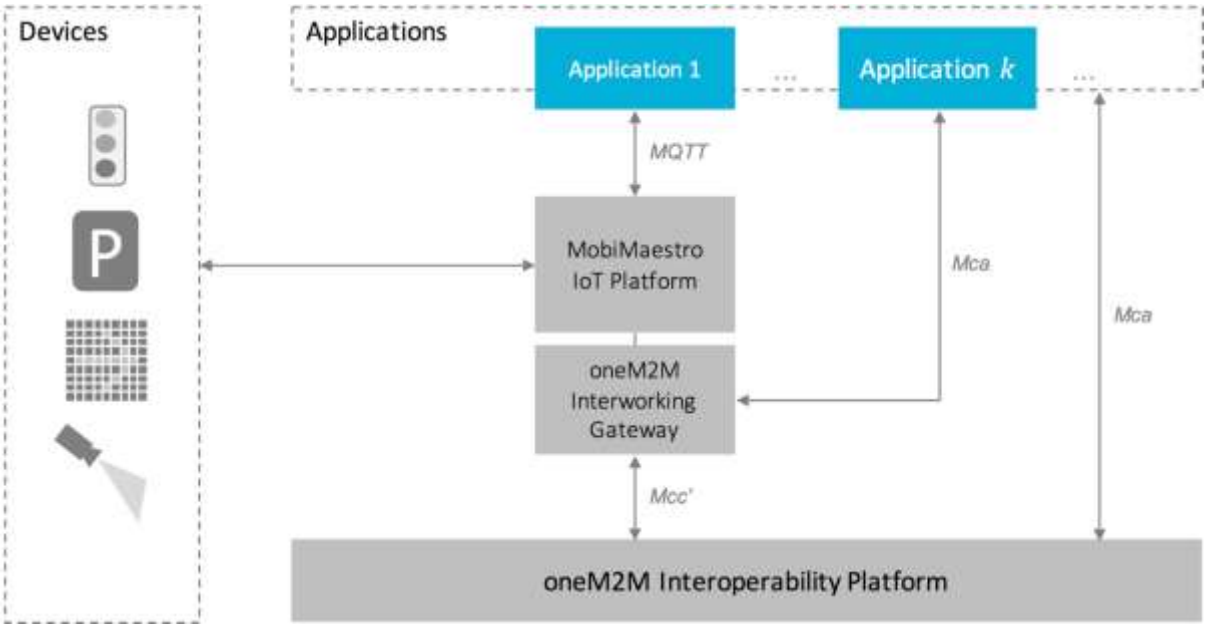

Figure 12 –Technolution MobiMaestro Target Architecture Platform

The MobiMaestro IoT platform will contribute TMC information to the Brainport pilot. It will provide, through the oneM2M interoperability platform, lane-information (for supporting the shoulder access for the platooning use case), traffic light information (for supporting the urban driving and platooning use cases) and parking information (for supporting the valet parking use case).

3.8.2 MobiMaestro Platform Deployment and Access

Technolution provides an instance of its IoT MobiMaestro on the (KPN) data centre, a commercial platform for hosting, managed by Technolution. The platform is exposed on public Internet. The MobiMaestro status of deployment and access are summarised in Table 9.

Table 9 – MobiMaestro – Deployment Status and Access

Platform	Technolution MobilMaestro Platform
Hosting	KPN Data Centre managed by Technolution
Deployment Status	One instance will be deployed for use (Q3 2018)
Purpose/Pilot Site	Exchange TMC data with the Brainport pilot site use cases: ○ Shoulder access ○ Traffic Light information: time to green (TTG) and time to red (TTR) ○ Parking information Exchange general data on road status ○ Hazardous events Actual speeds
Standard/Protocol	Datex II / ETSI / SPDP (Dutch open standard for parking)
URL	Not yet available
Connected Devices and Applications	TLC, Parking, Datasources for roadstatus, VMS
Access Process	Using user ID and password on data connections.
Restrictions	

3.9 Future Work

Currently, use case developers in all the pilot sites already started using an IoT platform of their choice out of the listed platforms. Some partners started exchanging data through the oneM2M IoT platform. One missing step is to make all deployed IoT platforms available and accessible for the partners who want to leverage them. Due to certain security measures, this process requires some time. Moreover, the IoT platform providers should provide necessary technical support for the easy usage of the IoT platforms and maintain the server components.

One important step for future work is to provide information exchanges between different IoT platforms. For this step, all the interworking gateways should be deployed and tested with data. Currently, only the Watson IoT Platform and FIWARE interworking gateways have been tested. Note that the TIM oneM2M platform is already oneM2M-compliant. The interworking gateways are expected to facilitate interoperability between the platforms, which will help make the use cases uniform across all the pilot sites

4 Pilot Site IoT Ecosystems

This chapter provides a status update of the implementation of the pilot site IoT ecosystems for the pertaining use cases. This includes the deployed and planned IoT platforms and devices.

4.1 Finland (Tampere)

4.1.1 Overall IoT Architecture of the Finnish Pilot Site

The Finnish pilot site ecosystem involves AD vehicles with a oneM2M IoT platform, a mobile road side unit on which a traffic camera is installed, and connections to a traffic light server and an application for parking reservations. A parking management application will be developed, for managing the parking route and for monitoring the unmanned parking manoeuvre.

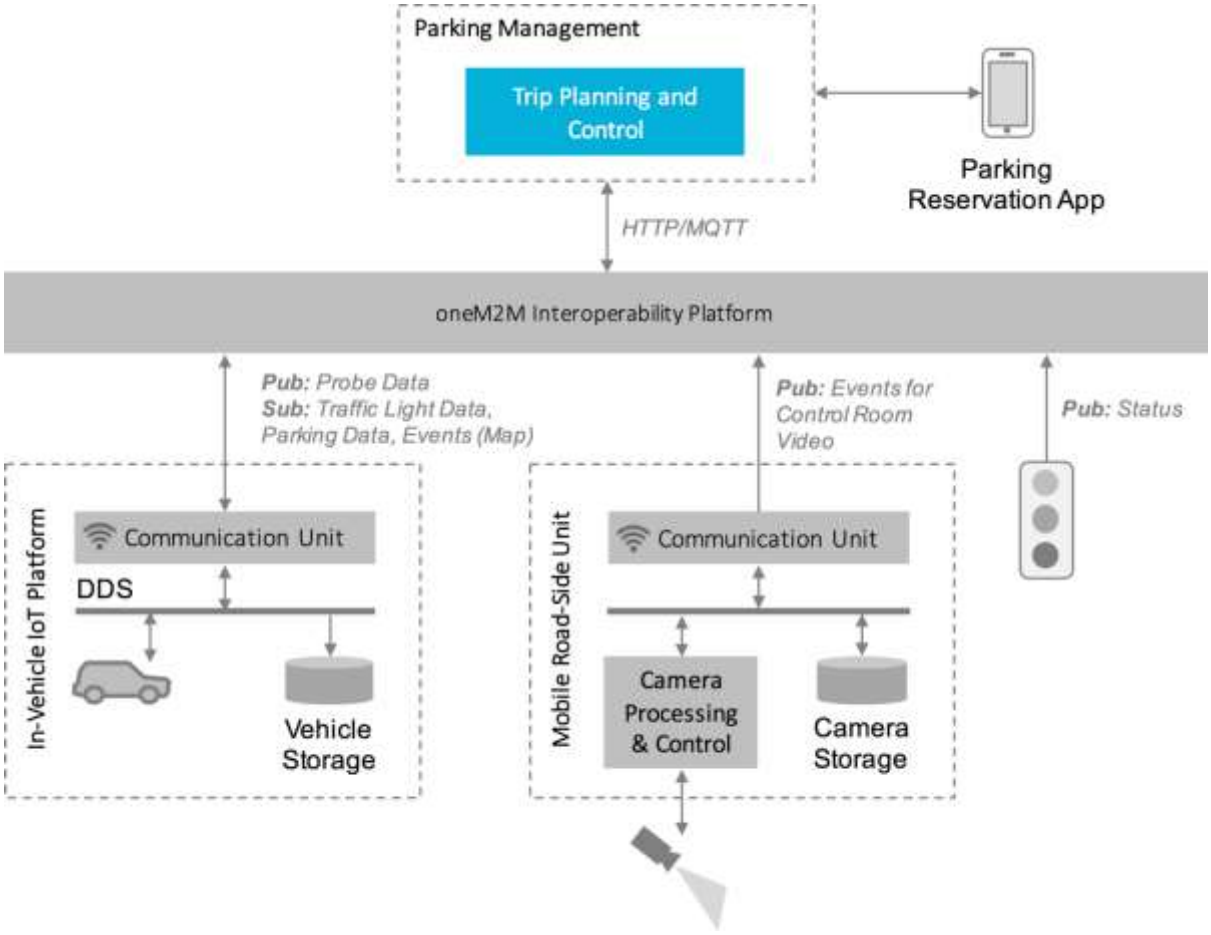

Figure 13 – Finnish Pilot Site Overall Architecture

Two use cases are being implemented in the Finnish pilot site: urban driving and valet parking. Figure 13 shows the architecture of the Finnish pilot site and the required components for both use cases under development. The pilot site devices include: IoT connected cameras and connected traffic lights. Parking spots are booked through a parking reservation application.

The traffic camera, vehicles and the parking management application will be connected to the open IoT platform, providing continuous updates of their status and making these available to the different applications that might make use of them.

4.1.2 IoT Ecosystem Implementation in the Finnish Pilot Site

Details of the implemented IoT ecosystem components for each of the Finnish pilot site use cases are provided in the following subsections.

4.1.2.1 Automated Valet Parking Use Case Implementation

The goal of this use case is to provide to the user a service that, through IoT technologies and autonomous car driving functionalities, can automate and improve the way of booking parking spots and enable the car to automatically park itself once left in a designated drop-off area. The AVP use case in Tampere will be achieved through the following components and functionality.

Object Detection with Cameras

As shown in Figure 13, a camera, installed at the mobile road side unit, is used to detect pedestrians or obstacles on the parking site. The information is processed locally and events are extracted and published to the IoT platform. Both the in-vehicle platform and the parking management system receive this information. The parking management system calculates the available route for the vehicle. The in-vehicle IoT platform uses this information to update the world model and then either modifies the trajectory or stops to avoid any collision. The traffic camera also transmits video to the parking management system for visual monitoring of the parking route.

Parking Management System

The Parking management system both reserves parking spot and optimises routes to the parking spots. It performs visual monitoring of the parking manoeuvre. When the vehicle is driving in unmanned mode, the operator in the parking management system acts as a driver and can force the vehicle to start moving or to come to a stop.

Connected Car with in-Vehicle IoT Platform

A connected car will also act as an IoT device publishing information to the IoT platform. The vehicle will receive information from the parking management system related to the parking manoeuvre and from the camera on objects on the path. The in-vehicle platform is further defined in the AUTOPILOT deliverable D2.1 [D2.1].

4.1.2.2 Urban Driving Use Case Implementation

The goal of this use case is to show how IoT can impact the safety of VRUs and the performance of autonomous driving at signalised intersections.

Pedestrian Detection with Camera (VRU)

The traffic camera, which is installed at the mobile road-side unit, is used to detect VRUs (pedestrians or cyclists) who have a green traffic light at the same time as the turning vehicle. The detection information is processed locally and published to the IoT platform and to the RSU.

The in-vehicle IoT platform in the vehicle receives the information and the vehicle stops before the pedestrian crossing until the crossing is free.

Traffic Lights

Traffic light status information and time to the next signal phase are received either through V2X or from the server of the traffic light operator (Dynniq). The information is transmitted to the vehicle.

Connected Car with in-Vehicle IoT Platform

The vehicle, as in the AVP use case, acts as an IoT device publishing information to the IoT platform. The vehicle will receive information from the traffic camera and the traffic signals.

4.2 France (Versailles)

4.2.1 Overall IoT Architecture of the French Pilot Site

Three use cases are being implemented in the French pilot site: car sharing, urban driving and platooning. The IoT architecture of the pilot site for these three use cases is provided in Figure 14.

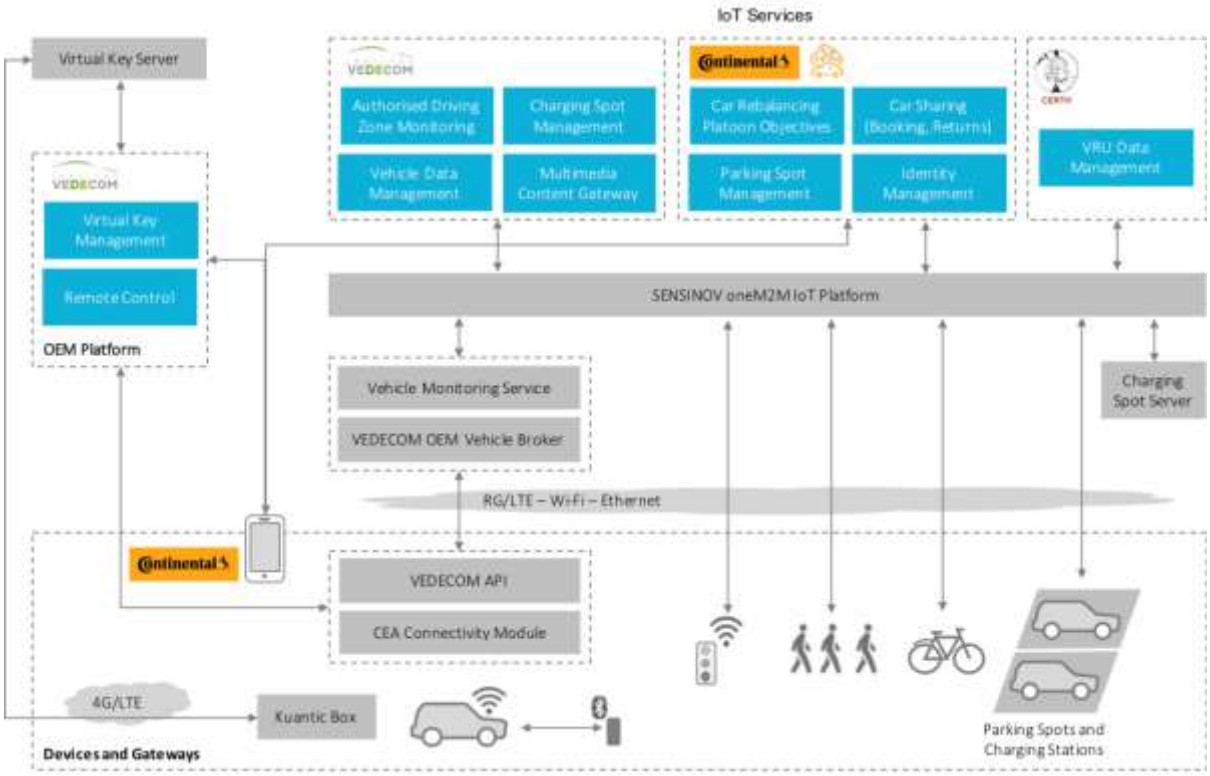

Figure 14 – French Pilot Site Overall Architecture

4.2.2 IoT Ecosystem Implementation in the French Pilot Site

The implementation of the IoT ecosystem in the French Pilot Site is summarised in the following sections.

All the data mentioned in the following sections are stored and shared between services on the SENSINOV oneM2M platform. This platform is provided by Sensinov and accessible to all the partners involved in the pilot site.

4.2.2.1 Car sharing

The car sharing use case requires multiple IoT devices on the infrastructure. Each parking slot used in the experiment will be equipped with sensors used to detect whether the slot is available or occupied. Sensors will be able to determine which car of the fleet is parked in a specific spot, allowing the system to manage the whole fleet. The charging spots will also provide information about their states, indicating whether a car is currently charging or not.

4.2.2.2 Urban driving

Cars used for the experimentation are Renault Twizy and have been equipped with multiple sensors and communication systems. These devices are used by the car to detect objects in its environment and communicate with other users, other cars or road equipment.

Some information about interesting places in the city of Versailles can be sent to the driver. To do so, some Bluetooth low energy (BLE) beacons will be installed on selected points of interest. These beacons will communicate with the embedded screen of the car to provide audio or video content to the driver.

Vulnerable road users, such as pedestrians and cyclists, carry sensors to determine their positions and state. This information is sent to the car to increase the accuracy of the perception of its environment.

4.2.2.3 Platooning

To rebalance the number of cars available in the different parking sites of the experiment, platooning will be used to allow a single operator to move multiple cars. This use case requires different sensors and communication systems in the cars. It also requires communication with road side units to handle road crossings with the platoon. The traffic lights in the way of the platoon need to be equipped with such units.

4.3 Italy (Florence-Livorno)

The Italian pilot site is a testing infrastructure encompassing the Florence-Livorno freeway together with road access to the Livorno sea port settlement. The testbed consists of three zones: The Livorno-Florence freeway, The Traffic control centre (TCC) located in Empoli and the port landside just in front of the cruise terminal. The vehicles that will be used in the test site are FCA Jeep Renegade with different functions and roles:

- Two vehicles by CRF with automated driving functions, to be used to demonstrate the performance of the IoT-ITS ecosystems when the automated driving scenarios beyond SAE 3 levels are running;
- Five service vans by CRF and AVR with advanced V2X communication capabilities, to be used for tuning and pre-testing the vehicular systems and services in the IoT enhanced ITS environment.

Two use cases are being implemented in Livorno:

1. **Highway Pilot:** A cloud service merges the sensor measurements from different IoT devices, such as vehicles and roadside cameras, to locate and characterise road hazards.
2. **Urban Driving**: Focuses on the interaction with traffic lights and legacy traffic, on the robustness of the AD functions of the vehicle, safety when dealing with vulnerable road users and positioning.

4.3.1 Overall IoT Architecture of the Italian Pilot Site

Figure 15 shows the IoT architecture of the Livorno pilot site for both highway driving and urban driving use cases. Details of the architectural elements for both use cases are provided in the subsequent subsections.

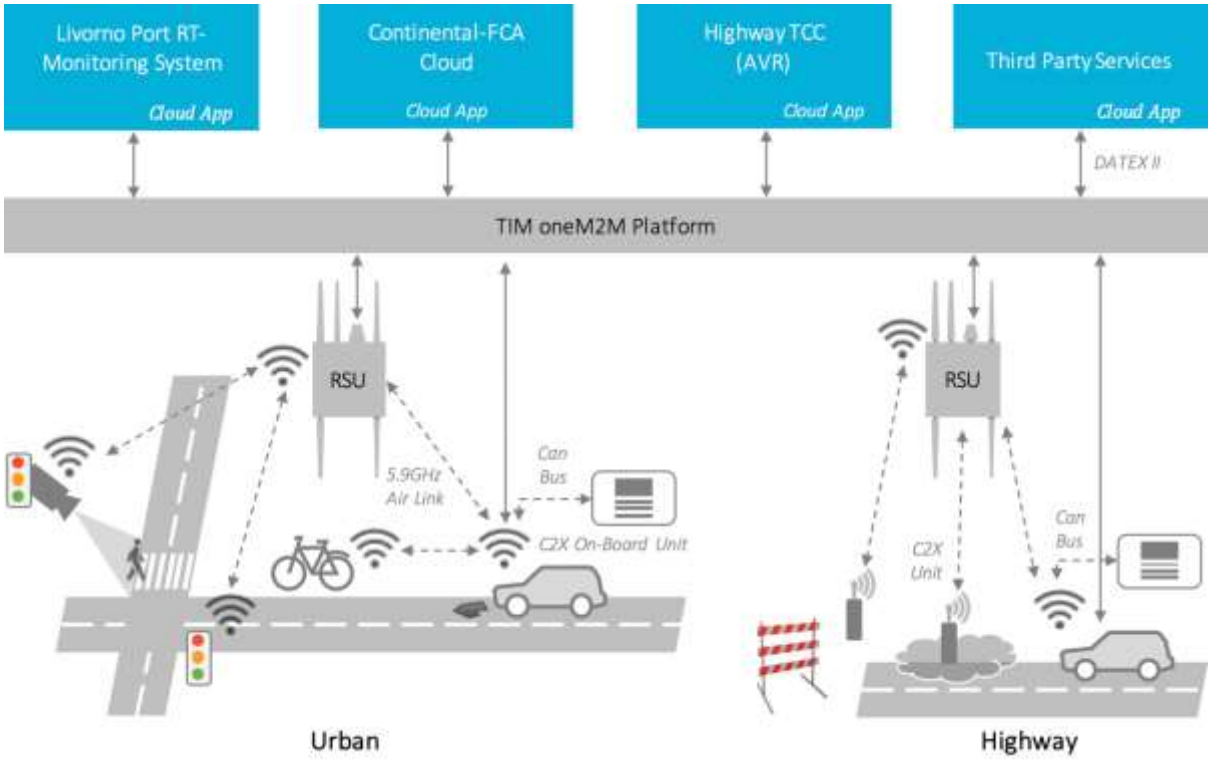

Figure 15 – Livorno Pilot Site Overall Architecture

4.3.2 IoT Ecosystem Implementation in the Italian Pilot Site

Details of the implemented IoT ecosystem components for each of the Italian pilot site use cases are provided in the following subsections.

4.3.2.1 Highway Pilot Use Case Implementation (Livorno-Florence)

The scope of these tests involves cars with IoT-enhanced AD functions, driving in a "smart" highway. The cars are Jeep Renegades with on-board equipment, the so-called IoT open vehicular platform, enabling IoT-triggered AD functions: speed adaptation, lane change, lane keeping. Some cars have special sensors also, such as the IoT-based pothole detector.

The "smart" highway is a freeway where a pervasive IoT ICT system is deployed based on a network of roadside sensors or other sources, capable of collecting information and making it available to cloud-based applications. Connected cars and the traffic control centre have an important role. For safety reasons, connected cars drive in a convoy, following the AD car.

The goal is to show how the combined use of IoT and C-ITS can mitigate the risk of accident for an AD car when hazards occur on the road. Here, we deal with two types of hazards: (1) puddles and (2) road works.

Puddles

- *Puddle IoT sensors:* IoT sensors placed along the highway continuously monitor the presence of puddles. When a hazard is detected, the sensors send an alert to the road-side unit (RSU) with detailed information, using IoT standard protocols. The RSU broadcasts the Decentralized Environmental Notification Message (DENM) [DENM14] to vehicles and to the traffic control centre (TCC). The TCC validates the alert and forwards the DENM message to farther away RSUs. At the same time, the TCC feeds the TIM OneM2M compliant platform with alert-related data. Two kinds of sensors are deployed, using different communication

technologies: the 6loWPAN puddle sensors send messages to the Road Side ITS-Station using CoAP[3]; the NB-IoT puddle sensor sends the message straight to the oneM2M platform using the Long-Time Evolution (LTE) cellular network and REST protocols.

- *Road-Side ITS Station:* Road-Side ITS Station is a programmable gateway with multi access technologies (notably 6LowPAN, ETSI ITS G5, LTE, Eth, etc.). It is a road side unit, compliant with ISO/TC204 WG16 standards [ISOTC204], able to exchange information over different networks, using different protocols, including the IoT ones. The RSU always listens the 6loWPAN sensors and sends the measurement to the OneM2M IoT platform of the PS with a certain frequency. When the hazard occurs, the RSU broadcasts a DENM with the lowest quality level of information (i.e. not yet validated by the TCC) to the approaching vehicles via the IEEE 802.11OCB network and the oneM2M platform via LTE cellular network. Furthermore, the RSU publishes on the oneM2M platform the cooperative awareness messages (CAM) collected from the vehicles in the Dedicated Short-Range Communications (DSRC) communication range [CAM14].
- *Traffic Control Centre:* The TCC implements a DATEX II[4] node that can supply information from the whole highway network. The TCC is also responsible for managing ITS on the oneM2M platform of the Italian PS. Two kinds of services are provided leveraging the subscription to the oneM2M platform: hazard validation and DENM forwarding. It also publishes to the oneM2M platform the relevant traffic information from the DATEX II node, to be consumed by the highway infotainment service (Fi-PI-LI App). When a hazard (e.g. flooding on the road) occurs, the TCC is notified through its subscription to the oneM2M platform. After assessing the severity of the danger, it validates the hazard and broadcasts a DENM with the highest quality level of information (i.e. validated by the TCC) to the RSUs along the Highway, using the cabled LAN. The TCC subscribes the CAMs of the vehicles published by the RSUs on the oneM2M platform. The information is combined with the Bluetooth and Wi-Fi transit data loggers to perform the travel time analysis and live overview on the TCC video wall. The TCC subscribes the AD car's sensor data on the oneM2M platform to provide ITS services to the users of the highway.
- *Autonomous Driving Car:* The AD car broadcasts CAMs over the IEEE 802.11OCB network. At the same time, the AD car publishes data from its sensors to the oneM2M platform. The AD car is approaching the hazard on the road. The in-vehicle application (Connected eHorizon) subscribes the alert from the oneM2M platform. The in-vehicle IoT platform combines the information obtained by the Connected eHorizon (CeH) with that obtained by DENM via the IEEE 802.11.OCB network. It, then, feeds the appropriate autonomous functions, which performs the necessary adaptation of the driving style in a "smooth" way if the message is received well in advance of reaching the hazard. However, if the vehicle is close to the hazard and for some reason (e.g. the warning from the IoT service wasn't received, or the warning was received just by the safety channels of DSRC), then an emergency braking is needed, this event is registered by the in-vehicle application and sent to the OneM2M IoT platform of the pilot site. At the same time, the FCA cloud monitors the performance of the

[3] Constrained Application Protocol (CoAP): http://coap.technology

[4] DATEX II: http://www.datex2.eu

vehicle, checks that the in-vehicle application feeds the appropriate autonomous functions, send notification/warning to the in-vehicle HMI.

- *Connected Cars: C*onnected cars lead and follow the AD car; they continuously broadcast CAMs over the IEEE 802.11OCB network. At the same time, they publish its sensor's data to the oneM2M platform. As the connected cars are approaching the hazard on the road, the in-vehicle IoT platform receives the information from both the RSU along the track and the OneM2M IoT Platform. The in-vehicle application pre-alerts the driver about the hazard using the information obtained by the OneM2M IoT Platform of the PS and by the DENM. The AD car on-board unit (OBU) can instantiate either smooth IoT-enabled speed adaptation and lane change, or eventually suggest an alternative route (if any) to the driver through HMI.

Roadworks

- A (WSN) sensor node is attached to the road works trailer and announces the presence of roadway works to an RSU. The RSU, in turn, triggers DENM messages, broadcasting information about available lanes, speed limits, geometry, alternative routes, etc. This RSU is located close to the roadway works and it can be a temporary ITS station as well, because if the road works are far away from the permanent RSUs put on gantry. They cannot be reached by the signal transmitted by the trailer. The temporary RSU will use the LTE network to communicate with the TCC, as Ethernet wiring is not available on site.
- The TCC broadcasts the DENM messages to farther away RSUs. At the same time, the TCC feeds the ETSI OneM2M platform with road works related data. This information is consumed by the Connected eHorizon application from CONTINENTAL and transmitted to the FCA cloud as a modified dynamic speed limit based on the generated dynamic event. The FCA cloud immediately notifies the enabled vehicles about the updated information for CeH devices installed on prototypes. The in-vehicle application, then, feeds the appropriate autonomous functions, which perform the necessary adaptation of the driving style, taking into consideration information obtained from DENM messages. A notification/warning through the in-vehicle HMI can then be generated.

4.3.2.2 Urban Driving Use Case Implementation (Livorno-Florence)

This use case demonstrates how IoT may impact the safety of VRUs in an urban-like scenario (instantiated at a harbour settlement) with AD cars, pedestrians at a traffic light crossing, connected bicycles and a sea port monitoring centre.

Pedestrian Detection with Cameras

Figure 15 shows a smart traffic light detecting a pedestrian or an obstacle on the lane. The information is processed locally and submitted to the RSU using IoT protocols and to vehicles via standard C-ITS messages. Moreover, a connected traffic light sends information about the time-to-green/red (SPAT/MAP messages [AG15]). The RSU receives the information, fuses the data and sends it by DENM to all the interested actors on the roads.

The OBU of the AD car receives the information and smoothly adapts the speed to the situation, e.g. if a pedestrian is crossing the road when the traffic light is green for the cars, the AD car will behave as if the traffic light is red. Moreover, the detection of VRUs and the traffic light status is displayed on the HMI in the car. The information from RSUs and OBUs is also sent to the IoT data platform via IoT standard protocols and it can then be processed by the Port Monitoring Centre for real time risk assessment and safety services.

Connected Bicycle

An OBU installed on an ITS G5 bicycle sends standard CAM notifying its presence on the road. The in-vehicle IoT platform of the AD car receives the information and smoothly adapts the speed to the situation. The driver is informed about the proximity of vulnerable users with a notification on the HMI while the AD car algorithms decide what actions to take. Information from the OBUs is also sent to the oneM2M IoT platform over CoAP and/or HTTP.

Potholes Detection

The goal of this use case is to demonstrate how additional IoT sensors placed in the AUTOPILOT prototype can enhance the functions of the car itself. In such a way, the vehicle can be used as an IoT sensor for detecting the surface conditions for both highway and urban scenarios.

A wireless vibration sensor is deployed on the connected vehicle, which transmits to the OBU via 6LowPAN protocols the occurrences of vibrational shock above a certain level. When the car encounters a pothole (see Figure 5), the vibration exceeds the threshold and the IoT vibration sensor sends an alert to the OBU. The latter combines this information with data from the CAN bus (speed, odometer, etc.) and GPS and sends this data to the IoT platform, where they can be fed into the AD car applications.

4.3.3 IoT Platform and IoT Devices Integration

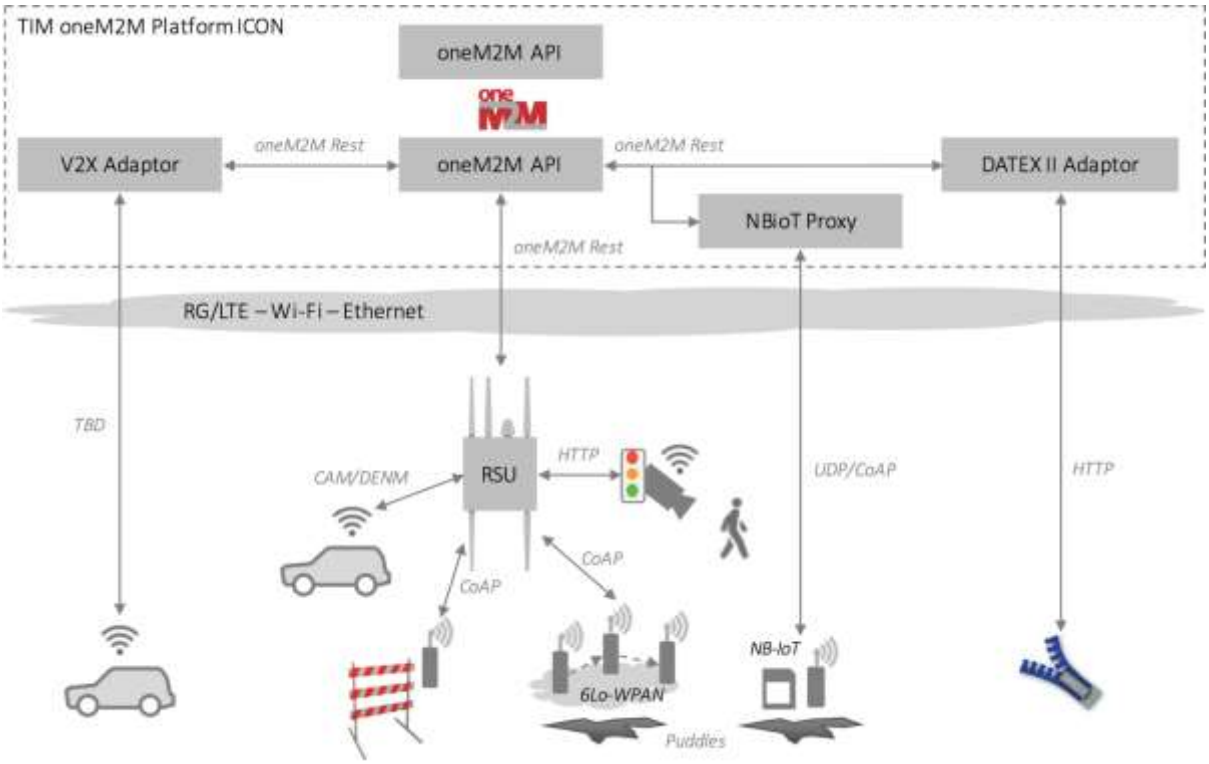

Figure 16 – Integration of IoT Devices into the oneM2M IoT Platform in the Italian Pilot Site

In the Italian pilot site, IoT devices are connected to the IoT oneM2M platform through the oneM2M standard, as shown in Figure 16.

Devices (e.g. OEM in vehicle components, inertial sensors, smartphone) in the Italian PS communicate with the oneM2M platform through gateways (On-Board Units) that integrate the IoT information from various sensors.

The in-vehicle IoT platform provided by ISMB offers communication interfaces to components that from the following partners: TIM (OneM2M Cloud platform), CNIT (accelerometer sensor for the pothole algorithm), CRF and AVR (intra-vehicle sensors).

The OBU can also exchange data with additional IoT devices such as inertial sensors and smartphone motion sensors: these data are interfaced with the in-vehicle IoT platform using CoAP/6LoWPAN or MQTT.

4.4 Korea

No IoT platform has been implemented in the South Korean pilot site.

4.5 Netherlands (Brainport)

The Dutch pilot site (Brainport) has two locations:

- Helmond site, covering both highway (A270) and parking area (of the Automotive Campus)
- Technical University Eindhoven (TU/e) campus

In addition, specific locations are targeted in the neighbourhood where the road surface has some detectable deficiencies.

Six use cases are being implemented in Brainport:

1. **Platooning:** The goal is to show added value of IoT data to platoon formation and subsequent platooning. This goes beyond the direct information exchange between the platooning vehicles, but also involves a so-called platooning service, which runs in the cloud and makes use of available IoT originated data to guide the platooning vehicles on the tactical level, e.g. on speed and lane advice. The platooning use case makes use of the oneM2M IoT platform.
2. **Highway Pilot:** Information about the state of the road surface (potholes, cracks, patches of ice, etc.) are collected and sent (published) to a central server and is also used to update electronic map data. This data is then made available to other vehicles and applications, for example to adjust their speeds. The use case involves at least two vehicles: a registration vehicle and one user vehicle.
3. **Automated Valet Parking:** In this use case, an end-user drops an AD vehicle at a car park (Automotive Campus terrain) and the vehicle is then able to park itself in a free parking spot. To leave the car park, the end user can call the car, which will drive itself to the pick-up location. Detection of free parking spots is applied through cameras installed on the premises equipped with smart software. This information is used to determine the vehicle's path planning.
4. **Urban driving:** The goal is to demonstrate the added value of IoT systems and additional information being available to AD vehicles and applications, for improved situational awareness in urban driving.
5. **Car rebalancing:** The objective is to demonstrate the added value of IoT information in the process of AD car rebalancing, i.e. automatically moving a fleet of shared AD cars from a place where they are least needed to a place where they are most needed. In this case, we use information about VRUs to schedule car rebalancing and reroute the cars to avoid VRUs.
6. **Car sharing:** The objective is to enable customers to share a fleet of cars (either self-driving or not) to reach their destinations. A car sharing service finds the closest available car and

assigns it or dispatches it to the customer. Car sharing can also be intended as ride sharing, where multiple customers that possibly have different origins and destinations share a part of the ride on a common car. The Car Sharing use case should be considered as an umbrella use case linked to other use cases like Platooning and AVP.

The following vehicles are being used in Brainport to support the above use cases:

- TNO/Tass: 3 Toyota Prius
- TU/e: 1 Toyota Prius and 3 VFLEX
- VALEO: 1 VW Tiguan
- TT: 1 mobile mapping van
- NEVS: 3 electric vehicle (D-class) (planned)

In cooperation with the Innovatieve Verkeerscentrale and TASS (third-party partners), a simplified Traffic Management functionality is emulated to accommodate information exchange about the A270 conditions. This concerns information (per lane) about lane accessibility and maximum allowable speed.

4.5.1.1 Overall IoT Architecture of the Brainport Pilot Site

Unlike the remaining pilot sites, which have centralised IoT platforms, the Brainport IoT platform is federated and follows the same architecture as the project's IoT platform illustrated in Figure 3.

Four IoT platforms are deployed in Brainport and inter-connected exactly in the same way as in in Figure 3:

1. FIWARE IoT platform, provided by NEC,
2. OceanConnect IoT platform, provided by HUAWEI,
3. Watson IoT Platform, provided by IBM,
4. oneM2M interoperability platform provided by SENSINOV and instrumented and deployed by TNO.

4.5.2 IoT Ecosystem Implementation in Brainport

Figure 17 shows the Brainport IoT ecosystem and the device connectivity.

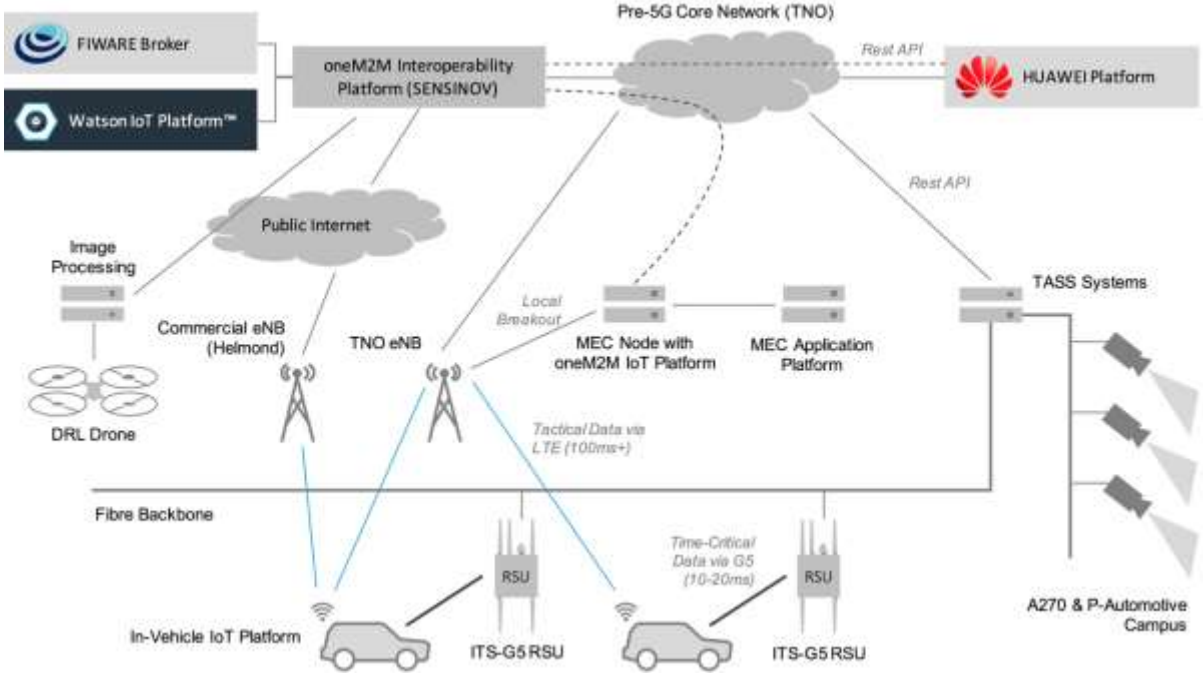

Figure 17 – Integration of IoT Devices into the oneM2M IoT Platform in the Dutch Pilot Site

On highway A270 (Helmond – Eindhoven), 5 kilometres of road are covered by ITS-G5 RSUs and TASS cameras. Every 500 meters, there is one RSU and camera. RSU and cameras are connected by a fibre backbone, reaching the TASS offices. All data from ITS-G5 RSUs and cameras are collected and processed by the TASS back-office servers. The processed camera data is pushed to the oneM2M platform with a minimum update rate to make it available to other vehicles and applications using the A270 trajectory. At the Automotive Campus, several cameras are installed for Valet Parking.

All use cases in the Brainport area will make use of available commercial LTE services (A270, Automotive Campus, TU/e campus site). The TU/e Campus also offers Wifi services. In addition, TNO has deployed an eNB as part of a pre-5G mobile network for R&D purposes, operating in a licence-free band. This eNB which offers a coverage area of a few kilometres supports also NB-IoT, but not yet LTE-V2X (availability is very uncertain at this point). Having our own small scale mobile network allows us to conduct experiments with a Mobile Edge Computing (MEC) node at the road side, running a oneM2M middle node platform instantiation. The latter allows the use of more time critical data gathering and distribution. A specific edge application that makes use of the available IoT functionality is a Local Dynamic Map / Shared World Model application which makes IoT sourced information available to participating vehicles that are entering the LDM service area. The application runs on the MEC node and is fed by the central oneM2M platform and can be considered as a bulletin board to which participating vehicles can subscribe. At this point we do not expect capacity management issues due to connections from vehicles to the MEC node and vice versa.

Vehicles will be equipped with a combination of communication technologies: some will have Ultra-Wide Band (UWB), while all will have ITS-G5 and LTE interfaces. For validation purposes, several VRUs will be equipped with ITS-G5.

The various communication technologies are being used for different purposes as follows:

- **ITS-G5:** Currently, for time-critical communication, only ITS-G5 can support the required latencies. Therefore, ITS-G5 is the protocol used for time-critical communication. Data is communicated using ETSI-G5.
- **LTE:** Vehicles are equipped with LTE modems to connect to commercial LTE services which are widely available. For the sake of the BP pilots, one commercial provider will be selected (purchase of SIM cards). TNO's Hi-5 pre 5G platform provides a specific LTE service which can be subscribed to with a regular LTE interface on the vehicle(s). Depending on the type of data (near-time-critical or non-real-time), data will be processed on the MEC node or in the cloud. The eNB node along the A270 is being deployed at the time of writing, but when it will be, we will perform local breakout at eNB of near-time-critical data. This will be done based on the destination IP address.
- **Fixed Connection:** LTE eNB is connected via fixed (fibre) connection to the pre-5G core network and then to the one2M platform, which is deployed at the premises of TNO The Hague.

4.6 Spain (Vigo)

The Spanish pilot site combines the efforts from PSA, CTAG and Vigo City Council to provide a test environment in the city of Vigo. This test environment will support the use cases of urban driving (UD) and automated valet parking (AVP), which will be developed. Both use cases will be implemented and tested in different parts of the city. The UD use case will be tested in a central street in Vigo, while the AVP will be tested in the public indoors parking of the city council.

The Spanish pilot site includes three vehicles, two contributed by PSA and one contributed by CTAG (PSA branded).

4.6.1 Overall IoT Architecture of the Spanish Pilot Site

The Spanish pilot site ecosystem involves the mentioned vehicles with a oneM2M IoT platform, several IoT devices connected to the IBM Watson IoT platform, a few RSUs working as gateways for the devices and some IoT applications connected to the IBM Watson IoT Platform.

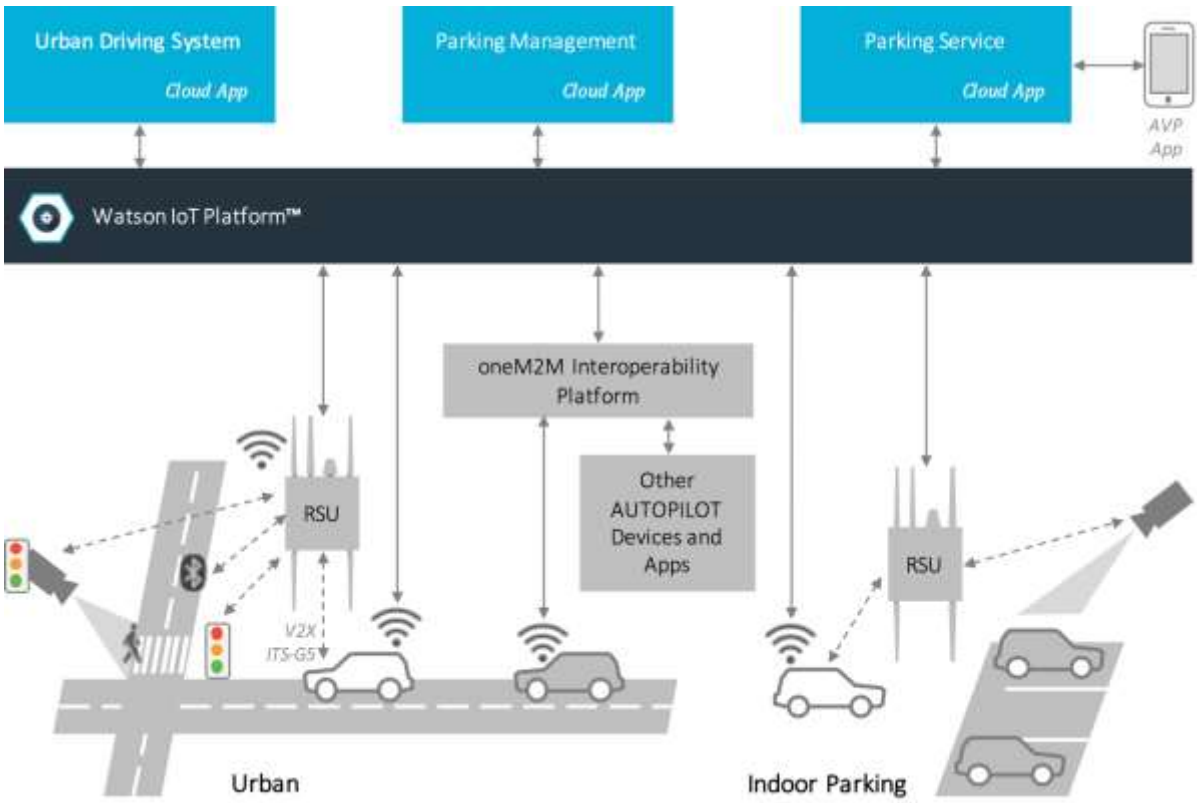

Figure 18 – Spanish Pilot Site Overall Architecture

Figure 18 shows the architecture of the Spanish pilot site and the required components for both urban driving and valet parking use cases. The pilot site devices include: IoT connected cameras, connected traffic lights and a hazard event server that will provide all available information about road conditions as road warnings, traffic jams or accidents.

All these devices will be connected to the IBM Watson IoT platform, providing continuous updates of their status and making these available to the different applications that might make use of them. Some of these applications (urban driving system, parking management and parking service) will be connected to the IBM Watson IoT platform, while others will be connected through the oneM2M connector.

The indoor parking that will be used for the AVP use case will also provide information to the IoT platform, indicating the available parking spots in real time and instructing the routes for the cars to reach their assigned parking spaces.

4.6.2 IoT Ecosystem Implementation in the Spanish Pilot Site

Details of the implemented IoT ecosystem components for each of the Spanish pilot site use cases are provided in the following subsections.

4.6.2.1 Automated Valet Parking Use Case Implementation

The goal of this use case is to provide to the user a service that, through IoT technologies and autonomous car driving functionalities, can automate and improve the way of finding available parking spots and enable the car to automatically park itself once left in a designated drop-off area.

This will be achieved through the following components and functionality.

Pedestrian Detection with Cameras (VRU)

As shown in Figure 18, a camera is used to detect pedestrians or obstacles on the parking site. The information is processed locally and published to the IoT platform and to the RSU.

The in-vehicle IoT platform receives this information and smoothly adapts the speed to the situation or stops completely to avoid any collision.

Parking Management System

As mentioned before, the AVP use case will also provide information about available parking spots and allow these to be booked in advance. This will be performed by the parking management system, which will publish to the IBM IoT platform the status of the parking site, including available spots. Using the DATEX II schema as a reference, all this information can be provided and linked, from the different parking sites to the specific parking spots or groups of parking spots. These data can then be processed and managed by the parking management system, which will enable the booking and management of the Spanish parking site.

Android AVP Application

For this AVP use case, the Spanish pilot site will have an Android application that will be the interface that the user will use to perform the possible actions for AVP, by using the IoT messaging protocols and the provided APIs of the implemented services.

Connected Car with in-Vehicle IoT Platform

A connected car will also act as an IoT device publishing information to the IoT platform. Moreover, the in-vehicle IoT platform will have the required services to translate the IoT messages that command the pickup and drop-off actions of the AVP use case into the needed AD messages to start the manoeuvres.

The in-vehicle platform is further defined in the AUTOPILOT deliverable D2.1.

4.6.2.2 Urban Driving Use Case Implementation

The goal of this use case is to show how IoT can impact the safety of VRUs and the performance of autonomous driving in an urban-like scenario with AD cars, pedestrians at a traffic light crossing, hazards and connected traffic lights.

Pedestrian Detection with Camera (VRU)

As shown in Figure 18, cameras are used to detect pedestrian or obstacles on the lane. The detection information is processed locally and published to the IoT platform and to the RSU.

The in-vehicle IoT platform or the OBU of the AD car receives the information and smoothly adapts the speed to the situation or stops completely to avoid any collision. In case an event is received only by the OBU, it is then published by the in-vehicle platform to the IoT data platform so that it may be processed by other IoT applications in the cloud and made available to other vehicles.

Traffic Light Events

Connected traffic lights send information about the time-to-green/red to the IoT platform. From the IoT platform, the vehicle can receive this information and adjust its speed depending on the traffic light status, stopping when lights are green and moving when they are green.

Moreover, the car can display the traffic light status in its HMI according to the received IoT messages from the traffic lights.

Hazard Events

A traffic control centre, connected to the IoT platform, provides information about different hazard events, such as road works, accidents, etc.

The traffic control centre will send every registered event to the IoT platform, allowing the AD vehicles to access this information and act accordingly, by lowering their speeds or even using alternative routes.

Connected Car with in-Vehicle IoT Platform

Similarly to the AVP use case, the car is always publishing its own status as an IoT device.

5 Interoperability

As already mentioned in Chapter 2, the AUTOPILOT IoT architecture was designed as a federation of IoT platforms, allowing it to be **open** and **flexible**. Developers may plug their own (proprietary) IoT platforms or devices in the architecture and exchange data with existing IoT platforms and devices. As each IoT platform provides a different set of services (features) and may expose a different interface and use a different data exchange protocol, an effort is needed to achieve interoperability while allowing for openness and flexibility.

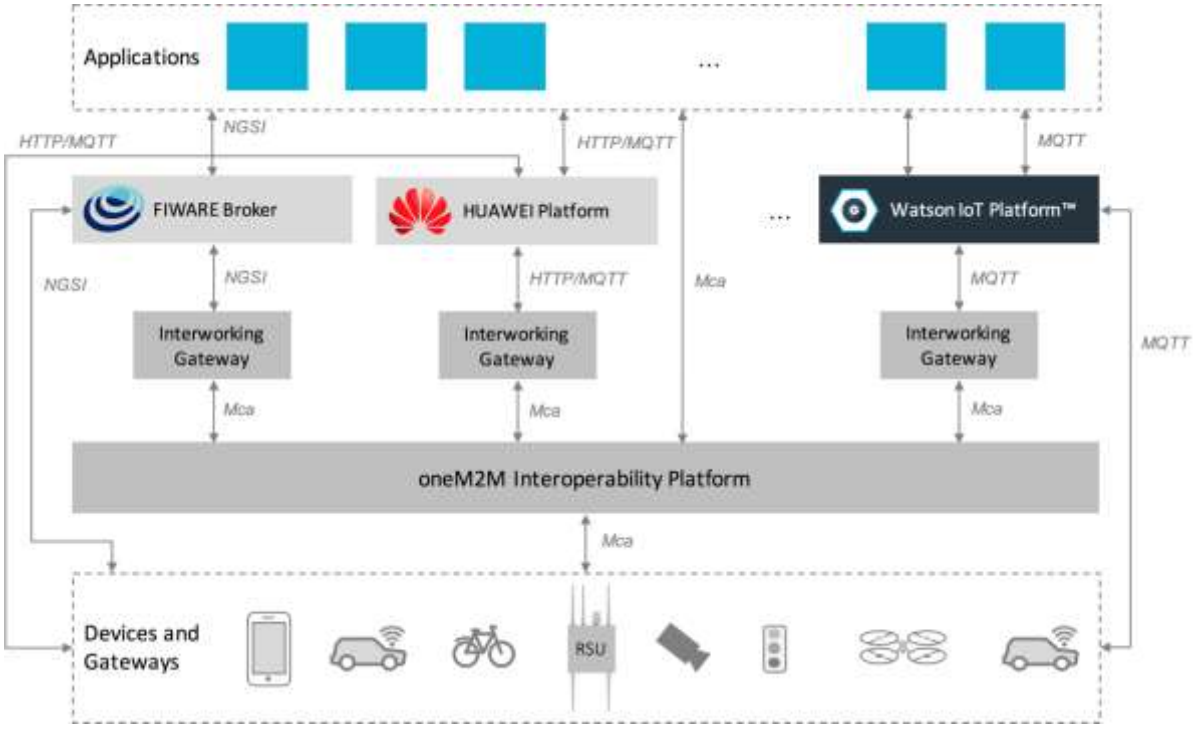

Figure 19 – Autopilot Federated IoT Architecture (Copied from Chapter 2).

Interoperability in AUTOPILOT is achieved based on the following three principles:

- **oneM2M Interoperability Platform and Interworking Gateways:** As shown in Figure 19 (copied from Chapter 2), proprietary IoT platforms are interconnected through interworking gateways and the oneM2M interoperability platform.
- **Standardised IoT Data Models:** IoT data requiring to be exchanged across the IoT platforms are standardised.
- **Standardised Ontologies:** To achieve semantic interoperability, IoT data fields values (e.g. hazard types, vulnerable road user types, detected object types, etc.) are semantically standardised in ontologies.

Each of the above interoperability principles is discussed in the following sections.

5.1 oneM2M Interoperability Platform and Interworking Gateways

An interworking gateway is a oneM2M wrapper to a proprietary IoT platform, allowing it to expose a oneM2M Mca interface and to be connected to the oneM2M interoperability platform. We refer to the gateway between Watson IoT and oneM2M platforms as "Watson-oneM2M Interworking Proxy and to the gateway between FIWARE and oneM2M as "Semantic Mediation Gateway (SMG)" or "Morphing Mediation Gateway (MMG)". MMG is a more dynamic and advanced version of SMG.

The oneM2M platform serves as the bridge for interoperability, allowing data to flow from one IoT platform to another in both directions. Using this architecture, a IoT platform may push data to other IoT platforms and receive from them.

Mapping between the internal data representation of an IoT platform and the oneM2M message contents are specified in the interworking gateways. These also act as filters allowing only selected data to be exchanged.

In this architecture, data providers or consumers, such as applications, may use any of the available IoT platforms according to their requirements. For instance, a data provider may publish data to Watson IoT platform and this data can be shared with FIWARE through the oneM2M platform, so that an application developer who uses FIWARE can access it through the FIWARE platform.

This approach offers flexibility to the pilot sites and application developers. However, for this to work, data providers and consumers need to exchange data with the oneM2Mplatform using standard data models and vocabularies as explained in the following sections.

5.2 Standardised Data Models

oneM2M provides a standard protocol for exchanging IoT messages, but it does not specify the content of the messages as this is domain specific. To achieve interoperability in AUTOPILOT, we are standardising the contents of the oneM2M messages exchanged between the IoT platforms, devices, applications and vehicles, through the oneM2M interoperability platform. A Data Modelling Activity Group (DMAG) was created in AUTOPILOT for this purpose.

The scope of the data model standardisation activity in AUTOPILOT covers the IoT messages and data fields required to implement the project's use cases uniformly across the pilot sites. This will allow AD vehicles to access the same types of data regardless of their locations (pilot sites) and to be able to process the data and work with it. For instance, a message notifying AD vehicles about a hazard on the road, or instructing them to avoid a given road lane, should be the same in all pilot sites, allowing vehicles to consume these messages and react to them correctly as they are moving from one place (e.g. pilot site) to another.

Details of the data model standardisation work are provided in Chapter 6.

5.3 Standardised Ontologies

Data standardisation deals with the structure of the IoT messages exchanged and their field names, types, values and units of measure. This usually works only to a certain extent, as some fields remain challenging to standardise. This is specifically the case when dealing with enumerative fields whose possible values are too numerous to be specified exhaustively in advance, or are language-dependent or use-case-dependent (e.g. vehicle types, detected object types, hazard types, proprietary vehicle parameters, etc.). In such cases, the dilemma is whether to:

- Specify a high-level enumeration that covers all the cases but whose values may turn out to be too broad and useless for certain applications,
- Or leave the field values as plain text, making the data model flexible at the cost of rising semantic interoperability issues.

As can be seen, there are pros and cons for each approach. In AUTOPILOT, we solve the problem by using ontologies that define the common values of the data fields and provide semantic mappings between them. This constitutes a compromise between openness and flexibility, on the one hand and field value standardisation on the other hand.

The AUTOPILOT ontologies are the responsibility of the DMAG. Details of the ontology standardisation work are provided in Chapter 7.

6 Common IoT Data Model

This chapter provides an overview of the IoT data model standardisation work carried out by the AUTOPILOT Data Modelling Activity Group (DMAG). This is work in progress and it will be updated throughout the project lifespan.

For the latest version of the AUTOPILOT data models, please refer to the dedicate GitLab repository at https://gitlab.com/autopilot/iot-data-model and wiki at https://gitlab.com/autopilot/iot-data-model/wikis/home.

6.1 Scope of the Work

It is important to note that it is not the intention of the DMAG to standardise all the data across all the use cases and pilot sites. Rather, the scope of the DMAG work covers only the IoT messages 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, etc.) are beyond the scope of the data modelling activity.

Work of the DMAG is based on reusing and possibly extending, existing standards rather than creating new models.

6.2 Use Case IoT Data Requirements

Use case IoT data models were gathered by NEC from the project partners in a dedicated folder on Project Place at https://service.projectplace.com/#project/1289252236/documents/1412384559. This allowed the DMAG to examine the requirements pertaining to the IoT data model and to select the messages to standardise. The current list of IoT messages under standardisation by the DMAG is provided in Table 10. This table is subject to continuous updates during the lifespan of the project.

Table 10 – AUTOPILOT Use Case IoT Messages Selected for Standardisation

Message Type	Pertaining Use Cases
Vehicle Probe Data (GPS position, speed, status)	Car sharing, AVP
Notifications about detected objects	AVP, highway pilot, urban pilot, platooning, car sharing
Notifications about VRUs	AVP, highway pilot, urban pilot, platooning, car sharing
Notifications about hazards and obstacles	AVP, highway pilot, urban pilot, platooning, car sharing
Notifications about traffic conditions	Highway pilot, urban pilot, platooning, car sharing
Notifications about environmental conditions	Highway pilot, urban pilot, platooning, car sharing
Traffic light states and time to red/green	Highway pilot, urban driving, platooning
Notifications about parking space availabilities	AVP, parking, car sharing
Notifications about charging spot availabilities	AVP, parking, car sharing
Routing instructions	Highway pilot, urban driving, AVP, car sharing
ADAS instructions	AVP, highway pilot, urban pilot, platooning, car sharing
Platoon instructions	Platooning

As can be seen in Table 10, data sources (sensors) are not mentioned here. In fact, we are aiming to standardise the IoT messages regardless of the originating sensor or application.

6.3 Common IoT Data Model

Following the initial IoT data requirement identification phase, the DMAG is currently working on standardising the IoT messages. This is work in progress and subject to changes throughout the

project lifespan. The latest version of the data models and how they are used in AUTOPILOT are available from the GitLab repository at https://gitlab.com/autopilot/iot-data-model.

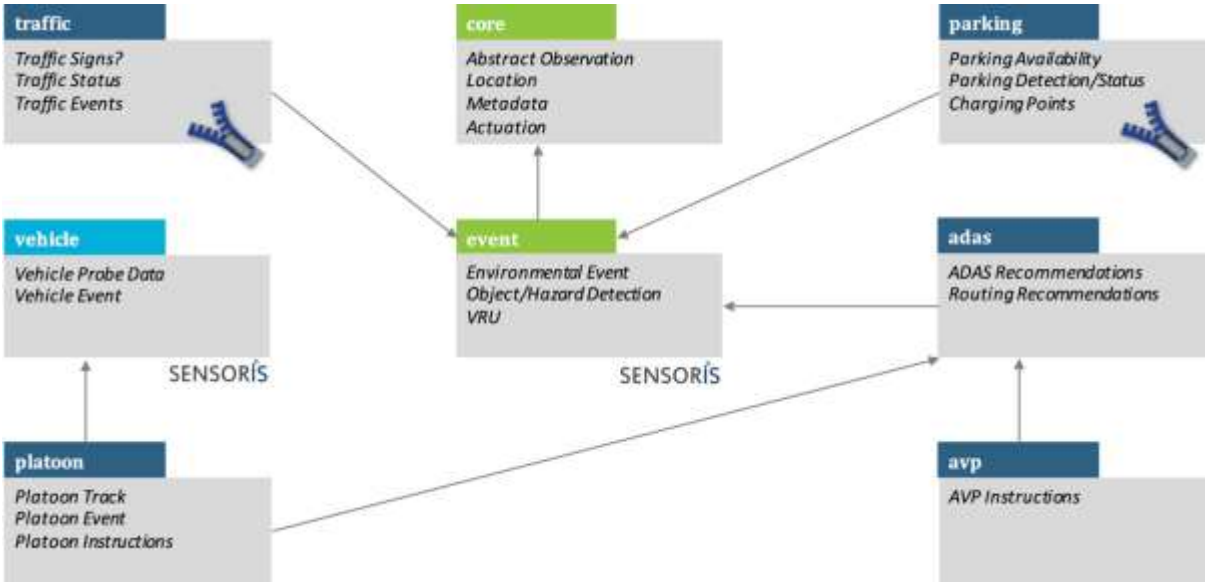

Figure 20 – Overview of the AUTOPILOT IoT Data Model Packages

The AUTOPILOT common IoT data model is split into several packages, based on different standards (e.g. SENSORIS, DATEX II). Figure 20 shows an overview of the current packages being designed and their dependencies. Packages leaders are listed in Table 11.

Table 11 – IoT Data Model Package Lead Partners

Package	Lead Partner
core and event	NEC, Gurkan Solmaz <gurkan.solmaz@neclab.eu>
vehicle	IBM Ireland, Yassine Lassoued <ylassoue@ie.ibm.com>
road side equipment	CNIT, Mariano Falcitelli <mariano.falcitelli@cnit.it>
parking	CTAG, Silvia Alén <silvia.alen@ctag.com>
traffic	CNIT, Mariano Falcitelli <mariano.falcitelli@cnit.it>
adas	?
avp	DLR, Louis Touko Tcheumadjeu <louis.toukotcheumadjeu@dlr.de>
platoon	TNO, Jacco van de Sluis <jacco.vandesluis@tno.nl>

An overview of the packages currently under development is provided in the following subsections.

6.3.1 Vehicle Package

The vehicle package covers the messages sent from the AD vehicles to the IoT platform. This is relevant to the car sharing, AVP and platooning use cases, where vehicles receiving a service need to be tracked. Vehicle data are consumed by the relevant car sharing, AVP or platooning service, but not shared with other services.

The vehicle package is based on the SENSORIS data model [HERE15]. An overview of the elements of a SENSORIS message is provided in Figure 21. Greyed elements will not be supported in AUTOPILOT.

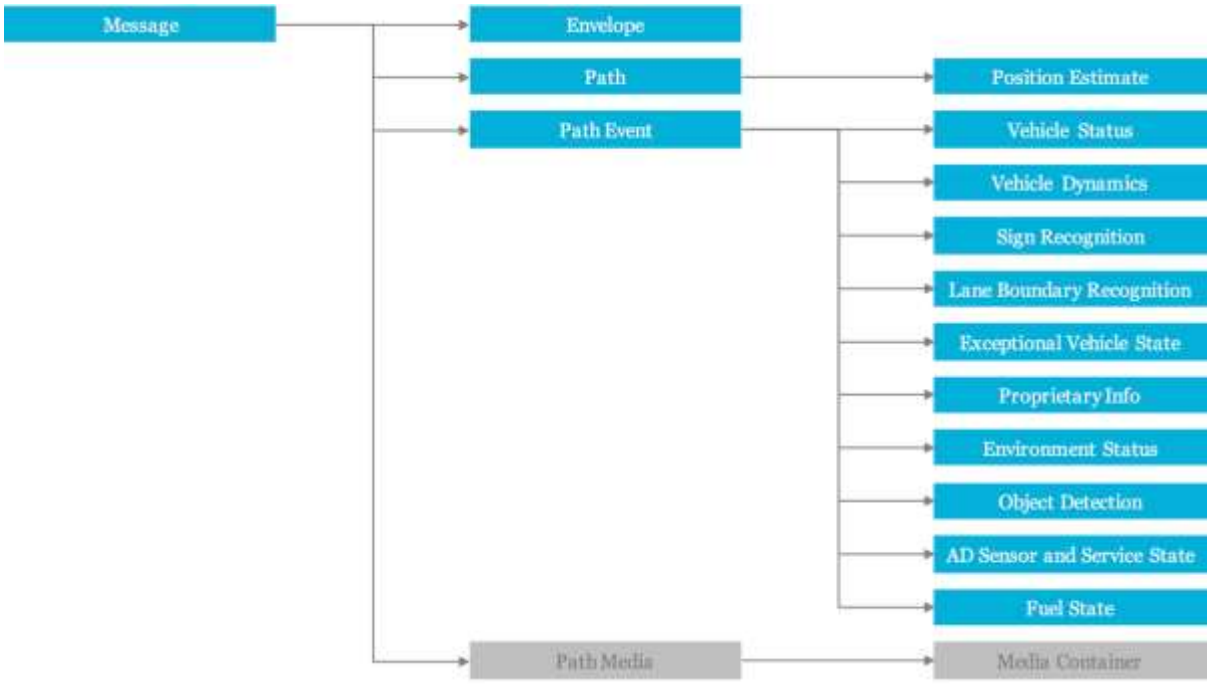

Figure 21 – Overview of the SENSORIS Message Elements

In addition to vehicle GPS data and statuses, SENSORIS messages include events detected by vehicles along their paths, e.g. object detection, environment status, lane boundary detection, etc. SENSORIS event information will be supported in AUTOPILOT, but will be considered as raw data. This means that SENSORIS event messages submitted by vehicles will not be consumed directly by other vehicles. Rather they will be intercepted by event processing services that retrieve the event detection sections from the vehicle messages, analyse them (e.g. for quality check) and republish them as standardised event messages (as specified in the package event).

6.3.2 Parking Package

The parking spot service allows end users to access information about parking spot locations and availability. Operational work of the service requires receiving real-time updates about the states of parking spots. These updates are published through an IoT Platform which allows multiple users to aggregate this information. Information about parking spots include both static and dynamic data. Static parking spot data are the time-invariant properties of the parking spots, such as their geographical locations, shapes, access points, etc. These properties usually remain constant or might change only rarely. Dynamic data represent parking spot properties that are expected to change over time. A typical example of the dynamic information is a parking spot occupation status (occupied or free) that changes as soon as a car enters or leaves the parking spot. Only dynamic information is going to be published through the IoT platform.

Different data models for parking spots are introduced by the transport community:

- FIWARE Parking Harmonized Data Models [FPM18]
- TomTom On-Street Parking [TTP18]
- Datex II Parking Publications Extension [DATEX2]

We selected DATEX II to represent the parking package, since DATEX II is an official multi-part standard, maintained by CEN Technical Committee 278, CEN/TC278, (Road Transport and Traffic Telematics).

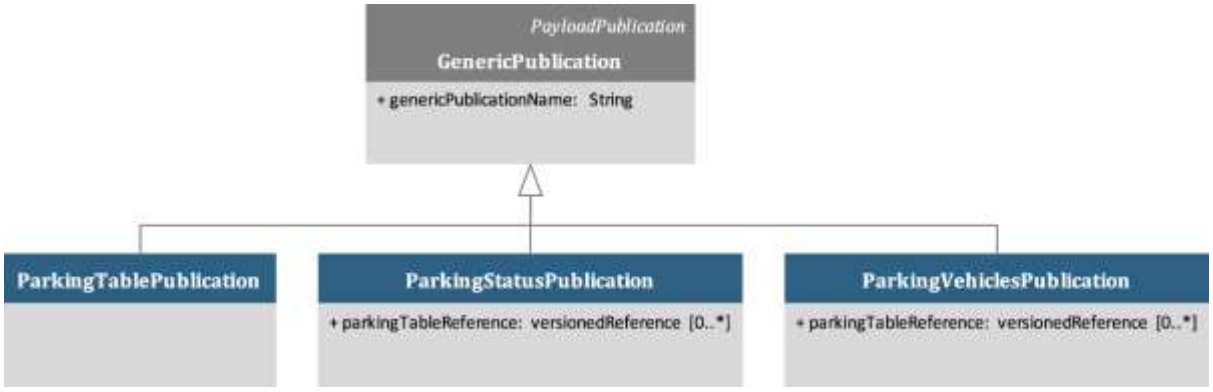

Figure 22 – DATEX II Parking Extension – High-Level Data Elements

Figure 22 shows high-level data elements (classes) of the DATE II parking model extension:

- Element *ParkingTablePublication* contains information about the parking spots (location, basic properties);
- Element *ParkingStatusPublication* is used for operational updates about parking spot occupation;
- Element *ParkingVehiclesPublication* is dedicated to the parked vehicle and is out of scope for our work.

The Parking Table contains static information of parking sites and groups of parking spots. This covers urban or interurban parking areas, with both on-street and off-street parking (e.g. parking garages, parking places, motorway parking, etc.). Within one parking area, information can be specified down to groups of parking spots or even for individual parking spots.

The list below represents the main topics and properties covered by *ParkingTablePublication*:

- Parking allowance for different types of users and vehicles
- Basic parking information (number of parking spots, parking time and fees, etc.)
- Special conditions for specific users or vehicles (e.g. disabled permit, private permit)
- Information about an owner of a parking spot or group (name, address, etc.)
- Geographical location and drop off and pick up points
- Electric charging possibilities including technical characteristics and connection types
- Detailed rate information including specifications for different user / vehicle types, different seasons, payment options and booking fees
- A colour mapping for visualisation of possible occupancy statuses

The *ParkingStatusPublication* element is used for publishing information about parking spot occupancies and other dynamic properties, such as validity of parking spots. We extended the P*arkingStatusPublication* element to support project-specific information, such as confidence levels indicating the reliability of the published status (since this is typically detected by sensors).

6.4 Future Work

The DMAG has proposed a first version of the common IoT data models based on existing data models (SENSORIS and DATEX II) and extensions on them. The group has multiple teams working on different independent packages in parallel, such as the events package, vehicle package, parking package and so on.

The data models will be further refined after the initial feedback from all partners. The data models are supposed to help easier development of interworking gateways that will make automatic translations from one interface to another (e.g. from oneM2M to Watson IoT).

7 Ontologies

Ontologies may be used in AUTOPILOT to provide controlled vocabularies and semantic mappings for the values of IoT data fields to facilitate interoperability between the distributed IoT platforms, services, devices and AD vehicles across pilot sites. This is under consideration, but not planned yet.

Ontology standardisation will be under the responsibility of the Data Modelling Activity Group (DMAG). So far, the DMAG has reviewed existing ontologies with respect to adoption in AUTOPILOT. The review is available in GitLab **iot-ontologies** wiki at https://gitlab.com/autopilot/iot-ontologies/wikis/review-of-ontologies.

The actual standardisation may start early in the third quarter of 2018, following the release of the first version of the data model. Any developed ontologies will be maintained in the GitLab **iot-ontologies** repository at https://gitlab.com/autopilot/iot-ontologies.

Currently, the following data types are considered for ontology standardisation. But these are subject to changes:

- Vehicle types for the SENSORIS model
- AD-specific field names (as SENSORIS proprietary fields)
- Detected object types
- VRU types
- Hazard types
- Environment event types
- ADAS instruction types
- Metadata fields
- Road side equipment types

8 References

[AG15] Amsterdam Group: Signal Phase and Time (SPAT) and Map Data (MAP), white paper, 01 September 2015: https://amsterdamgroup.mett.nl/downloads/handlerdownloadfiles.ashx?idnv=500795. Accessed on 20 February 2018.

[CAM14] ETSI EN 302 637-2 (V1.3.2): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service". November 2014.

[D2.1] D2.1 – Vehicle IoT Integration Report. AUTOPILOT Deliverable. February 2018.

[DATEX2] DATEX II User Guide: https://gitlab.com/autopilot/iot-data-model/uploads/f4c9c67169cad3dc30e0ec444ce7f34a/DATEXuserGuide.pdf. Accessed on 25 February 2018.

[DENM14] ETSI EN 302 637-3 (V1.2.2): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service". November 2014.

[FPM18] FIWRE Data Models, Parking Harmonized Data Models: http://fiware-datamodels.readthedocs.io/en/latest/Parking/doc/introduction/index.html. Accessed on 25 February 2018.

[FW] FIWARE Wiki: https://forge.fiware.org/plugins/mediawiki/wiki/fiware. Accessed on 23 February 2018.

[FWa] FIWARE Architecture Description, IoT Backend, IoT Discovery: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.Architecture Description.IoT.Backend.IoTDiscovery. Accessed on 23 February 2018.

[HERE15] Vehicle Sensor Data Cloud Ingestion Interface Specification (v2.0.2), 2015: https://lts.cms.here.com/static-cloud-content/Company_Site/2015_06/Vehicle_Sensor_Data_Cloud_Ingestion_Interface_Specification.pdf. Last accessed on 25 February 2018.

[HOC] HUAWEI Developers Official Site, OceanConnect: http://developer.huawei.com/ict/en/site-oceanconnect_doc. Accessed on 24 February 2018.

[IBM18a] Getting Started with Watson IoT Platform: https://console.bluemix.net/docs/services/IoT/index.html. Accessed on 07 February 2018.

[IBM18b] Cloudant NoSQL DB: https://console.bluemix.net/catalog/services/cloudant-nosql-db. Accessed on 07 February 2018.

[IBM18c] Watson IoT for Automotive: https://www.ibm.com/support/knowledgecenter/en/SSNQ4V/iot-automotive/overview/overview.html. Accessed on 12 February 2018.

[IBM18d] IBM IoT for Automotive : Vehicle Data Hub: https://developer.ibm.com/api/view/id-551:title-IBM_IoT_for_Automotive___Vehicle_Data_Hub. Accessed on 12 February 2018.

[IBM18e] IBM Watson IoT Context Mapping: https://developer.ibm.com/api/view/id-263:title-IBM_Watson_IoT_Context_Mapping. Accessed on 12 February 2018.

[ISOTC204] ISO/TC 204, Intelligent Transport Systems: https://www.iso.org/committee/54706.html. Last visited on 20 February 2018.

[ONE17] oneM2M - Standards for M2M and the Internet of Things: http://onem2m.org/. Accessed on 13 February 2018.

[OIP17] Ocean IoT platform: http://www.iotocean.org/. Accessed on 23 February 2018.

[OSGi] OSGi Alliance official website, https://www.osgi.org . Accessed on 27 February 2018.

[MQTT] MQTT official website, http://mqtt.org. Accessed on 25 June 2018.

[NGSI10] FI-WARE NGSI-10 Open RESTful API Specification: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI-10_Open_RESTful_API_Specification. Accessed on 23 February 2018.

[NGSI9] FI-WARE NGSI-9 Open RESTful API Specification: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI-9_Open_RESTful_API_Specification. Accessed on 23 February 2018.

[TS09] oneM2M Technical Specification: TS-0009-V2.6.1 – HTTP Protocol Binding: http://www.onem2m.org/images/files/deliverables/Release2/TS-0009-HTTP_Protocol_Binding-V2_6_1.pdf. Accessed on 25 February 2018.

[TS10] oneM2M Technical Specification: TS-0010-V2.4.1 – MQTT Protocol Binding: http://onem2m.org/images/files/deliverables/Release2/TS-0010-MQTT%20Protocol%20Binding-V2_4_1.pdf. Accessed on 25 February 2018.

[TS08] oneM2M Technical Specification: TS-0008-V1.0.1 – CoAP Protocol Binding: http://onem2m.org/images/files/deliverables/TS-0008-CoAP_Protocol_Binding-V1_0_1.pdf. Accessed on 25 February 2018.

[TTP18] TomTom On-Street Parking: https://developer.tomtom.com/street-parking/street-parking-api-draft-one-page. Accessed on 25 February 2018.