Federating IoT and cloud infrastructures to provide scalable and interoperable Smart Cities applications, by introducing novel IoT virtualization technologies

EU Funding: H2020 Research and Innovation Action GA 814918; JP Funding: Ministry of Internal Affairs and Communications (MIC)

Deliverable 5.1
Pilot design and validation methodology - First Release

Deliverable Type:	Report
Deliverable Number:	5.1
Contractual Date of Delivery to the EU:	30.06.2019
Actual Date of Delivery to the EU:	30.06.2019
Title of Deliverable:	Pilot design and validation methodology - First Release
Work package contributing to the Deliverable:	WP5
Dissemination Level:	Public
Editor:	Antonio Skarmeta, Kenichi Nakamura
Author(s):	Antonio Skarmeta (OdinS), Juan Antonio Martinez (OdinS), Juan Andres Sanchez (OdinS), Andrea Detti (CNIT), Giuseppe Tropea (CNIT), Paolo Fattoruso (CNIT), Frank Le Gall (EGM), Pascal Pellizoni (EGM); Hidenori Nakazato (WAS); Kenji Kanai (WAS); Kenichi Nakamura (PAN); Testuya Yokotani (KIT); Hiroaki Mukai (KIT)
Internal Reviewer(s):	Andrea Detti (CNIT)
Abstract:	This deliverable presents the most relevant information obtained from the design phase of the pilots. In this sense, we present the whole path from the available information, by specifying also its structure, to the required process to transform this information into virtual things, thanks to the use of ThingVisors, to the API and Virtual Silos to be used by each of the pilots, to the expected information the application and services will consume. Additionally, we identify performance parameters, so as to help validating our whole Fed4IoT architecture. Finally, we examine legislation and privacy concerns about the pilots.
Keyword List:	Domain modelling, Silo, ThingVisor, NGSI-LD

Disclaimer

This document has been produced in the context of the EU-JP Fed4IoT project which is jointly funded by the European Commission (grant agreement n 814918) and Ministry of Internal Affairs and Communications (MIC) from Japan. The document reflects only the author's view, European Commission and MIC are not responsible for any use that may be made of the information it contains

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Definition
API	Application Programming Interface
CIM	Context Information Management
ETSI	European Telecommunications Standards Institute
HTTP	HyperText Transfer Protocol
ICN	Information Centric Networks
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MQTT	Message Queue Telemetry Transport
NGSI	Next Generation Service Interfaces Architecture
NGSI-LD	Next Generation Service Interfaces Architecture - Linked Data
OMA	Open Mobile Alliance
REST	Representational State Transfer
SDK	Software Development Kit
TV	ThingVisor
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VNF	Virtual Network Functions
vSilo	Virtual Silo
vThing	Virtual Thing
WLAN	Wireless Local Area Network
RPZ	Regulated Parking Zones
OCB	Orion Context Broker
GDPR	General Data Protection Regulation
PIA	Privacy Impact Assessment

Table 1: Abbreviations

Fed4IoT Glossary

Table 2 lists and describes the terms that have been considered relevant in this deliverable.

Term	Definition
FogFlow	An IoT edge computing framework that automatically orchestrates dynamic data processing flows over cloud- and edge-based infrastructures. Used for development of ThingVisors
Information Centric Networking	New networking technology based on named contents rather than IP addresses. Used for development of ThingVisors
IoT Broker	Software entity responsible for the distribution of IoT information. For instance, Mobius and Orion can be considered as Brokers of oneM2M and FIWARE IoT platforms, respectively
Neutral Format	IoT data representation format that can be easily translated to/from the different formats used by IoT brokers
Real IoT System	IoT system formed by real things whose data is exposed through a Broker
System DataBase	Database for storing system information
ThingVisor	System entity that implements Virtual Things
VirIoT	Fed4IoT platform providing Virtual IoT systems, named Virtual Silos
Virtual Silo (new name for IoT slice in D2.1)	Isolated virtual IoT system formed by Virtual Things and a Broker
Virtual Silo Controller	Primary system entity working in a virtual Silo
Virtual Silo Flavour	Virtual silo type, e.g. a "Mobius flavour" is related to a virtual silo with Mobius broker, a "MQTT flavour" refers to a virtual silo with MQTT broker, etc.
Virtual Thing	An emulation of a real thing that produces data obtained by processing/controlling data coming from real things
Tenant	User that accesses the Fed4IoT VirIoT platform to develop IoT applications

Table 2: Fed4IoT Dictionary

1 Introduction

1.1 Deliverable Rationale

This report collects all the information we have considered in the design phase of each of the proposed pilots. This information ranges from the information models available at the source, to the data model required for each of the pilots from the point of view of the services or applications they end up providing.

1.2 Quality Review

The internal Reviewer responsible of this deliverable is CNIT.

Version Control Table			
V.	**Purpose/Changes**	**Authors**	**Date**
0.1	ToC	Juan A. Martinez (OdinS)	09/05/2019
0.2	ToC update	Juan A. Martinez (OdinS)	04/06/2019
0.3	Initial Version	All Authors	15/06/2019
0.4	New section 2	Andrea Detti (CNIT)	20/06/2019
0.5	Section 1	Juan A. Martinez, Juan A. Sanchez (OdinS)	21/06/2019
0.5	Document Edition, Generating Annex	Juan A. Martinez, Juan A. Sanchez (OdinS)	24/06/2019
0.6	Document revision and regulation (GDPR) section	Andrea Detti, Giuseppe Tropea (CNIT)	25/06/2019
0.7	Review of Concept and Services sections	Andrea Detti (CNIT)	26/06/2019
0.8	Consistency check	Giuseppe Tropea (CNIT), Juan A. Martinez (OdinS)	27/06/2019
1.0	Final review	Andrea Detti (CNIT)	28/06/2019

Table 3: Version Control Table

1.3 Executive Summary

1.3.1 Deliverable Description

This deliverable is the result of Task 5.1: Pilot Design and validation methodology, of this project. Within the scope of this task all of the partners present the design process for each of their pilots, as well as a list of different performance parameters specific for each of them, which will allow us to validate our architecture.

In a thorough analysis, firstly we identify the information forming the *Root Data Domain* (see Deliverable D2.2). This is the information available by the cities of Murcia,

Grasse, Hakusan and Kumamoto. Secondly, for each of the pilots we follow an approach where we specify: (a) the needed Applications and Services; (b) the requirements for each of them in terms of the representation of information; (c) the required interfaces and API; (d) how ThingVisors must process the information of the Root Data domain in order to come up with Virtual Things matching these requirements.

In a different section we elaborate a preliminary list of performance parameters, which allow us to validate our pilots, as well as legislation and privacy concerns raised by the architecture, lawful processing and compliance to the GDPR of the pilots.

1.3.2 Summary of Results

In this deliverable we report:

1) the design process of each of the pilots;

2) the identification of different performance parameters per pilot, which will allow us to validate our architecture;

3) a Privacy Impact Assessment of the most difficult pilots for what concerns compliance with the GDPR, which helps us validate the pilots from the lawful and privacy perspective.

Concerning point 1), we highlight, from the infrastructure point-of-view, the federated Root Data Domain of our test-beds, where we describe the available information provided by each of the scenarios by different cities. From the other point of view, which is the pilots' consumer's one, i.e. the point of view of the applications to be fabricated on top of the virtualized infrastructure, we make a proposal regarding the expected information. This task then considers the transformation that the information coming from the federated Root Data Domain must undergo in order to be consumed by those applications and services.

Concerning point 2), we define a list of performance parameters per pilot, paving the way that allows us to validate our architecture. This is a preliminary list since the project is at an early stage (Y1), in which we have defined the supporting architecture, but not yet fully implemented it. The final list of target performance parameters (aka, KPIs) will be finalized in the second release of this deliverable, when pilots' deployment will start, thus making it possible to better identify challenging and realistic performance indicators, to be measured on the field.

Concerning point 3), we find out that the most challenging pilots, for what concerns compliance with the GDPR, are the cross-border person finder and the waste management ones. For these two pilots we carry out a draft PIA and we draw conclusions that help us validate them in terms of lawfulness and user acceptance when it comes to requesting consent to processing of personal data.

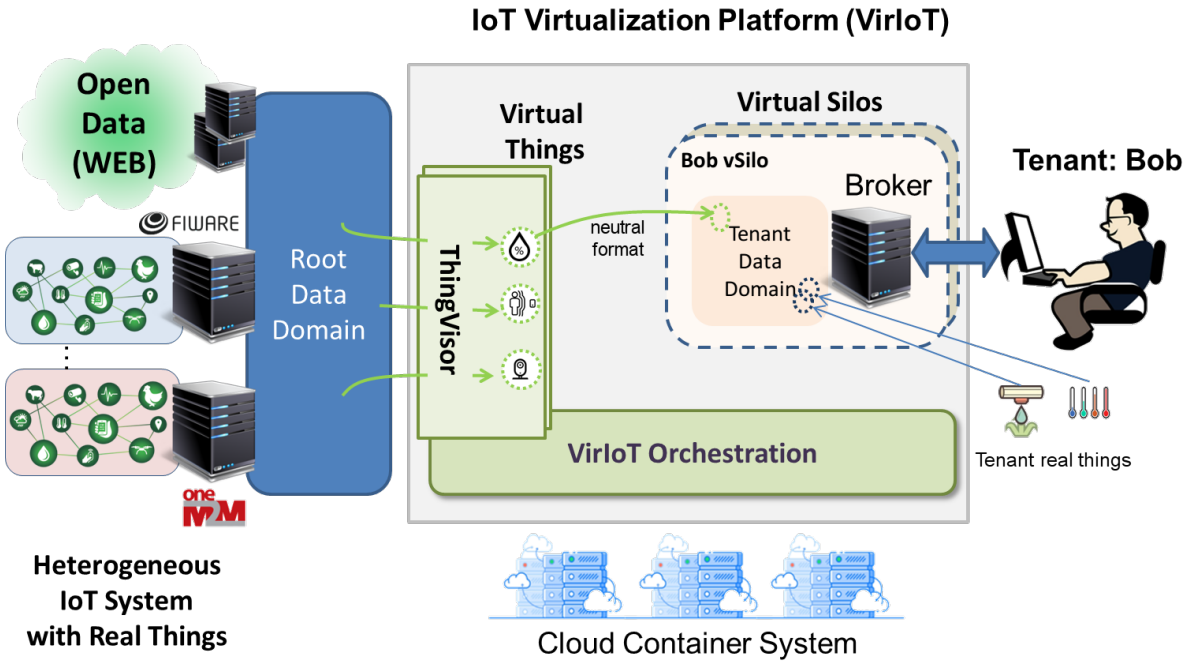

Figure 1: Fed4IoT VirIoT Platform

2 Overview and Concept

2.1 Virtualization Platform

Figure 1 visualizes the main concepts behind the architecture of the Fed4IoT virtualization platform, namely VirIoT, introduced in Deliverable 2.2. On the left, we have different IoT Systems exposing their information through heterogeneous interfaces, including IoT brokers compliant with IoT standards such as NGSI, NGSI-LD, oneM2M, etc., but also non-standard interfaces e.g. based on HTTP Rest proprietary API, or raw MQTT data streams. This information forms the *Root Data Domain* from which VirIoT gathers information in turn.

VirIoT *ThingVisors* use information of the Root Data Domain to create *Virtual Things*, i.e. emulations of real things producing data obtained by processing/controlling information coming from the Root Data Domain (see Figures 2 and 3).

As shown in the right-hand side of Figure 1, Tenants can create isolated IoT environments, dubbed *Virtual Silos* (vSilos), where on-demand data produced by Virtual Things is added and exposed through the broker technology of choice, by the tenant. For instance, a tenant can create a FIWARE vSilo that offers the Virtual Things' data through an internal Orion FIWARE Broker, which is of exclusive use of the tenant. Another tenant can create a oneM2M vSilo in which the same Virtual Things data is exposed by a dedicate Mobius oneM2M Broker.

Accordingly, the VirIoT platform copes with the heterogeneity issue adapting the generated IoT data to the format preferred by the tenant. This adaptation is made in two steps: i) the ThingVisors output Virtual Thing data in a common internal neutral data format (actually NGSI-LD, see Figure 3); ii) an internal vSilo Controller "translates"

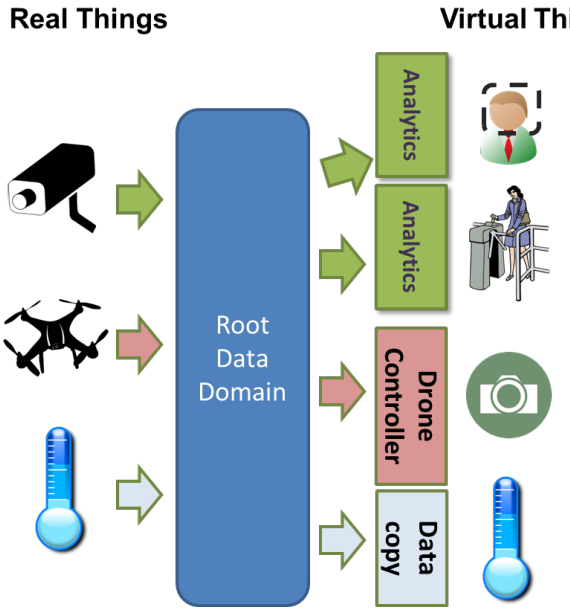

Figure 2: Virtual Things (vThings)

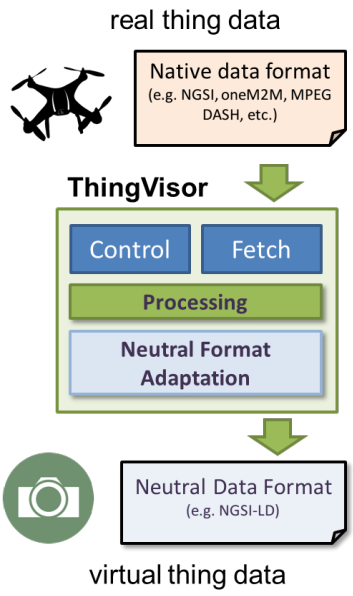

Figure 3: ThingVisor

the neutral data format to the format used by the vSilo Broker.

Just like in a traditional cloud infrastructure-as-a-service the tenant can choose the operative system, in Fed4IoT a tenant can choose the IoT Broker technology. Further, just like in a cloud infrastructure-as-a-service the tenant can choose the virtual hardware (number of CPUs, disk size, etc.) of her virtual server, in Fed4IoT the tenant can choose the Virtual Things of her IoT Virtual Silo. **That is actually exactly the novel IoT virtualization concept the project is pursuing**.

The VirIoT platform is used for Fed4IoT testbeds with a twofold objective:

- data federation: to collect IoT information coming from heterogeneous IoT deployments available in Murcia, Grasse, Hakusan, and Kumamoto;
- applications' development: to provide Fed4IoT testbed Applications with Virtual Silos containing necessary virtual thing data.

Accordingly, this deliverable describes the data forming the Fed4IoT Root Data Domain in Section 4, and how Service and Virtual Silos offered by the platform are used by the Fed4IoT testbed applications in Section 5. The internal aspects of the Fed4IoT platform will be described in the deliverable of WP3 and WP4.

3 Methodology

This section elaborates on the plan we devised in order to validate our architecture through the deployment of the different pilots.

As already described in Deliverable 2.2 [1] the architecture we designed inside Fed4IoT is quite complex, since it is capable to dynamically generate virtual things on behalf of its clients depending on different requests, credentials, and access level.

This architecture counts three main entities that are responsible for carrying out such a task. Firstly, ThingVisors. Specific components are responsible for transforming the information obtained at origin into the neutral format based on NGSI-LD, and that are able to create more complex virtualised things (sensors) based on aggregation techniques, for instance. Secondly, vSilos. They are the components responsible for transforming the information the other way round. Now, the neutral format is converted to the specific format desired by the clients/tenants. Finally, the Master Controller coordinates the deployment of these components, orchestrating the whole process.

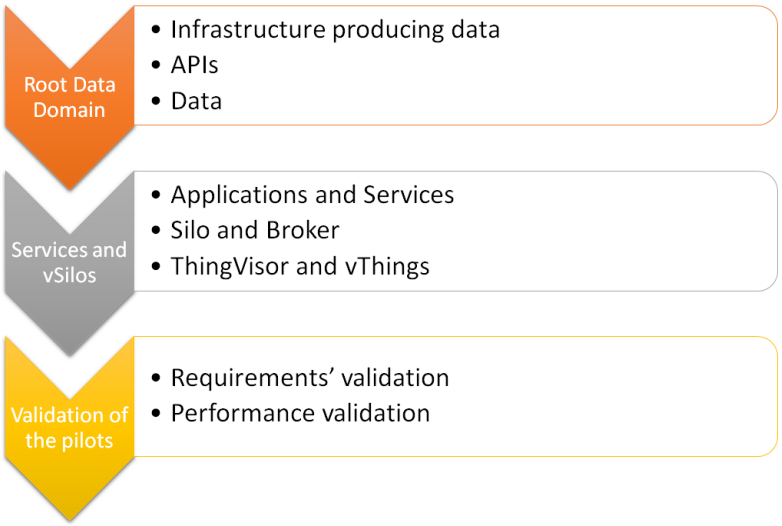

Figure 4: Methodology

Fed4IoT defined four different pilots which address different domains spanning a variety of classical IoT and societal problems, such as finding free parking spots in the city, correctly managing city waste, monitoring of wildlife for protection of people and efficient management of wild animals, and finding a missing person throughout several geographical areas.

The methodology followed by each one of the pilots so as to validate our architecture comprises the following phases, as depicted in 4.

- In the first phase, for each site belonging to the pilot, we have to describe the information originally **available**. This way the variety of information formats, as well as technologies, are analyzed and unmasked.
- In the second phase, from the point of view of the consumers of that information, we find out the **required** information, as well as the specific requirements in terms of representation format.

- Finally, for each one of the use cases, we refer to the set of requirements we have identified in deliverable D1.2, and we plan to validate the pilots with specific functional and non-functional tests, and by also measuring some performance parameters (KPIs), which allow us to validate our architecture from that specific point of view.

This methodology is embodied in the structure of this Deliverable, where the following sections corresponds to each one of the phases of this methodology.

4 Root Data Domain for Federated Testbeds

This section presents the information provided by Murcia, Grasse, Hakusan, and Kumamoto, thanks to the collaboration of the partners of this consortium. As we will see the resulting Root Data Domain comprises several heterogeneous sources.

4.1 Murcia

Thanks to the commitment of the City Council of Murcia at enhancing the quality of living of its citizens and its goal regarding the optimal management of their current resources, the city of Murcia was awarded in the Smart City tender, with one of the most ambitious Smart City projects.

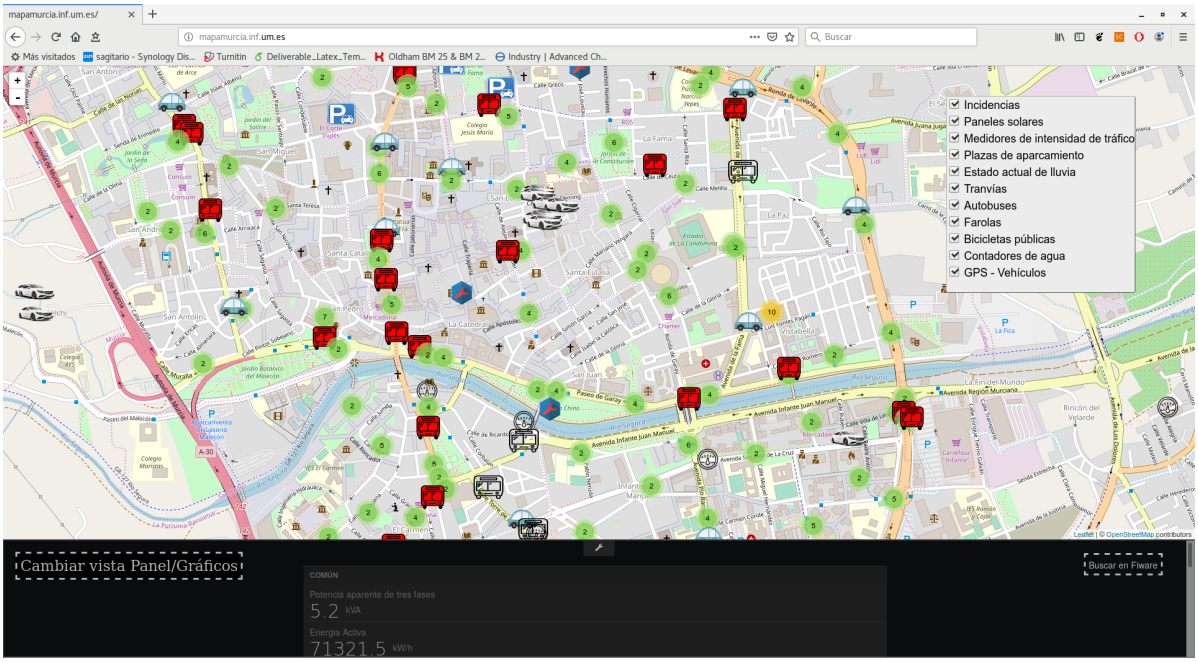

Figure 5: Map of the city of Murcia with services integrated

Before its beginning, thanks to the collaboration of the City Council, the University of Murcia and Odins, a first and limited FIWARE-based smart city platform was deployed with the objective of proving the integration of heterogeneous sources of information. It counts with an instance of Orion Context Broker which receives information from different services such as parking sites, traffic information, solar panel information, trams and bike rental, to name a few, as can be seen in Figure 5. This information is stored thanks to the NGSIv2 representation which also allows for a uniform way for accessing the stored information.

Selecting from the whole set of context sources stored in the platform, the information from the availability of parking sites, as well as the one coming from Regulated Parking Zones are the ones that will be exploited in the Smart Parking pilot.

4.1.1 Infrastructure producing data

Producing information about parking site and Regulated Parking Zone availability requires a specific deployment for each of the domains.

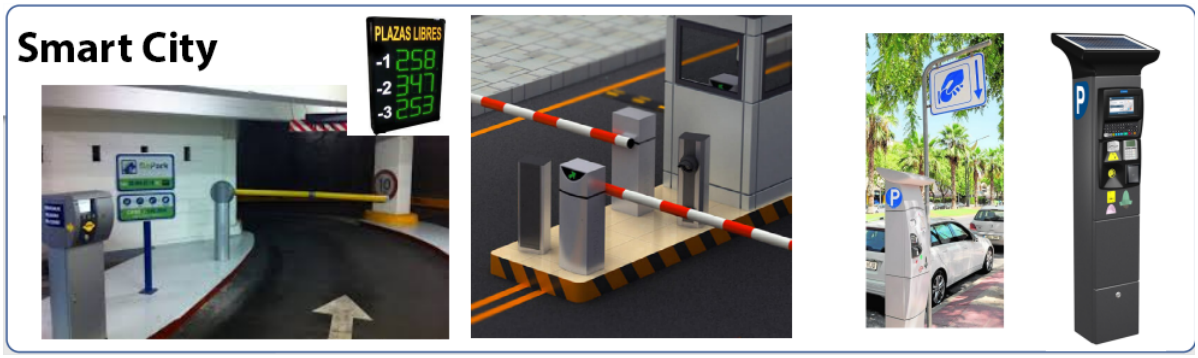

Figure 6: Infrastructure required for providing parking site availability

Figure 6 sheds some light onto the specific equipment required for such information provisioning. For what concerns the domain of parking site availability, they are access control barriers, induction loops, as well as an IoT device or controller able to receive the information from the sensors and actuate over the barrier. The information stored in the IoT device must communicate the information to a central server. To do so, it can employ different wireless communication technologies such as LoRa or WiFi for instance. It can also use a wired connection to the server.

For the case of Regulated Parking Zones (RPZ), parking-meters are the devices responsible for issuing tickets for their corresponding sector or area. They must transmit all the ticketing information to a central server. Nowadays, Murcia city relies on old devices that transmit all the gathered information by each parking-meter at the end of the day, taking advantage of the fact that during night the corresponding RPZ is free. Regarding communication technologies, there exist different alternatives such as using optical fiber or wireless communications like LoRa, NBIoT or WiFi.

All this information is finally received by the Orion Context Broker, which stores it following the NGSIv2 representation and exposes them to the ThingvVsors of the Fed4IoT Architecture (fig. 1).

4.1.2 APIs

Orion Context Broker (OCB) has an REST API, which is based at NGSI interface, in our case NGSIv2, and promoted by Open Mobile Alliance (OMA). This REST API allows reading, writing, subscribing, etc... operations in a simple way.

In the use case, NGSIv2 offers two options to access to OCB's entities data. The first option is using entity queries, the second one is via the subscription mechanism.

- Querying entities: in order to send entity queries to OCB, using NGSIv2 API, a GET request to entities method must be sent. This request receives "offset" and "limits" parameters.

These parameters allow determining the number of entities to be received in the response, so that the request could be sent several times in a row, until obtaining all of the OCB entities.

- Entity subscription: NGSIv2 API allows a subscription mechanism to obtain OCB's entities data. In this way, when an OCB entity is created or updated, the system sends a notification, in JSON format, to all of the subscribers that are monitoring this entity or any of its attributes.

4.1.3 Data

As indicated in 4.1.1, a provider OCB stores the information that IoT devices send to it.

Provider OCB also stores additional context information associated with the use case, for example, the location of IoT devices or other entities or entity attributes with an informative character or configuration ones. This said , Murcia's provider OCB has the following entity types that will be exposed to the Root Data Domain:

Type	Description
parkingsite	This entity contains a harmonised description of a parking site.
policy	This entity contains a harmonised description of a parking site policy.
sector	This entity contains a harmonised description of a Regulated Parking Zone sector.
parkingmeter	This entity contains a harmonised description of a Regulated Parking Zone parking meter.
ticket	This entity contains a harmonised description of a Regulated Parking Zone parking meter ticket.

Table 4: Smart parking entity types

Although in the following subsections these entities are defined briefly, we provided a thorough view of each of them in the Annex.

Parking Site

This entity contains a harmonized description of a parking site. This entity has attributes to define:

- The number of total and free parking site spaces, considering the type of vehicle (motorcycle or private car), disabled persons and EV vehicles.
- Maximum admittable vehicle dimensions.
- Accepted payment methods.
- Policies, considering the type of vehicle (motorcycle or private car) and public holidays. The policy attributes are related to the policy entity to define the parking site policy and rate.

- Open hours, considering weekday, weekend and public holidays.
- Services (carwash, valet, EV chargers).
- Contacts and address.

Policy

This entity contains a harmonized description of a parking site policy. This entity has attributes to define:

- The time period when the parking site policy will be applied (with their corresponding rates). The rate could be different considering the duration, vehicle type, day of the week, public holidays and disabled persons.
- If public holidays will apply the parking site policy.
- Grace period, currency rates, the maximum duration of the parking site.

Sector

This entity contains a harmonized description of a Regulated Parking Zone sector. This entity has attributes to define:

- The total and the theoretical number of free parking site.
- Policies, considering the public holidays.
- The geolocation polygon of the sector.

Parking meter

This entity contains a harmonized description of a Regulated Parking Zone parking meter. This entity has attributes to define:

- Related sector.
- The geolocation point of the parking meter.

Ticket

This entity contains a harmonized description of a Regulated Parking Zone parking meter. This entity has attributes to define:

- Related parking meter.
- Booked period and duration.
- Rate, price and payment method.

4.2 Grasse

4.2.1 Infrastructure producing data

The experimental Grasse network producing the data for the Root Data Domain builds upon the LoRaWAN technology which is a low power long-range radio technology for the internet of things. This technology imposes constraints on the maximum available bitrate as it makes use of the 868MHz ISM radio band which has a regulated 10% duty cycle. The available bitrate is in the order of magnitude of 125kbits/s. Overall, only a few hundred of bytes can be exchanged per hour.

The different sensors deployed over the territory send their information over the LoRaWAN radio link. These packets are captured by LoRaWAN gateways, which forward them to a cloud Lora server. This server deciphers the received payloads and transmits them to an application server in charge of decoding the payloads to produce SenML structures, serialized in JSON and transmitted over MQTT. These packets are then captured by a oneM2M IoT layer which maps the SenML payloads to a oneM2M resource tree, exposed by a oneM2M Mobius broker to the ThingVisors of the Fed4IoT architecture (fig. 1). To allow for automated building of the oneM2M resource tree, each device is having its own Application Entity (AE) within which each measured data is stored in a container holding the unit and the measured values with their timestamps.

4.2.2 APIs

To ease interactions with the oneM2M platform, a python based API has been built to create, read, subscribe to the different instances.

Listing 1: Parameters for Python based oneM2M API

```
{
### Ressources types ###
### AE ty=2 ###
### Container ty=3 ###
### ContentInstance ty=4 ###
### Subscription ty=23 ###

#ACP (Addition for right attribution)
#63 gives CREATE, RETRIEVE,
          UPDATE, DELETE, DISCOVERY and NOTIFY rights
#Value         Interpretation
#1             CREATE
#2             RETRIEVE
#4             UPDATE
#8             DELETE
#16            NOTIFY
#32            DISCOVERY

### Notification TYPE for Subscription ###
#Net = notificationEventType
# 1     Update_of_Resource
# 2     Delete_of_Resource
```

```
# 3      Create_of_Direct_Child_Resource
# 4      Delete_of_Direct_Child_Resource

#    API CALLs EXAMPLES
# python oneM2M_basics.py createACP http://127.0.0.1:8080 egmACP egm:franck 63
# python oneM2M_basics.py createAE http://127.0.0.1:8080 egm.camera Camera1 [/in-cse/acp-609336319]
# python oneM2M_basics.py createContainer egm:franck http://127.0.0.1:8080/in-name/ObservedArea ObservedArea1
# python oneM2M_basics.py createContentInstance egm:franck http://127.0.0.1:8080/in-name/Camera1/ObservedArea1 {geopolygon}
# python oneM2M_basics.py getContentInstanceLatest egm:franck http://127.0.0.1:8080/in-name/Camera1/ObservedArea1
# python oneM2M_basics.py createSubscription egm:franck http://127.0.0.1:8080/in-name/Camera1/ObservedArea1 CamerasSubs http://127.0.0.1:6000 1
# python oneM2M_basics.py deleteContentInstanceLatest egm:franck http://127.0.0.1:8080/in-name/Camera1/ObservedArea1
# python oneM2M_basics.py deleteContainer egm:franck http://127.0.0.1:8080/in-name/Camera1/ObservedArea1
# python oneM2M_basics.py deleteAE egm:franck http://127.0.0.1:8080/in-name/Camera1

}
```

4.2.3 Data

The planned experiment involves the development of smart cameras able to capture wild deposits of waste in certain areas of the city. This service is actually done through automated cameras whose picture are stored in a local Secure Digital card and processed manually from time to time. This process is manpower intensive and implies a delay in the treatment of the offense.

A more automated and real-time reactivity is thus wished. The proposed approach is to apply *machine learning techniques* upon the images taken by a video camera. However, the deployment of cameras in public spaces is a complex process as a number of authorizations have to be asked to ensure that private information is being treated confidentially. In the present case, it is wished to be able to change the place of the cameras from time to time to avoid degradation. Thus, to limit complications in respect to data privacy as well as reduce the requirements for high bandwidth connectivity in all of the places, local processing (edge processing) of the image is wished. Doing so, only limited information is being sent. It includes a timestamp, geolocation, type of event detected and car plate number of the car. The series of pictures taken remain stored locally. In addition, offenders are identified through the plate of their vehicle whereas faces need to be blurred.

Finally, the solution has to run autonomously from a local energy source (*solar power harvesting*) and thus has to use control strategies optimizing energy consumption. For that reason, sleep modes have to be implemented with wake-up of the system on movement detection being initiated by the camera or an additional sensor such as the lidar one.

The infrastructure thus includes on site:

EGM.png

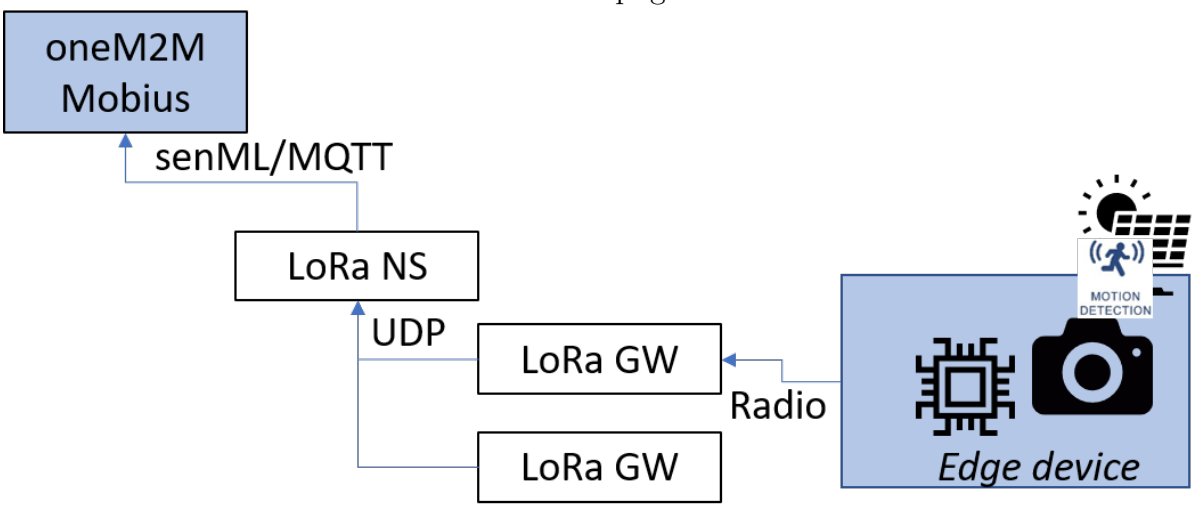

Figure 7: Data collection infrastructure

- A video camera
- A movement detector (possibly provided by the camera)
- A processing unit able to locally run image processing algorithms
- A LoRaWAN interface

A detected movement triggers picture captures which are then processed. In case a wild waste deposit is detected and an associated car plate is visible, the event is recorded and information is sent to the cloud over LoRaWAN.

The core of the associated data model (see figure 8 and table 17) considers mainly 3 entities being the camera, the site to be monitored and the detected wild deposit. They are detailed in the forthcoming sections.

Waste Management entities	
Entity Name	**Description**
Camera	This entity contains a harmonized description of the camera, its capabilities and status. See Section 9.2.1
Sensitive Site	This entity contains a harmonized description of the site top be monitored for wild deposits. See section 9.2.2
Wild Deposit	This entity describes the detected wild deposit. See section 9.2.3

Table 5: Waste Management use case entities

All the details regarding these entities have been placed into the Annex, where you can find a thorough view of the representation of this information.

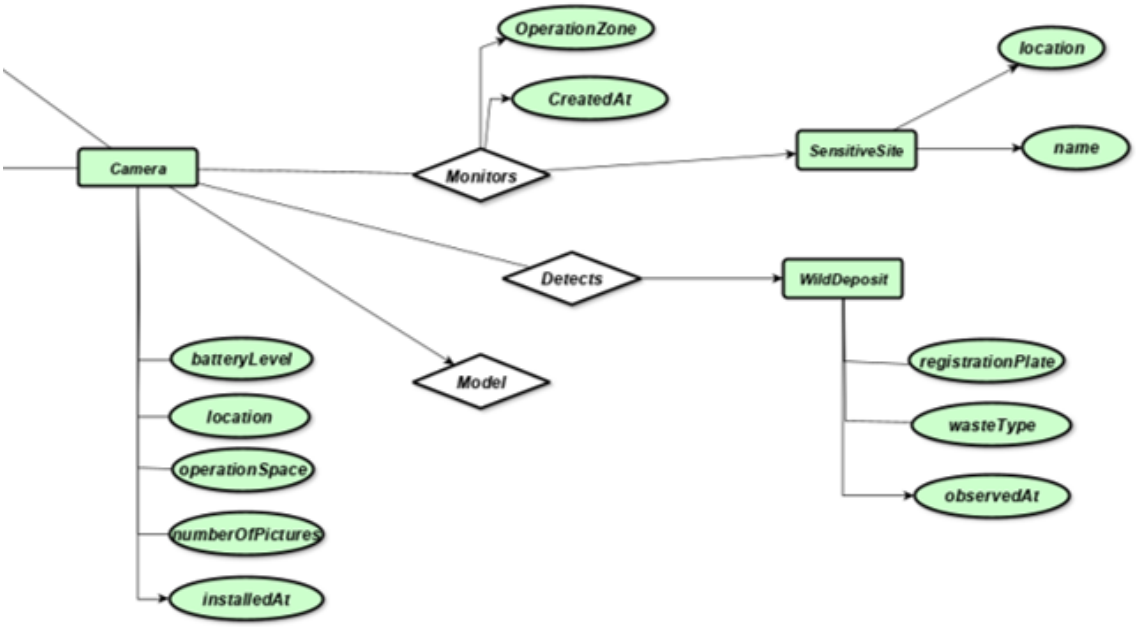

Figure 8: Waste wild deposit video detection model

4.3 Hakusan

Hakusan city is located in the mountain area in Ishikawa prefecture of Japan and at the south of Kanazawa city which is a provincial city in Japan. Detailed location and profiles are shown in Figure 9. We plan the trial in Kanazawa Institute of Technology (KIT) Hakusan mountain campus and around areas.

4.3.1 Infrastructure producing data

The infrastructure of the IoT services producing data for the Root Data domain is shown in fig. 10. End devices consist of two categories, e.g., common service devices and specific service devices. The common service devices include cameras and environmental sensors. Specific service devices include animal traps with sensors and direction detector for wildlife monitoring, for example. In addition to end devices, local servers are deployed in this infrastructure. One is an Integrated recorder to store contents from several kinds of cameras. Another is an Animal detector for specific services. It supports detection of monkeys as captured animals, using contents stored in the integrated recorder for wildlife monitoring.

As shown in fig. 10, networks collecting data consist of three categories, e.g., broadband wireless networks (WiFi), narrowband wireless networks (LoRaWAN) as wireless access networks and a backbone network (MQTT). These wireless networks are connected to a backbone network through Access Nodes. As shown in Figure 11, in mountain areas, LoRaWAN is applied. In residential areas and KIT campus, WiFi and MQTT can be applied. In a backbone network, a Gateway (GW) aggregates transferred data across a backbone network. GW produces data at the reference point of Root Data Domain.

4.3.2 APIs

APIs are specified between the ThingsVisors of Fed4IoT architecture and GW in Figure 10. Briefly, ThingsVisors shall access any locations to get/set data by a unique method. Therefore,

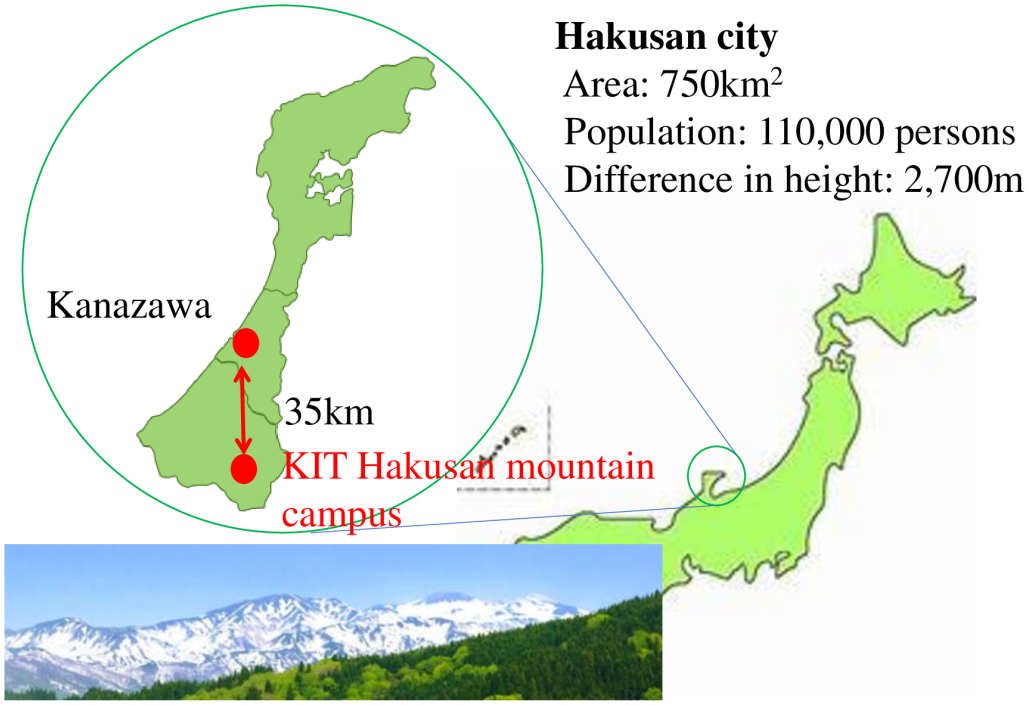

Figure 9: Overview of Hakusan city

Data models and API at the Root Data Domain reference point shall comply with standardized models, e.g. oneM2M, FIWARE, etc. that will be identified in the second release of this deliverable. Data at this point is translated starting from data produced by the infrastructure, to comply with these specifications. Moreover, location, time, and common service data are added to each data of a specific service. Specifications of API are based on the existing standards, e.g. Rest API. Overview of APIs is shown in fig 12.

4.3.3 Data

The produced data set for this infrastructure is detailed in the Annex. They include: information about monitored video frame matching some conditions (e.g. wild animal detection, or person finder), monitored conditions, such as direction of moving targets, and weather information. Since the project is at an early stage, Hakusan deployment is on-going, thus the resulting list of available information can be updated in the second release of this deliverable.

4.4 Kumamoto

4.4.1 Infrastructure producing data

In Kumamoto, a surveillance camera system is installed to detect mainly human presence and extract human face features. The surveillance camera system is composed of the surveillance cam-

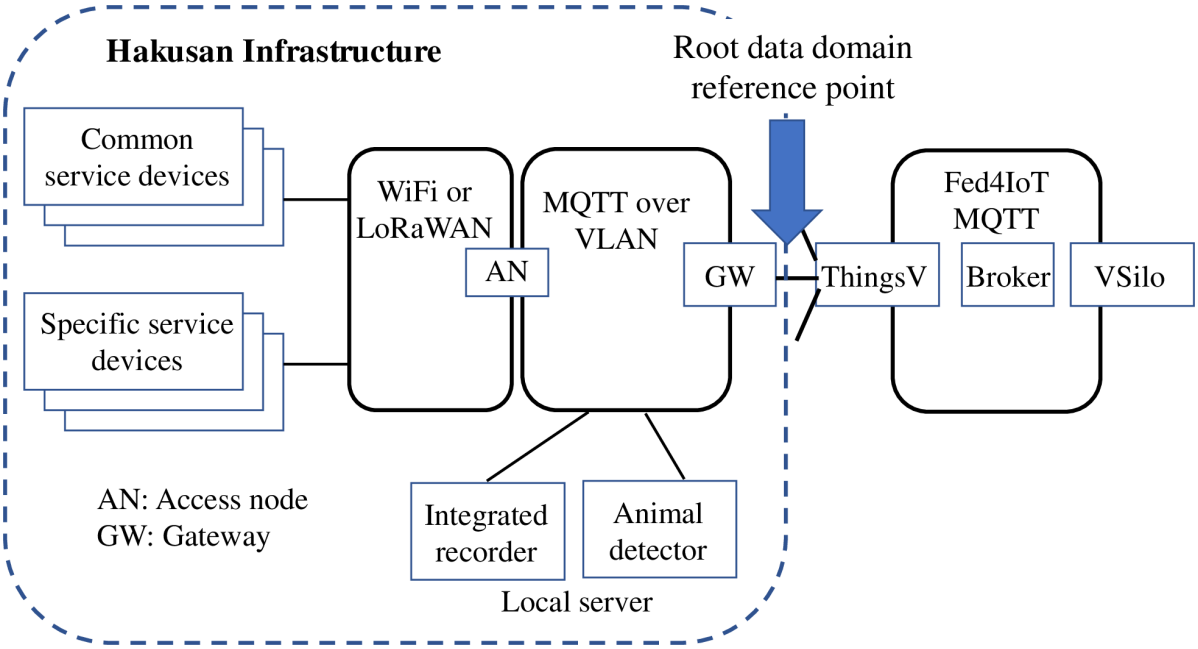

Figure 10: Configuration of infrastructure in Hakusan trial area

era and processing nodes. The surveillance camera has capabilities of capturing images/videos and transmission of the images videos. The processing node has capabilities of executing image processing and extracting the context information hidden in the images/videos. Therefore, infrastructure at Kumamoto produces images/videos plus the information contained within the captured imaged/videos. Details are described as following:

- The surveillance camera can capture raw images/videos, compress images/videos, and stream to the processing nodes which are located on the local surveillance camera, IoT gateway, and the cloud. Image/video compression algorithms are adopted the standardized algorithms, such as PNG and JPEG for image compression and H.264/AVC for video compression. Fundamental parameters of image/video compression can be controlled by system administrators and users. In addition, the surveillance camera equips GPS and produces the geo-location information.
- The processing nodes has capabilities of image processing, such as human detection and face detection. By performing such image processing, the processing nodes extract the context information contained the captured images/videos. By adopting human detection, the processing nodes can notify human existences and the number of humans. Similarly, by adopting face detection, the processing nodes can extract face features from the human images.
- The images, videos and context information are stored in the local device, IoT gateway node, and the cloud, and the (authorized) application providers/users can access them by APIs.

As an option, LiDAR sensors are also installed in Kumamoto to detect moving objects, such as humans and doors. LiDAR sensors can produce the event trigger information. Such trigger information can control the other IoT devices, including surveillance cameras. For example, the trigger information can switch the surveillance camera activation. LiDAR equips GPS and

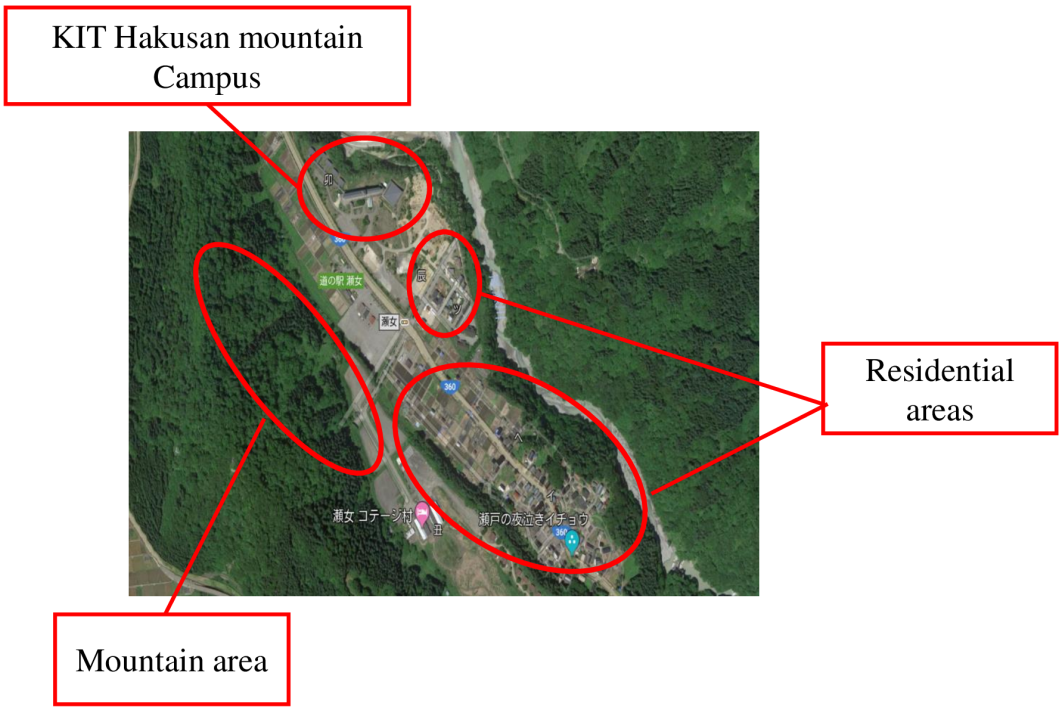

Figure 11: Geometrical overview in Hakusan trial area

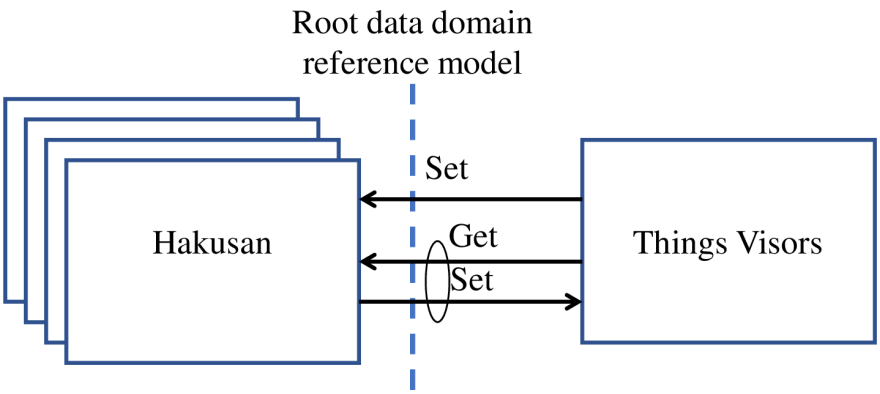

Figure 12: Type of data from/to an Infrastructure

identifies the geo-location of the detected event provided to a ThingVisor of the Root Data Domain.

4.4.2 APIs

As indicated in section 4.4.1, by using APIs, the (authorized) application providers/users/ThingVisors can not only access infrastructure producing data but also control the surveil-

lance camera system. Thus, two types of APIs are defined; APIs for accessing infrastructure producing data and controlling surveillance camera system. The APIs allows reading, writing, subscribing operations. In Kumamoto, the APIs are provided by REST and other protocols, such as topic-based messaging protocol and Information Centric Networking (ICN) protocol.

- APIs for accessing infrastructure producing data can retrieve the data produced by the surveillance camera and processing nodes. In general, the (authorized) application providers/users send a request message (HTTP GET, subscription message) to obtain the data. Thus, at least, the APIs are provided by REST (pull-based protocol), but other protocols may also be available.
- APIs for controlling the surveillance camera system can change the camera parameters, such as compression ratio, transmission interval, and data types (image or video). In general, the APIs are used to change the camera status and maintenance of the surveillance camera system. Similar to the previous APIs, at least, these APIs are provided by REST, but other protocols may also be available.

4.4.3 Data

As indicated in section 4.4.1, the surveillance camera system stores the images/videos and the context information contained images/videos. The context information is two kinds of context sources: person detection information and face feature information. Therefore, the surveillance camera system has the next entity types:

Type	Description
Camera	This entity contains a harmonised description of surveillance camera. See section 9.4.1.
Human Detector	This entity contains a harmonised description of Human detector. See section 9.4.2.
Face Feature Detector	This entity contains a harmonised description of ace Feature Detector. See section 9.4.3.
LiDAR	This entity contains a harmonised description of LiDAR. See section 9.4.4.

Table 6: Surveillance camera system entity types for Kumamoto

In the Annex, we can find the details regarding this information.

5 Services and Silos

This section describes the application(s) and services offered by each pilot, and how the Fed4IoT architectural components (vSilo, ThingVisors, etc.) are exploited to implement these applications. According to the project GANTT, pilot implementation, as well as the Fed4IoT architecture one, is not yet completed. Consequently, the following information represents just a preliminary plan, which however is harmonizing the technical research activities of the project, while also considering regulation aspects.

5.1 Smart Parking

The daily traffic congestion is a big problem that cities must face. The commuting and activities carried out by the citizens are factors that increase this traffic congestion.

One important factor that contributes to the considerable increase in this traffic congestion is a large number of vehicles wandering along with the city. The citizens are finding free parking spots near their destinations and can't park their vehicles.

Smart Parking application devised by Fed4IoT aims to provide a service to reduce the vehicles wandering, the citizens are assisted in their vehicle parking activity, and as a consequence, it will cause the reduction of traffic congestion.

5.1.1 Applications and Services

Smart Parking application will provide a GUI, allowing the user to specify some selection criteria as destination location (presenting a map-based web interface/App), estimated arrival time, estimated parking duration, the maximum distance admittable to the destination location, the maximum price of parking, etc.

When the user establishes all needed selection criteria, Smart Parking application will make reasoning to generate an informed recommendation about the best destination area to park the vehicle.

Smart Parking application uses NGSIv2 API to obtain the information. This NGSIv2 API is provided by a Virtual Silo component, which has an Orion Context Broker that handles the user request.

5.1.2 vSilo and Broker

As indicated above, to handle the Smart Parking application requests, it will be necessary to use a specific vSilo following a NGSIv2 flavour, i.e. exposing data through a NGSIv2 broker.

This component has a silo-controller that, with starting from received neutral format data, produces NGSIv2 data which is stored in an Orion Context Broker embedded in the vSilo by using a POST request to NGSIv2 API op/update method. This methods receives a JSON format body with the identifier, entity type, and attributes value. It allows an action type (APPEND), which enable to introduce new entities or update the value of existing entity attributes.

As a result of this, the vSilo offers to Smart Parking application an NGSIv2 API to access to his data. As indicated in 4.1.1, NGSIv2 offers two options to access to OCB entities data, using the entity queries or the subscription mechanism.

The vSilo component is connected to an MQTT broker to receive data generated by a ThingVisor component. ThingVisor creates the vThings that produces the neutral format data to send information to the vSilo component.

5.1.3 ThingVisors and vThings

As indicated above, to send data to the vSilo component that Smart Parking application uses, it will be necessary to use a specific ThingVisor component that obtains data from a provider in the Root Data Domain and generates the neutral format data for the vSilo.

This ThingVisor receives the provider data (as indicated in 4.1.1), process it and produces NGSI-LD data which is sent to MQTT broker in a specific vThing topic.

As a result of this, Thingvisor sends an NGSI-LD payload to the vSilo component of its vThings. If vSilo component is subscribed to the corresponding vThings, it receives the data.

5.2 Wild Waste deposit Management

Incivilities is a concern for cities as a cause of city degradation and reduced quality of life for the population. Reducing such incivilities requires a double action of the city to not only fine major incivilities but also by educating the citizen about the impact of such actions. In both cases, it is required to detect events as soon as they occur to reduce their impact. In the Fed4IoT context, the focus is made on wild waste deposit where people throw, in general large, items in undue places.

5.2.1 Applications and Services

The application has to provide a dashboard displaying the situation of the spots under surveillance as well as sending an alert when a new event occurred. On the dashboard, the presence of waste on the monitored site will be displayed to inform the cleaning squads whereas the status of the camera (battery level, number of detected events, number of images taken) will be provided for system maintenance purpose. The alerts will be sent by email to the person in charge of details about the infraction.

5.2.2 vSilo and Broker

The application design will build upon a third party IoT platform such as thingsboard.io. Accordingly, we will use a novel vSilo flavour *ready for easy programming* directly exposing thingsboard.io API. The vSilo controller will connect to the MQTT broker of the thingsboard gateway, injecting there the received vThing data, so that key-value MQTT pairs get generated appropriately to feed the dashboard.

5.2.3 ThingVisors and vThings

An initial deployment provide the use of a wild waste deposit ThingVisor creating the Wild deposit vThing publishing wild waste events. Next generation of the pilots will foresee deployment of new scenarios based on additional vThings still fetching data from same devices of Root Data Domain (IoT devise sharing). As examples are the person data finder which will initiate face recognition algorithms on ThingVisor deployed on the edge or a traffic monitoring system building upon camera movement recognition capabilities (see figure 13).

5.3 Cross Border Person Finder

The number of travelers using international flights is rapidly increasing and many elderly persons enjoy traveling abroad after their retirement. They are not able to speak local languages and their children sometimes worry that their parents are well and enjoy their trip.

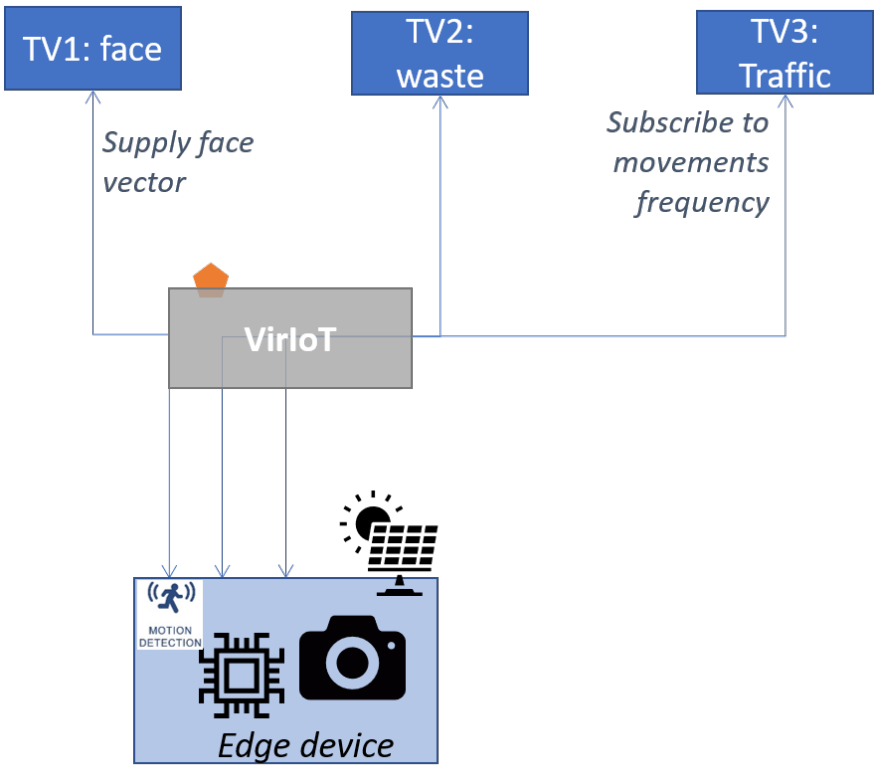

Figure 13: Foreseen ThingVisors on Grasse testbed

There are many cameras in Smart Cities, however, these cameras are aimed to be used for security and safety of the city. It would be useful if these cameras can be used for checking the safety of such travelers from their motherland and share the safety information with police officers, travel agencies in case of some accidents.

The Cross Border Person Finder application is strictly aimed at notifying authorized users (e.g., security authority, parents) about the presence of a given person in a specific place/area. This functionality can be used for instance to find a lost child or an elderly person, expanding the automatic search over different EU or JP regions. Strong implications in terms of consent and processing of biometric data can arise from this applications. We try to analyze them in the following chapter 7

5.3.1 Applications and Services

This application will provide a GUI for

- identifying the person to be found with his photo;
- identifying the place/area to find the target;
- verifying that the use of the application is an appropriate person to find him/her;
- verify the user consent to find the picture image of the target person;
- notify the match of the possible target with encrypted face information;
- show the picture (or streaming recording) of the target.

The application finds the target person by matching the face image extracted from the photo provided by the user of the application and the face feature information gathered from the Smart City infrastructure. The face feature information is encrypted and only authorized application can receive them.

Person detection algorithms will carry out matching operations using photos stored in a Person DB, operated by Identity document issuing authority. The identity document issuing authority assures the integrity of photo data and credentials of the users. A part of photos stored in the Person DB are pre-fetched on edge storage resources **according to user consent**. Pre-fetching will be optimized, moving only the part of the DB that could be used for finding the person. For instance, the image of a lost senior person in a city will be initially pre-fetched in the edge storage of that city and then, after some time, also in the edge storage of other close cities and so forth (expanding ring search). End users can enter images of the person to be searched and be notified after their finding.

The application will be implemented on Microsoft Azure cloud environment, connected to our virtualization infrastructure, thus exploiting information gathered from a dedicated vSilo.

5.3.2 vSilo and Broker

The vSilo component required for the Cross Border Person Finder may be based either on a raw MQTT or a NGSI-LD flavour both handling JSON-LD data produced by the ThingVisors. The raw MQTT vSilo merely exposes vThing data through a dedicated MQTT broker, whereas a NGSI-LD vSilo expose this data through a dedicated NGSI-LD broker (developed within the project, WP4). The broker will be in turn connected to an application server in Microsoft Azure cloud where the application logic run.

5.3.3 ThingVisors and vThings

For supporting Cross Border Person Finder application, we plan to use the two ThingVisor components generating two vThings, respectively, whose data have to be sent to the pilot vSilo.

- Virtual human detector
- Virtual face feature detector

ThingVisors receives and republish the provider NGSI-LD data in the form of events whose attributes are described in the Appendix.

5.4 Wildlife Monitoring

Damage to field crops by wildlife is a serious problem in rural areas. For the wildlife damage prevention, real-time monitoring of the presence of animals is required. Moreover, for utilizing animals as gibier or furs, trap status needs to be informed as soon as possible. Wildlife monitoring application of Fed4IoT provides real-time monitoring information of wildlife.

5.4.1 Applications and Services

Wildlife monitoring application will provide a GUI, allowing users to identify the place of traps, the status of the traps, animals captured time, the presence of animals around residential area, and the like. The users are able to get such information in real time with their terminals. Required information can delivered from vSilo.

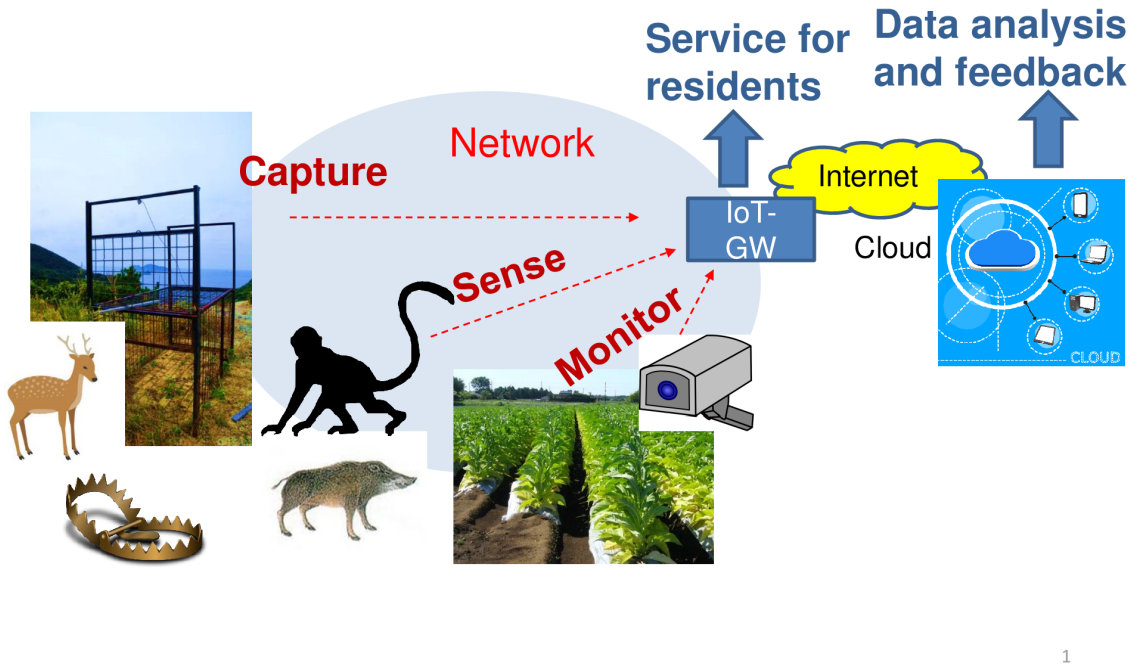

Figure 14: Overview of the Wildlife Monitoring System

5.4.2 vSilo and Broker

An application which utilizes the monitored data generated by the wildlife monitoring system gathers the following data.

- Types of trapped animals
- Number of trapped animals
- Places of trapped animals
- Time of trapped animals
- Where animals are
- Types of animals
- Number of animals
- Time when animals were detected

As shown in Figure 15, vSilo for the wildlife monitoring system provides the data to the application which utilizes the monitored data. Data transfer between the Application and the Broker in vSilo is achieved via the standardized method such as oneM2M, FIWARE or NGSI-LD that

will be specified in next releases of this deliverable. vSilo gathers data from the ThingVisor via internal Pub/Sub system (Fed4IoT Broker). With this IoT virtualization platform, tenants are able to develop various applications without owning the actual devices.

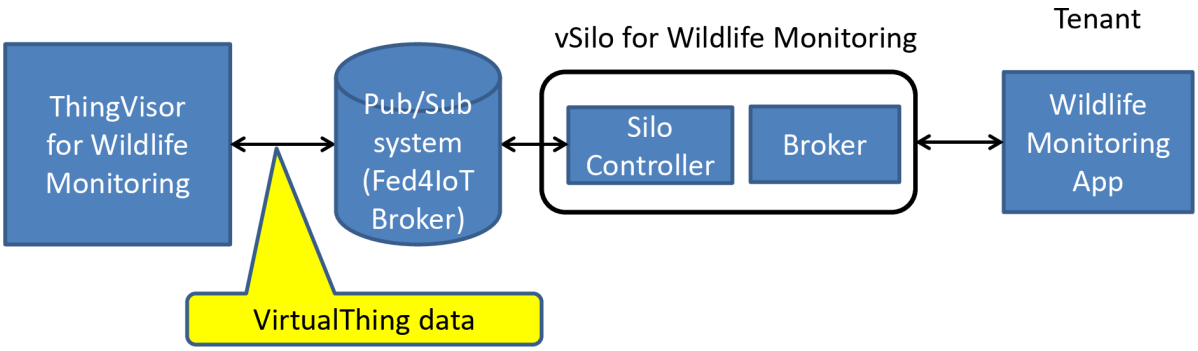

Figure 15: vSilo for Wildlife Monitoring

5.4.3 ThingVisors and vThings

As shown in Figure 16, ThingsVisor for the wildlife monitoring system provides VirtualThings data. Data transfer between real things and ThingVisor for Wildlife Monitoring is achieved via the standardized method, likely oneM2M. ThingVisor possibly processes the **real thing** data then produces the VirtualThing data. For example, "amount of sunlight", "temperature", and the like, possibly being generated by processing the image data of animals captured by cameras. In such example, sensors are virtual things. However, image processing itself is out of scope of this project, then such data are generated by the real sensors in Hakusan city. In this project, ThingsVisor for the wildlife monitoring system copies data delivered from real things, in case adapting them to the internal neutral format (NGSI-LD).

5.5 Citizen Made IoT Applications

The objective of this pilot, *Citizen Made IoT Applications (CMIA),* is to show the user programmability of the Fed4IoT systems where the owners and the users of IoT systems can integrate different local IoT devices and VirtualThings whose data is provided by source available in different Root Data Domain, and let them work together to implement a variety of services in a specific environment. Unlike the other pilots described in Sections 5.1∼5.4, this pilot does not implement IoT services. **This pilot provides a Fed4IoT native programming environment to create IoT services**.

5.5.1 Applications and Services

CMIA will provide an IoT service development environment with GUI, *CMIA Facade*, for the owners and users of IoT systems with no prior experience of computer programming to specify the IoT services they want. The IoT services are described by placing *service components (functions or vThings)* from the list of components prepared by Fed4IoT system provider in advance, linking these service components, and configuring parameters of the components similarly to Node-RED [2].

CMIA will also provide the capability for developers of IoT functions and ThingVisors to register their programs to CMIA and to deploy them in the Fed4IoT virtualizatin platform

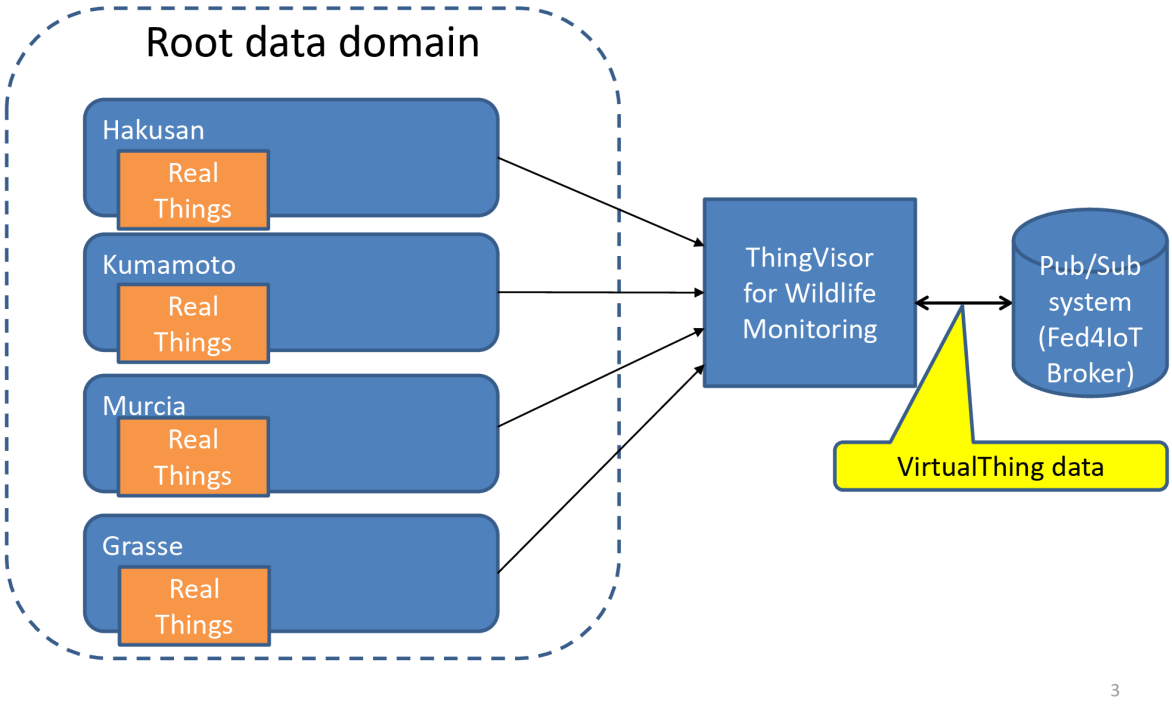

Figure 16: ThingVisors for Wildlife Monitoring system

for sharing them with other tenants. Accordingly, CMIA will support both application and platform programming.

CMIA Code Generator generates programs to be executed at computing resources distributed over the network and to cooperate among them. CMIA deploys the programs to appropriate computing resources, establishes communication channels among them, configures the deployed programs, and controls the execution of the programs.

5.5.2 vSilo and ThingVisors

CMIA is composed of the components shown in Figure 17. A citizen who intends to create an IoT service application compose a service using CMIA Facade. CMIA Code Generator generates programs and a configuration description based on the specification prepared using CMIA Facade. The configuration description is communicated to the vSilo Controller, and based on the description there, the vSilo Controller deploys the created programs at computing resources in VirIoT in cooperation with the Master Controller. *CMIA ThingVisor Register* registers functions and ThingVisors created by developers to VirIoT ThingVisor Store.

Some of the programs generated from the CIMA Facade may be already operating in VirIoT since they are common functions and the programs are shared among many IoT services/vSilo. Others are created for the IoT service just created and dedicated to the service. They are operating in its own vSilo.

With respect to similar frameworks, we observe that Node-RED creates a monolithic pro-

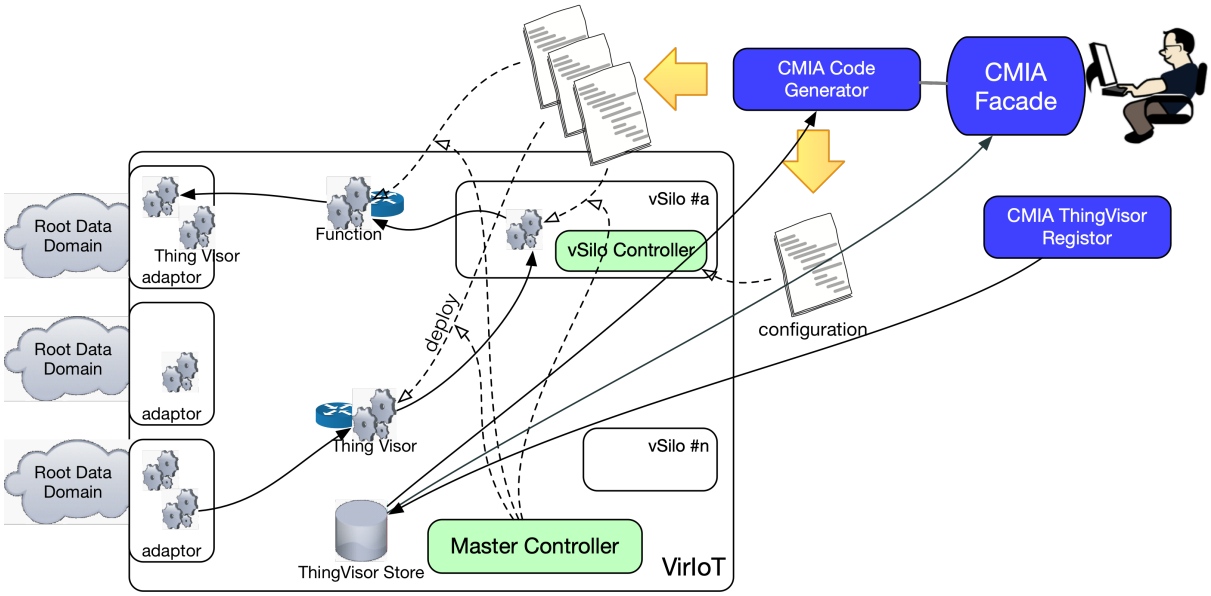

Figure 17: Architecture of Citizen Made IoT Application Pilot

gram to implement an IoT service while CMIA creates a group of programs some of which are already running in the system while others are created for the IoT service and to be initiated. They communicate with each other. Although the created IoT service communicates many sources of information of the Root Data Domain, the IoT service does not need to distinguish them because they are represented by ThingVisors, and by linking and configuring them, Root Data Domain forms a federated environment and used through the ThingVisors transparently.

6 Validation of the pilots

The Fed4IoT platform is a software and hardware system that makes extensive use of different technologies for achieving the project goals and meeting the requirements listed in deliverable D2.1. Differently from "verification", which involves the evaluation of the software during the development phase within the specific work packages, in order to assure that it is compliant with all the requirements stated in the design phase, the validation process comprises testing the software once its development phase is finished, in order to verify that the technical features are valid for the final release of project pilot applications.

We remind that from the *"the project research and innovation focus is on IoT interoperability and virtualization, whose proofs of concept should not require a large number of humans, as instead would be if the focus were on developing image processing algorithms"*, it follows *that the deployment of pilots .... will be likely done in small-scale, private, environments fully controlled by the consortium members"*. Accordingly, for the validation of the pilots we plan to focus on requirement and performance tests only, as hereafter described.

6.1 Requirements' tests

These testing activities aim at verifying the fulfillment of functional and non-functional requirements presented in deliverable D1.2, which concern:

- the technologies developed within the scope of the project,
- technologies and networking of sensors/actuators already in place or to be deployed,
- data communications networks and the design of the parts that are visible to end-users.

For each requirement we will prepare a test report. We report the designed template in table 7.

Req. ID	Req. Description	Test Description	Result	% Passed
ID as in D1.2	*Description of the requirement as reported in D1.2*	*Description of the test executed to verify the requirement*	*Result of the test*	*Percentage of requirement fulfillment*

Table 7: Requirement Test Report

6.2 Performance tests

This section presents the identified performance parameters that, at this stage, we have considered for each of the pilot. They will serve the purpose of validating our platform.

6.2.1 Smart Parking

The objective of the Smart Parking pilot is to reduce the vehicles wandering and, as a consequence, to obtain reduction of traffic congestion. To assist the citizens in their vehicle parking activity, the following performance indicators will be considered.

- Timeliness of the information (can be applied to Private Parking Sites)

- Average Time by User. Sum of the parking periods per user / total number of users (can be applied to both data).
- Recommendation failure rate. Ratio between Total number of recommendation failures and Total number of recommendations in a time period. (can be applied to both data).
- Issued ticket frequency by Sector. Total issued tickets in a Time units [days/weeks/-months] per Sector (can be applied to RPZ information).

6.2.2 Wild Waste deposit management

The objective of the wild waste deposit scenario is to be able to automatically detect misbehavior of people throwing waste in public places (wild deposit) and report the detected events to the city managers. The following performance will be considered for validation purposes.

- System precision. Ratio $\mathbf{TP}/(\mathbf{TP}+\mathbf{FP})$ with TP being the number of true positive and FP the number of false positive over a given period of time. Target $> 90\%$
- System latency. Maximum timing between event detection and warning on dashboard. Target $< 30min$.
- Autonomy. Duration over which the on-site system can run autonomously (storage space, energy). Target > 3 weeks.
- Event frequency. Number of event detected per week and per monitored sites.

6.2.3 Cross-border Person Finder

A main goal of cross-border person finder (CBPF) application is to find out the geo-location of requested persons. Thus, we plan to evaluate CBPF application by means of:

- Detection Accuracy of requested persons. The detection accuracy is defined as a proportion of number of true positive and true negative. F-measure, proportion of precision and recall, is one of good candidates of evaluation metrics, and such metric is frequently adopted in the Computer Vision research field.
- Network parameters. In addition, CBPF application requires the surveillance cameras, and such cameras produce (and streams) huge amount of data in Internet, and achieving a network-friendly CBPF application (less traffic, lower network delay) is mandatory. Thus, network traffic volumes and end-to-end network latency are also evaluation metrics.

6.2.4 Wildlife Monitoring

Wildlife monitoring is focused onto monkey capturing, other animal capturing, and alert of animal approaching. Therefore, the following performance parameters will be monitored to show application effectiveness.

1. Monkey recognition ratio
2. Accuracy of animal capturing
3. Accuracy of detection of animal moving direction
4. Response time

6.2.5 Citizen Made IoT App

Since CMIA is targeting people without prior experience of programming, usability of the system is most important when the system is complete. However, at the first stage of development, the goal is to be able to generate working IoT services that federate a plurality of IoT data sources to form the Root Data Domain.

7 Legislation and Privacy Concerns

We attempt in this section to assess the impact of GDPR (Regulation EU 2016/679) [3] on our pilots. Specifically, we must explore the following three aspects which, more than others, are crucial in our cases:

- Decisions based on **automatic processing**; this is important for the case of automatic recognition of people in the cross-border person finder, and of plates in the waste management, because in these use cases we operate under the assumption that acquiring consent from the data subjects is impossible.
- Involvement of **cloud providers** and other sub-processors; this is transversal to all pilots and central to the project, since the adoption of a virtualization platform implies more complexity in the chain of data processors.
- **Cross-border** EU-JP transfer of data, investigating also non-GDPR scenarios.

Throughout the section we will make use of the fundamental definitions set forth in Article 4 of the GDPR, concerning personal data and controllers of such data. Definitions are as follows:

Term	Definition
Data subject	An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.
Personal data	Any information relating to an identified or identifiable natural person (data subject).
Processing	Any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction.
Controller	The natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data; where the purposes and means of such processing are determined by Union or Member State law, the controller or the specific criteria for its nomination may be provided for by Union or Member State law.

Table 8: Fundamental definitions from GDPR's Article 4

7.1 Decisions based on automatic processing

Making decisions based on automatic processing of incoming data is central to all IoT pilots, but in cases where impact of such automated decisions has legal or significant implications for the data subject, or is based on biometric data, and data controllers and processor are unable to easily acquire consent from the data subjects, are particularly critical. Acquiring consent is difficult in:

- the waste management, whereas data subjects are misbehaving citizens;
- the cross-border person finder, whereas data subjects may not be the ones who entered into a contract with the operators of the system.

Automatic processing in the context of GDPR is the subject of a guidelines document (adopted on 3 October 2017, last revised on 6 February 2018) issued by the Working Party on the protection of individuals with regard to the processing of personal data, set up under Article 29 of Directive 95/46/EC (WP29) [4]. The guidelines aim at providing some clarity around the concepts of profiling and automated decision-making of GDPR. Currently, the European Data Protection Board (EDPB), which includes representatives from the data protection authorities of each EU member state and adopts guidelines for complying with the requirements of the GDPR, has replaced the Article 29 Working Party. WP29's adopted guidelines on automated individual decision-making and profiling have been endorsed by the EDPB.

7.1.1 Profiling vs. decision-making

There is a distinction between profiling and automated decision-making. In general terms, profiling is the act of collecting information about an individual or group, in order to infer characteristics or behavioural patterns, so that the system can assign her to a certain (pre-defined, usually) category. More precisely the GDPR defines profiling in Article 4(4) as:

Term	Definition
Profiling	Any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person, in particular to analyse or predict aspects concerning that natural persons performance at work, economic situation, health, personal preferences, interests, reliability, behaviour, location or movements.

Table 9: Definition of profiling in Article 4(4)

We observe that profiling is composed of some key elements:

- automated processing...;
- ...to be carried out on personal data;
- ...to evaluate personal aspects about a natural person;
- profiling is not about decision-making.

The word "decision" does not appear in the above definition, meaning that a data processor may subdivide personal data processing activities into (a) profiling and (b) using an individuals profile in order to make an automated decision. Profiling is not in itself an automated decision. For instance, the bank may ponder someones credit profile to decide whether or not to concede a loan: the decision is whether or not to activate the loan; the individuals profile is just used to support the decision.

GDPR clearly supports that **decision-making based on personal** data of a data subject, legally or significantly impacting the individual, **is more risky than profiling** alone.

7.1.2 Human involvement in decisions about data subjects

This is a key distinction that is further clarified in Article 22, which specifically refers to "solely" automated processing. A decision based solely on automated processing is a decision with no human involvement in the decision process.

To better understand this difference, we formulate another example (adapted from the guidelines, fit to our waste management scenario): imposing fines to misbehaving users of the waste bins purely on the basis of evidence from cameras capturing the user's car plate, is an automated decision-making process that does not necessarily involve profiling. However, if we monitor the individual over time, recording her habits, and then this information (such as whether misbehaving is a repeated offence, or whether she has had other recent violations), is used to tune the amount of the fine, then this would certainly become a decision based on profiling.

The key point is that Article 22(1) of the GDPR specifically addresses and limits the situations in which you can make:

- **solely automated decisions** that have
- **legal or similarly significant** effect on individuals.

> "The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly affects him or her"
> Article 22(1)

Thus, if the process excludes any human influence on the outcome, we conclude that it is solely and totally automated a decision-making process. The easy interpretation of the guidelines is that a process wont be considered solely automated if a human is part of the process and interprets results of an automated decision-making, before applying it to the individual. For instance [5] say that the human involvement has to be active and not just a token gesture, and that the question is whether a human reviews the decision before it is applied and has discretion to alter it, or whether they are simply applying the decision taken by the automated system.

The guidelines warn that fake involvement of a human in the process, in order to circumvent the rules on solely automated decision-making, would not work, as the human involvement must be meaningful and not just a repeated gesture. The individual needs to have the authority to change the decision considering all the information available. Specifically (page 21) we read that "the controller cannot avoid the Article 22 provisions by fabricating human involvement. For example, if someone routinely applies automatically generated profiles to individuals without any actual influence on the result, this would still be a decision based solely on automated processing. To qualify as human involvement, the controller must ensure that any oversight of the decision is meaningful, rather than just a token gesture. It should be carried out by someone who has the authority and competence to change the decision. As part of the analysis, they should consider all the relevant data".

From the above, we easily infer that Article 22's "solely automated" scenarios are more challenging and restrictive than human-supervised decision-making, but **only in the case where the decision produces legal or otherwise similar** effects on the subject.

7.1.3 Lack of consent

Further impacting some of our pilots is the issue of consent, or lack thereof, due to the very nature of the service we would like to test, which prevents (both in the case of cross-border person finding and when detecting misbehaviour at waste bins via car plate detection) the data processor to acquire consent in real-time during the system's operation.

Is consent always required in order to make processing of personal data lawful? GDPR states conditions for lawful processing of personal data in Article 6. Consent is just one of several.

> "Processing shall be lawful only if and to the extent that at least one of the following applies:
>
> (a) the data subject has given **consent** to the processing of his or her personal data for one or more specific purposes;
>
> (b) processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract;
>
> (c) processing is necessary for compliance with a legal obligation to which the controller is subject;
>
> (d) processing is necessary in order to protect the vital interests of the data subject or of another natural person;
>
> (e) **processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller**;
>
> (f) processing is necessary for the purposes of the **legitimate interests pursued by the controller** or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child."
>
> Article 6(1) (boldface added)

Article 6, paragraph 1, above clearly states that **public interest and exercise of public authority allow to get away with consent**. These circumstances are better specified in the subsequent paragraph 3.

> "The basis for the processing referred to in point (c) and (e) of paragraph 1 shall be laid down by:
>
> (a) Union law; or
>
> (b) Member State law to which the controller is subject."
>
> Article 6(3)

Thus, if public interest or exercise of public authority or compliance to legal obligations are involved, there must be a clear reference to an existing law supporting the purpose for processing personal data without the data subject's consent.

Let us now briefly examine whether legitimate interest, introduced in Article 6(1f), can be used to obviate the need for consent in our use cases.

Broad interpretations of this section have been openly discouraged: "open-ended exceptions along the lines of Article 6 GDPR, and in particular Article 6(1f) (legitimate interest ground), should be avoided." (See [6]). When trying to answer the question about what legitimate interest is, the GDPR itself provides some examples such as processing personal data for internal administrative purposes, including the processing of clients' or employees' personal data, to guarantee network security, to prevent unauthorised access to electronic communications networks and malicious code distribution, and to report possible criminal acts or threats to a competent authority, to prevent fraud. This latter interpretation could mildly be forced onto the waste management scenario, but it is still quite far away. What constitutes legitimate interest will become clearer, over time, when more decisions by the relevant bodies will be available. In any case, to genuinely assess that legitimate interest exists, organizations should document the necessity of the specific processing, and they have to balance the interest of the processing with the rights of data subjects, who can object to legitimate interest being used as a basis for processing. A careful assessment of legitimate interest must in fact include (Recital 47) **whether a data subject can reasonably expect collection** and processing of data for that purpose.

7.1.4 Biometric data

Recital 51 states that "personal data which are, by their nature, particularly sensitive in relation to fundamental rights and freedoms merit specific protection as the context of their processing could create significant risks to the fundamental rights and freedoms".

The types of data that fall into this category is explicitly listed in Article 9, dedicated to processing of special categories of personal data.

> "Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, **biometric data for the purpose of uniquely identifying a natural person**, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited."
> Article 9(1) (boldface added)

Thus, photographs are to be considered biometric data only when processed through a specific technical means allowing the unique identification or authentication of a natural person, as explicitly stated in Recital 51.

Article 9(2) continues by listing a number of possible situations where the prohibition stated in paragraph 1 would not apply. The first and foremost exception is **explicit consent given by the data subject** (except where a law provides that the prohibition may not be lifted altogether).

Other notable exceptions are related to protecting the vital interests of the data subject, when physically or legally incapable of giving consent.

Further exceptions are possible for reasons of substantial public interest, for the purposes of preventive or occupational medicine, and for reasons of public interest in the area of public health.

7.1.5 Discussion and Privacy Impact Assessments

The GDPR recognises that automated decision-making, including profiling can have serious consequences for individuals. We think that both the cross-border person finder service and the automated fines imposed to citizens who misbehave in their waste disposal habits, imply decision-making based solely on automation. It would appear impossible to scale up these services if active human intervention is employed in the identification process of each and every case under scrutiny. Person finding effectiveness is best measured by the real-time detection capability of the system. Automatic fines would not be cost-effective if human intervention is required.

Does the automated decision produce legal or similar effects on the subject? This point is crucial, because the Article22(1) prohibition only applies in specific circumstances when a decision based solely on automated processing, including profiling, has a legal effect on or similarly significantly affects someone. The GDPR does not define "legal" or "similarly significant" however the wording makes it clear that **only serious impactful effects are prohibited** by Article 22.

As stated in the guidelines, a legal effect requires that the decision, which is based solely on automated processing, affects someones legal rights, such as the freedom to associate with others, vote in an election, take legal action, or affects a persons legal status or their rights under a contract. This is not the case in our pilots. But, as said, automated decision-making could still fall within the scope of Article 22 if it produces an effect that is equivalent or similarly significant in its impact: the threshold triggering significance must be similar to that of a decision producing a legal effect. Recital 71 provides the following typical examples: "automatic refusal of an online credit application" or "e-recruiting practices without any human intervention".

With the above clarifications in mind, it is difficult to think that imposing automatic fines impacts legally (or in a similarly significant way) the data subject. Also the person finder use case does not fall, in our opinion, under prohibitions of Article 22. But the overall privacy impact in both cases is quite complex, and we will attempt a draft PIA in the following two tables.

Cross-border person finder	
Architecture	Existing infrastructures of **security surveillance cameras are used to build a person-finder service** on top of them. It is the responsibility of our service, as the user of surveillance equipment, of surveillance solutions and of surveillance services, to ensure GDPR compliance and the safeguarding of the rights of the individuals whose personal data we process.

Consent	Explicit consent from the persons we want to find **has to be collected in advance**. In case of attempting to find children, age of consent has to be taken into account, and we need to verify that anyone providing their own consent is old enough to do so. Article 8 of the GDPR allows member states to set the **age of consent between 13 and 16**. For instance, in Italy and Spain, only children aged 14 or over are able to give their own consent. For children **under this age, consent needs to be provided by the holder of parental responsibility** over the child and we are also required to make reasonable efforts (using available technology) in these circumstances to verify that consent provided on behalf of a child under the age of consent has, in fact, been provided by the holder of parental responsibility for that child.
Automated decision has legal or significant effect	No.
Biometric data	Yes.
Lawful processing based on	Consent. Please notice it may be difficult to acquire consent in this use case.
Summary	Because of the usage of biometric data for uniquely identifying the data subjects, the only way to make this service lawful and compliant with GDPR is to acquire consent of possible target persons in advance. In rare cases (such as lost children or search of terrorists or public enemies), given that a law supports it, social protection or public interest purposes may take over and authorize usage of the service without prior consent.

Table 10: Privacy Impact Assessment for the cross-border person finder pilot

Waste management	
Architecture	Cameras installed at public waste bins / waste collection facilities perform automatic detection of car plates, in order to send pecuniary fines to misbehaving citizens.
Consent	Not possible to acquire consent.
Automated decision has legal or significant effect	No.
Biometric data	No.
Lawful processing based on	Paragraph 1(e) of Article 6, that is "processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller".
Summary	Since we assess that the automated decision-making has no legal impact on the data subject, and insofar as biometric data is not involved, then consent is not required for the processing of car plates and associated personal data, as long as the controller operates in the public interest, as an official authority.

Table 11: Privacy Impact Assessment for the waste management pilot

7.2 Cloud providers and sub-processors

Several papers highlight the changes that GDPR will bring to companies. Usually, the focus is on companies in their data controller vest. There has been much less analysis of the impact of GDPR on IT service providers (data processors) who process personal data on behalf of the controllers, which are regarded as customers from the data processor perspective.

The GDPR (Article 2, Material scope) applies "to the processing of personal data wholly or partly by automated means", hence, in terms of scope, it does not distinguish between controllers and processors.

Recital 81 of GDPR says: "To ensure compliance with the requirements of this Regulation in respect of the processing to be carried out by the processor on behalf of the controller, when entrusting a processor with processing activities, the controller should use only processors providing sufficient guarantees ... **The carrying-out of processing by a processor should be governed by a contract** ... binding the processor to the controller, setting out the subject- matter and duration of the processing, the nature and purposes of the processing, the type of personal data and categories of data subjects, taking into account the specific tasks and responsibilities of the processor in the context of the processing to be carried out and the risk to the rights and freedoms of the data subject ... After the completion of the processing on behalf of the controller, the processor should, at the choice of the controller ..."

Controller's contractual requirements with processor are discussed in detail in Article 24 Responsibilities of Controller, Article 28 Processor, and Article 29 Processing under authority of controller or processor. To sum-up the relevant information from the above articles, we can group controller obligations and processor obligations (see [7]).

Controller obligations:

- Describe subject matter, duration, and nature of processing activity.
- Describe the nature and purpose of processing.
- Describe the types of personal data being processed.
- Describe categories of data subjects being processed.
- Only employ processors which can appropriately protect data subject data.
- Only employ processors who can meet GDPR regulations.

Processor obligations:

- Only process the data on documented instructions from the controller
- Ensure all individuals authorized to process the data have committed to confidentiality agreements
- Assist controller in handling data subject access rights requests
- Assist controller with obligations around security and requests from supervisory authorities.

- Be available and able to assist controller with compliance obligations
- Delete or return all data upon controller request or requirement
- Outline any data transfers outside EEA and describe safeguards which will protect the data
- Contribute to audits conducted by the controller or other required authority
- Ensure any engagement of sub-processors meet same obligations required by the controller.
- Only engage sub-processors upon approval of controller.

From the above, we learn that data processors cannot use another data processor (which we can call a sub-processor) **without the specific authorization from the data controller**. Furthermore, in the case of authorization, the data processor still has an information obligation, and the data controller still has the right to object. This requirement constitutes a significant strengthening of the former rules, which simply applied the provisions of the common contract law in the field of subcontracting. **GDPR creates a specific subcontracting regime in relation to the processing of personal data**, which most likely results in differentiation of contracts related to subcontracting, and the creation of one type of contracts for subcontracting related to data processing and one for other types of subcontracting.

This requirement will be particularly difficult to meet in many-to-many standardized service models (such as cloud computing services where there may be many customers and subcontractors).

It is also conceivable that, for these cloud service models, the right of the controller to object will result in a termination of the agreement. But, in practice, if the processor cannot smoothly pass from one IT service/cloud provider to another, associating the objection with a resolution implies that **the controller's right to the object will be ineffective**. Furthermore, The data processor must also pass through its obligation to any sub processor and the data processor remains fully liable towards the data controller for the performance of that sub processor [8].

7.3 Cross-border EU-JP transfer of data

Article 45(2) of the GDPR, establishes the elements that the European Commission shall take into account when assessing the adequacy of the level of protection in a third country or international organization. For example, the Commission shall take into consideration the rule of law, respect for human rights and fundamental freedoms, relevant legislation, the existence and effective functioning of one or more independent supervisory authorities and the international commitments the third country or international organization has entered into.

An adequacy decision is a decision taken by the European Commission establishing that a third country provides a comparable level of protection of personal data to that in the European Union, through its domestic law or its international commitments. As a result, personal data can flow safely from the European Economic Area (EEA) (the 28 EU Member States as well Norway, Liechtenstein and Iceland) to that third country, without being subject to any further safeguards or authorizations.

In Tokyo, 17 July 2018 (see press release[9]), the EU and Japan successfully concluded their talks on reciprocal adequacy. They agreed to recognize each other's data protection systems as "equivalent", which will allow data to flow safely between the EU and Japan. To live up to European standards, Japan has committed to implementing the following **additional**

safeguards to protect EU citizens' personal data, before the Commission formally adopts its adequacy decision:

- A set of rules providing individuals in the EU whose personal data are transferred to Japan, with additional safeguards that will bridge several differences between the two data protection systems. These additional safeguards will strengthen, for example, the protection of sensitive data, the conditions under which EU data can be further transferred from Japan to another third country, the exercise of individual rights to access and rectification. These rules will be binding on Japanese companies importing data from the EU and enforceable by the Japanese independent data protection authority (PPC) and courts.

- A complaint-handling mechanism to investigate and resolve complaints from Europeans regarding access to their data by Japanese public authorities. This new mechanism will be administered and supervised by the Japanese independent data protection authority.

Fortunately, the very recent piece of news, reported above, of substantial progress in creating a political bridge between EU an Japan concerning exchange of personal data, **allows our project to proceed smoothly** with the issue of deploying our computational infrastructure in servers distributed across EU and JP.

We do hope that the agreements are going to be fully operative withing the time span of the project, so that we can **equate the obligations** (for instance of the processors towards the cloud providers sub-processors) of our EU-JP distributed deploy **to those of an intra-EU distributed, many-to-many one**.

8 Conclusion and Enhancement

This deliverable reflects the work carried out during Task 5.1: "Pilot design and validation methodology". We have presented the information available from the different scenarios specifying their format, and APIs.

Additionally, from the pilots' point of view, we address their design by following a top-down approach: Firstly we defined the Application and services these pilots aim at, by defining their requirements in terms of needed information. Secondly, we have also defined, for each one of them, what the envisioned API is, so that we can match it to specific Silos. And finally, we have examined the processing that ThingVisors have to make in order to generate the virtual things needed by the Applications.

Another building block of this deliverable is the discussion and planning of validation of our architecture, which we paved thanks to the identification of a series of KPIs per pilot.

Our last contribution in this deliverable is a privacy impact assessment and detailed discussion about the legislation and privacy issues risen by two challenging pilots, for what concerns their compliance to GDPR EU regulation.

9 Annex - Root Data Domain

9.1 Murcia

The data model for information currently available in Murcia is NGSIv2.

9.1.1 Parking Site

Parking site attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity. In this case parkingsite.	Mandatory
timestamp	DateTime	Indicates the date/ time when the entity was last observed in ISO 8601 format. The value of this will be set by the server when the entity was observed, if the entity has not been observed it may have a null value.	Optional
name	Text	Indicates the name of parking site.	Recommended
disSpaceMc	Number	Indicates the number of free parking site spaces for disabled persons for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended
disSpaceMcCapacity	Number	Indicates the number of spaces for disabled persons for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended
disSpacePC	Number	Indicates the number of free parking site spaces for disabled persons for private cars. Its possible values: integer equal to or greater than zero.	Recommended
disSpacePCCapacity	Number	Indicates the number of spaces for disabled persons for private cars. Its possible values: integer equal to or greater than zero.	Recommended
EVSpaceMc	Number	Indicates the number of EV free parking site spaces for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended
EVSpaceMcCapacity	Number	Indicates the number of EV spaces for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended

Attribute Name	Attribute Type	Description	Constraint
EVSpacePC	Number	Indicates the number of EV free parking site spaces for private cars. Its possible values: integer equal to or greater than zero.	Recommended
EVSpacePCCapacity	Number	Indicates the number of EV spaces for private cars. Its possible values: integer equal to or greater than zero.	Recommended
numSpaceMc	Number	Indicates the number of free parking site spaces for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended
totSpaceMcCapacity	Number	Indicates the number of spaces for motorcycles. Its possible values: integer equal to or greater than zero.	Recommended
numSpacePC	Number	Indicates the number of free parking site spaces for private cars. Its possible values: integer equal to or greater than zero.	Recommended
totSpacePCCapacity	Number	Indicates the number of spaces for private cars. Its possible values: integer equal to or greater than zero.	Recommended
maxHeight	Number	Maximum admittable vehicle height (in centimeters) that is restricted based on its physical characteristics. Its possible values: integer greater than zero.	Recommended
maxLength	Number	Maximum admittable vehicle length (in centimeters) that is restricted based on its physical characteristics. Its possible values: integer greater than zero.	Recommended
maxWidth	Number	Maximum admittable vehicle width (in centimeters) that is restricted based on its physical characteristics. Its possible values: integer greater than zero.	Recommended
payMthd	StructuredValue	Indicates the accepted payment methods. It is an array with these possible values: "Cash", "PayPal"...	Recommended
payMthdCreditCard	StructuredValue	Indicates the accepted payment methods (Credit Card). It is an array with these possible values: "AmericanExpress", "DinersClub", "Discover", "JCB", "MasterCard", "VISA"...	Recommended

Attribute Name	Attribute Type	Description	Constraint
policyMc	RelationShip	Indicates the parking site policy for motorcycles. This attribute value must contain an identifier of an existing policy entity.	Recommended
policyPC	RelationShip	Indicates the parking site policy for private cars. This attribute value must contain an identifier of an existing policy entity.	Recommended
policyMcPHolidays	RelationShip	Indicates the parking site policy for motorcycles (public holidays). This attribute value must contain an identifier of an existing policy entity.	Optional
policyPCPHolidays	RelationShip	Indicates the parking site policy for private cars (public holidays). This attribute value must contain an identifier of an existing policy entity.	Optional
isOpen	boolean	Indicates when parking site is open. Possible values true or false.	Recommended
monday	StructuredValue	Indicates the open hours on Monday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
tuesday	StructuredValue	Indicates the open hours on Tuesday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
wednesday	StructuredValue	Indicates the open hours on Wednesday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
thursday	StructuredValue	Indicates the open hours on Thursday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended

Attribute Name	Attribute Type	Description	Constraint
friday	StructuredValue	Indicates the open hours on Friday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
saturday	StructuredValue	Indicates the open hours on Saturday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
sunday	StructuredValue	Indicates the open hours on Sunday. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
pHolidays	StructuredValue	Indicates the open hours on public holidays. Its value is an array of JSON items which has two attributes where specific the open an close hours. For example: [{"opens": "8:00", "closes": "20:00" }]	Recommended
carWash	boolean	Indicates if parking site has carwash service. Possible values true or false.	Optional
valet	boolean	Indicates if parking site has valet service. Possible values true or false.	Optional
EVCharger	boolean	Indicates if parking site has EV chargers. Possible values true or false.	Optional
forzado	Number	Indicates several free parking site spots publicly available. Its possible values: integer equal to or greater than zero.	Optional
phoneNumber	Text	Indicates parking site contact numbers.	Optional
webSite	Text	Indicates parking site web site.	Optional
mail	Text	Indicates parking site mail.	Optional
address	Text	Indicates parking site address.	Recommended
location	geo:json	The location point of the parking site. See GeoJSON Specification (RFC 7946).	Recommended

Table 12: Parking site attributes

NGSIv2 Context Definition The following NGSIv2 context definition applies to the parking site entity.

Listing 2: smartparking:parkingsite:context

```
"@context": {
    "name": "http://purl.org/goodrelations/v1#name",

    //********** Places & Occupancy **********
    "disSpaceMc": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumDisSpaceMotorcycle",
    "disSpaceMcCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "disSpacePC": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumDisSpacePrivateCar",
    "disSpacePCCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "EVSpaceMc": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumEVSpaceMotorcycle",
    "EVSpaceMcCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "EVSpacePC": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumEVSpacePrivateCar",
    "EVSpacePCCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "numSpaceMc": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumSpaceMotorcycle",
    "totSpaceMcCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "numSpacePC": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumSpacePrivateCar",
    "totSpacePCCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",

    //********** Maximum dimensions **********
    "maxHeight": "http://ontology.eil.utoronto.ca/icity/Parking/
       maxAdmittableHeight",
    "maxLength": "http://ontology.eil.utoronto.ca/icity/Parking/
       maxAdmittableLength",
    "maxWidth": "http://ontology.eil.utoronto.ca/icity/Parking/
       maxAdmittableWidth",

    //********** Accepted Payment Methods **********
    "payMthd": "http://purl.org/goodrelations/v1#
       acceptedPaymentMethods",
    "payMthdCreditCard": "http://purl.org/goodrelations/v1#
       acceptedPaymentMethods",

    //********** Policy & Rate **********
    "policyMc": "http://ontology.eil.utoronto.ca/icity/Parking/
       ParkingPolicy",
    "policyPC": "http://ontology.eil.utoronto.ca/icity/Parking/
```

```
      ParkingPolicy",
30    "policyMcPHolidays": "http://ontology.eil.utoronto.ca/icity/
      Parking/ParkingPolicy",
31    "policyPCPHolidays": "http://ontology.eil.utoronto.ca/icity/
      Parking/ParkingPolicy",
32
33    //********** Open hours **********
34    "isOpen": "http://ontology.eil.utoronto.ca/icity/Parking/
      isOpen",
35    "monday": "http://purl.org/goodrelations/v1#Monday",
36    "tuesday": "http://purl.org/goodrelations/v1#Tuesday",
37    "wednesday": "http://purl.org/goodrelations/v1#Wednesday",
38    "thursday": "http://purl.org/goodrelations/v1#Thursday",
39    "friday": "http://purl.org/goodrelations/v1#Friday",
40    "saturday": "http://purl.org/goodrelations/v1#Saturday",
41    "sunday": "http://purl.org/goodrelations/v1#Sunday",
42    "opens": "http://purl.org/goodrelations/v1#opens",
43    "closes": "http://purl.org/goodrelations/v1#closes",
44    "pHolidays": "http://purl.org/goodrelations/v1#PublicHolidays
      ",
45
46    //********** Services **********
47    "carWash": "http://ontology.eil.utoronto.ca/icity/Parking/
      CarWash",
48    "valet": "http://ontology.eil.utoronto.ca/icity/Parking/Valet
      ",
49    "EVCharger": "http://ontology.eil.utoronto.ca/icity/Parking/
      EVCharger",
50    "mediumEVCharger": "http://ontology.eil.utoronto.ca/icity/
      Parking/MediumEVCharger",
51    "quickEVCharger": "http://ontology.eil.utoronto.ca/icity/
      Parking/QuickEVCharger",
52    "standardEVCharger": "http://ontology.eil.utoronto.ca/icity/
      Parking/StandardEVCharger",
53
54    "forzado": "https://odins.org/smartParkingOntology/
      parkingProbability",
55
56    //********** Contacts **********
57    "phoneNumber": "http://ontology.eil.utoronto.ca/icontact.owl#
      PhoneNumber",
58    "phoneType": "http://ontology.eil.utoronto.ca/icontact.owl#
      hasPhoneType",
59    "areaCode": "http://ontology.eil.utoronto.ca/icontact.owl#
      hasAreaCode",
60    "countryCode":  "http://ontology.eil.utoronto.ca/icontact.owl
      #hasCountryCode",
61    "contactNumber": "http://ontology.eil.utoronto.ca/icontact.
```

```
        owl#hasPhoneNumber",
    "webSite": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasWebSite",
    "mail": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasEmail",

    //********** Location **********
    "address": "http://ontology.eil.utoronto.ca/icontact.owl#
        Address",
    "country": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasCountry",
    "state": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasState",
    "city": "http://ontology.eil.utoronto.ca/icontact.owl#hasCity
        ",
    "citySection": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasCitySection",
    "streetType": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasStreetType",
    "streetDirection": "http://ontology.eil.utoronto.ca/icontact.
        owl#hasStreetDirection",
    "streetNumber": "http://ontology.eil.utoronto.ca/icontact.owl
        #hasStreetNumber",
    "postalCode": "http://ontology.eil.utoronto.ca/icontact.owl#
        hasPostalCode",
    "location": "https://schema.org/location"
}
```

Example of parking site entity The following is an example instance of the parking site entity (NGSIv2 format).

Listing 3: Example of smartparking:parkingsite

```
{
    "id": "urn:ngsi-ld:parkingsite:Aparcamiento:101",
    "type": "parkingsite",
    "@context": {
        "type": "StructuredValue",
        "value": [
            "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
                jsonld",
            "https://odins.org/smartParkingOntology/parkingsite-
                context.jsonld"
        ],
        "metadata": {}
    },
    "timestamp": {
        "type": "DateTime",
        "value": "2019-04-29T12:30:00Z",
```

```
        "metadata": {}
    },
    "name": {
        "type": "Text",
        "value": "Libertad",
        "metadata": {}
    },
    "numSpacePC": {
        "type": "Number",
        "value": 51,
        "metadata": {}
    },
    "totSpacePCCapacity": {
        "type": "Number",
        "value": 330,
        "metadata": {}
    },
    "maxHeight": {
        "type": "Number",
        "value": 1.8,
        "metadata": {}
    },
    "maxLength": {
        "type": "Number",
        "value": 5.1,
        "metadata": {}
    },
    "maxWidth": {
        "type": "Number",
        "value": 2.3,
        "metadata": {}
    },
    "payMthd": {
        "type": "StructuredValue",
        "value": [ "Cash", "PayPal"  ],
        "metadata": {}
    },
    "policyPC": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:policy:Aparcamiento:101:PrivateCar"
           ,
        "metadata": {
            "entityType": {
                "type": "Text",
                "value": "policy"
            }
        }
    },
```

```
    "isOpen": {
        "type": "boolean",
        "value": true,
        "metadata": {}
    },
    "monday": {
        "type": "StructuredValue",
        "value": [
            {
                "opens": "8:00",
                "closes": "20:00"
            }
        ],
        "metadata": {}
    },
    "carWash": {
        "type": "boolean",
        "value": false,
        "metadata": {}
    },
    "webSite": {
        "type": "Text",
        "value": "https://aparcamientosnewcapital.es/pf/avenida-
           libertad-murcia/#info",
        "metadata": {}
    },
    "location": {
        "type": "geo:json",
        "value": { "type": "Point",  "coordinates": [ -1.1336517,
            37.9894006 ] },
        "metadata": {}
    }
}
```


9.1.2 Policy

Policy attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity. In this case policy.	Mandatory

Attribute Name	Attribute Type	Description	Constraint
appliesDuring	StructuredValue	Indicates, using an array of JSON items, the time period where parking site policy will be applied (”startTime”: ”00:00”,”endTime”: ”23:59”) and the parking site rates considering weekday (”parkingRateWeekDay”), weekend (”parkingRateWeekEnd”), public holidays (”parkingRatePHolidays”) and disabled persons (”parkingRateWeekDayDis”, ”parkingRateWeekEndDis”, ”parkingRatePHolidaysDis”). The parking sites rates attributes are array of JSON items, where can determinate differents rates in differents time periods. For example: [{”forDuration”: { ”minutes”: 1 }, ”fromDuration”: 1, ”toDuration”: 300, ”monetaryCost”: 0.042, ”minParkingCharge”: { ”minutes”: 5 }, ”maxParkingCost”: 12.6 }, {”forDuration”: { ”minutes”: 1}, ”fromDuration”: 300, ”toDuration”: 720, ”monetaryCost”: 0.04, ”minParkingCharge”: 0, ”maxParkingCost”: 19.4}]	Mandatory
exclPHolidays	boolean	Indicates if public holidays will apply the parking site policy. Possible values true or false.	Mandatory
gracePeriod	StructuredValue	Indicates the parking site grace period. Its value is a JSON which has an attribute where specific an integer equal to or greater than zero. For example: { ”minutes”: 10 }.	Mandatory
maxDuration	StructuredValue	Indicates the parking site maximum duration. Its value is a JSON which has an attribute where specific an integer equal to or greater than zero. For example: { ”minutes”: 3600 }.	Mandatory
currency	Text	Indicates the parking site rate currency. See SO 4217 code values.	Mandatory

Table 13: Policy attributes

NGSIv2 Context Definition The following NGSIv2 context definition applies to the Policy entity.

Listing 4: smartparking:Policy:context

```
1 "@context": {
```

```
    //********** Policy **********
    "appliesDuring": "http://ontology.eil.utoronto.ca/icity/
       Parking/appliesDuring",
    "exclPHolidays": "http://ontology.eil.utoronto.ca/icity/
       Parking/excludesPublicHoliday",
    "gracePeriod": "http://ontology.eil.utoronto.ca/icity/Parking
       /hasGracePeriod",
    "minutes": "http://www.w3.org/2006/time#minutes",
    "maxDuration": "http://ontology.eil.utoronto.ca/icity/Parking
       /maxDuration",

    //********** Rate **********
    "currency": "http://purl.org/goodrelations/v1#hasCurrency",
    "parkingRateWeekDay": "http://ontology.eil.utoronto.ca/icity/
       Parking/ParkingRate",
    "parkingRateWeekEnd": "http://ontology.eil.utoronto.ca/icity/
       Parking/ParkingRate",
    "parkingRatePHolidays": "http://ontology.eil.utoronto.ca/
       icity/Parking/ParkingRate",
    "parkingRateWeekDayDis": "http://ontology.eil.utoronto.ca/
       icity/Parking/ParkingRate",
    "parkingRateWeekEndDis": "http://ontology.eil.utoronto.ca/
       icity/Parking/ParkingRate",
    "parkingRatePHolidaysDis": "http://ontology.eil.utoronto.ca/
       icity/Parking/ParkingRate",
    "forDuration": "http://ontology.eil.utoronto.ca/icity/Parking
       /forDuration",
    "minutes": "http://www.w3.org/2006/time#minutes",
    "fromDuration": "http://www.w3.org/2006/time#minutes",
    "toDuration": "http://www.w3.org/2006/time#minutes",
    "monetaryCost": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasMonetaryCost",
    "minParkingCharge": "http://ontology.eil.utoronto.ca/icity/
       Parking/minParkingCharge",
    "maxParkingCost": "http://ontology.eil.utoronto.ca/icity/
       Parking/maxParkingCost"
}
```

Example of policy entity The following is an example instance of the Policy entity (NGSIv2 format).

Listing 5: Example of smartparking:policy

```
{
    "id": "urn:ngsi-ld:policy:Aparcamiento:101:PrivateCar",
    "type": "policy",
    "@context": {
        "type": "StructuredValue",
        "value": [
```

```
            "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
            "https://odins.org/smartParkingOntology/policy-context.jsonld"
        ],
        "metadata": {}
    },
    "appliesDuring": {
        "type": "StructuredValue",
        "value": [
            {
                "startTime": "00:00",
                "endTime": "23:59",
                "parkingRateWeekDay": [
                    {
                        "forDuration": { "minutes": 1 },
                        "fromDuration": 1,
                        "toDuration": 300,
                        "monetaryCost": 0.042,
                        "minParkingCharge": { "minutes": 5 },
                        "maxParkingCost": 12.6
                    },
                    {
                        "forDuration": { "minutes": 1} ,
                        "fromDuration": 300,
                        "toDuration": 720,
                        "monetaryCost": 0.04,
                        "minParkingCharge": 0,
                        "maxParkingCost": 19.4
                    }
                ],
                "parkingRateWeekEnd": [],
                "parkingRatePHolidays": [],
                "parkingRateWeekDayDis": [],
                "parkingRateWeekEndDis": [],
                "parkingRatePHolidaysDis": []
            }
        ],
        "metadata": {}
    },
    "exclPHolidays": {
        "type": "boolean",
        "value": true,
        "metadata": {}
    },
    "gracePeriod": {
        "type": "StructuredValue",
        "value": { "minutes": 10 },
```

```
            "metadata": {}
        },
        "maxDuration": {
            "type": "StructuredValue",
            "value": { "minutes": 3600 },
            "metadata": {}
        },
        "currency": {
            "type": "Text",
            "value": "EUR",
            "metadata": {}
        }
}
```

9.1.3 Sector

Sector attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity. In this case sector.	Mandatory
timestamp	DateTime	Indicates the date/ time when the entity was last observed in ISO 8601 format. The value of this will be set by the server when the entity was observed, if the entity has not been observed it may have a null value.	Optional
name	Text	Indicates the name of sector.	Recommended
numSpace	Number	Indicates the number of free parking site spaces. Its possible values: integer equal to or greater than zero.	Mandatory Recommended
numSpaceCapacity	Number	Indicates the number of parking site. Its possible values: integer greater than zero.	Recommended
policy	RelationShip	Indicates the parking site policy. This attribute value must contain an identifier of an existing policy entity.	Recommended
policyPHolidays	RelationShip	Indicates the parking site policy (public holidays). This attribute value must contain an identifier of an existing policy entity.	Optional
location	geo:json	The location point of the parking meter. See GeoJSON Specification (RFC 7946).	Recommended

Attribute Name	Attribute Type	Description	Constraint

Table 14: Sector attributes

NGSIv2 Context Definition The following NGSIv2 context definition applies to the sector entity.

Listing 6: smartparking:sector:context

```
"@context": {
    "name": "http://purl.org/goodrelations/v1#name",
    "numSpace": "http://ontology.eil.utoronto.ca/icity/Parking/
       hasNumSpacePrivateCar",
    "numSpaceCapacity": "http://ontology.eil.utoronto.ca/icity/
       Parking/hasVehicleCapacity",
    "policyPHolidays": "http://ontology.eil.utoronto.ca/icity/
       Parking/ParkingPolicy",
    "location": "https://schema.org/location"
}
```

Example of sector entity The following is an example instance of the sector entity (NGSIv2 format).

Listing 7: Example of smartparking:sector

```
{
    "id": "urn:ngsi-ld:sector:Sector:1",
    "type": "sector",
    "@context": {
        "type": "StructuredValue",
        "value": [
            "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
               jsonld",
            "https://odins.org/smartParkingOntology/sector-
               context.jsonld"
        ],
        "metadata": {}
    },
    "timestamp": {
        "type": "DateTime",
        "value": "2019-04-29T12:30:00Z",
        "metadata": {}
    },
    "name": {
        "type": "Text",
        "value": "Sector 1",
        "metadata": {}
    },
    "numSpace": {
        "type": "Number",
```

```
        "value": 200,
        "metadata": {}
    },
    "numSpaceCapacity": {
        "type": "Number",
        "value": 787,
        "metadata": {}
    },
    "policy": {
        "type": "Relationship", "value": "urn:ngsi-ld:policy:
           Sector:1",
        "metadata": {
            "entityType": {
                "type": "Text",
                "value": "policy"
            }
        }
    },
    "policyPHolidays": {
        "type": "Relationship",
        "value":"urn:ngsi-ld:policy:Sector:1:PublicHoliday",
        "metadata": {
            "entityType": {
                "type": "Text",
                "value": "policy"
            }
        }
    },
    "location": {
        "type": "geo:json",
        "value": {
            "type": "Polygon",
            "coordinates": [
                    [  [-1.134584, 37.996461], [-1.127803, 37.997
                       983], [-1.126236, 37.995582],
                    [-1.124928, 37.994753], [-1.124348, 37.992825
                       ], [-1.125099, 37.991743],
                    [-1.124992, 37.989257], [-1.127395, 37.989561
                       ], [-1.130271, 37.989341],
                    [-1.131193, 37.990711], [-1.131386, 37.992098
                       ], [-1.134584, 37.996461]  ]
            ]
        },
        "metadata": {}
    }
}
```

9.1.4 Parking meter

Parking meter attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity. In this case parkingmeter.	Mandatory
timestamp	DateTime	Indicates the date/ time when the entity was last observed in ISO 8601 format. The value of this will be set by the server when the entity was observed, if the entity has not been observed it may have a null value.	Optional
name	Text	Indicates the name of parking meter.	Recommended
sector	RelationShip	Indicates the sector of parking meter. This attribute value must contain an identifier of an existing sector entity.	Mandatory
location	geo:json	The location point of the parking meter. See GeoJSON Specification (RFC 7946).	Recommended

Table 15: Parking meter attributes

NGSIv2 Context Definition The following NGSIv2 context definition applies to the parking meter entity.

Listing 8: smartparking:parkingmeter:context

```
"@context": {
    "name": "http://purl.org/goodrelations/v1#name",
    "sector": "https://odins.org/smartParkingOntology/sector",
    "location": "https://schema.org/location"
}
```

Example of parking meter entity The following is an example instance of the parking meter entity (NGSIv2 format).

Listing 9: Example of smartparking:parkingmeter

```
{
    "id": "urn:ngsi-ld:parkingmeter:Parquimetro:166",
    "type": "parkingmeter",
    "@context": {
        "type": "StructuredValue",
        "value": [
            "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
               jsonld",
```

```
            "https://odins.org/smartParkingOntology/parkingmeter-context.jsonld"
        ],
        "metadata": {}
    },
    "timestamp": {
        "type": "DateTime",
        "value": "2019-04-29T12:30:00Z",
        "metadata": {}
    },
    "name": {
        "type": "Text",
        "value": "BUENOS LIBROS",
        "metadata": {}
    },
    "sector": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:sector:Sector:1",
        "metadata": {
            "entityType": {
                "type": "Text",
                "value": "sector"
            }
        }
    },
    "location": {
        "type": "geo:json",
        "value": {
            "type": "Point",
            "coordinates": [ -1.130981829, 37.99491059 ]
        },
        "metadata": {}
    }
}
```


9.1.5 Ticket

Ticket attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity. In this case ticket.	Mandatory

Attribute Name	Attribute Type	Description	Constraint
parkingmeter	Relationship	Indicates the parking meter that issued the ticket. This attribute value must contain an identifier of an existing parking meter entity.	Mandatory
fromDate	Datetime	Indicates the initial date/ time that the instance of the entity was created in ISO 8601 format, also the initial booked period.	Mandatory
toDate	Datetime	Indicates the ended booked period in ISO 8601 format.	Mandatory
duration	Number	Indicates the duration (in minutes) of booked period of time. Its possible values: integer greater than zero.	Mandatory
rate	Number	Indicates the rate relates to the issued ticket. Its possible values: float greater than zero.	Mandatory
price	Number	Indicates the price relates to the issued ticket. Its possible values: float greater than zero.	Mandatory
payMethod	Text	Indicates the payment option relates to the issued ticket. Possible values:"Cash", "PayPal", "AmericanExpress", "DinersClub", "Discover", "JCB", "MasterCard", "VISA"...	Mandatory

Table 16: Ticket attributes

NGSIv2 Context Definition The following NGSIv2 context definition applies to the parking meter ticket entity.

Listing 10: smartparking:ticket:context

```
"@context": {
    "parkingmeter": "https://odins.org/smartParkingOntology/
       parkingmeter",
    "fromDate": "https://odins.org/smartParkingOntology/
       bookedPeriod",
    "toDate": "https://odins.org/smartParkingOntology/
       bookedPeriod",
    "duration": "https://odins.org/smartParkingOntology/duration"
       ,
    "rate": "https://odins.org/smartParkingOntology/rate",
    "price": "https://schema.org/importe",
    "payMethod": "https://schema.org/PaymentMethod"
}
```

Example of parking meter ticket entity The following is an example instance of the parking meter ticket entit (NGSIv2 format).

Listing 11: Example of smartparking:ticket

```
{
    "id": "urn:ngsi-ld:ticket:Ticket:166",
    "type": "ticket",
    "@context": {
        "type": "StructuredValue",
        "value": [
            "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
               jsonld",
            "https://odins.org/smartParkingOntology/ticket-
               context.jsonld"
        ],
        "metadata": {}
    },
    "parkingmeter": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:parkingmeter:Parquimetro:166",
        "metadata": {
            "entityType": {
                "type": "Text",
                "value": "parkingmeter"
            }
        }
    },
    "fromDate": {
        "type": "DateTime",
        "value": "2019-05-06T12:35:00Z",
        "metadata": {}
    },
    "toDate": {
        "type": "DateTime",
        "value": "2019-05-06T13:20:00Z",
        "metadata": {}
    },
    "duration": {
        "type": "Number",
        "value": 2700,
        "metadata": {}
    },
   "rate": {
        "type": "Text",
        "value": "Azul",
        "metadata": {}
    },
    "price": {
        "type": "Number",
        "value": 1.3,
        "metadata": {}
```

```
    },
    "payMethod": {
        "type": "Text",
        "value": "Cash",
        "metadata": {}
    }
}
```

9.2 Grasse

The data model for the information presented here for Grasse is NGSI-LD.

9.2.1 Camera

Camera attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of an URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity.	Mandatory
monitors	Relationship	References the monitored sensitive zone by the camera.	Mandatory
detects	Relationship	References the detected wild deposit events.	Mandatory
createdAt	TemporalProperty	Indicates the date/ time that the instance of the entity was created in ISO 8601 format. The value of this will be set by the server when the entity was created.	Mandatory
modifiedAt	TemporalProperty	Indicates the date/ time when the entity was last modified in ISO 8601 format. The value of this will be set by the server when the entity was modified, if the entity has not been modified it may have a null value.	Optional
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD	Recommended
Model	Relationship	A reference to the associated camera Model.	Mandatory
installedAt	DateTime	Indicates the date/time at which date and time the camera was installed (nominally in UTC).	Recommended
location	GeoProperty	The installed / last known geo location (point/ polygon) of the camera.	Mandatory

Attribute Name	Attribute Type	Description	Constraint
operationSpace	GeoProperty	The geographic location (point/ polygon) in which the camera is active.	Mandatory
batteryLevel	Property	Battery level. It must be equal to: 1.0 When the battery charge is full; 1.0 When the battery charge empty;Null when it cannot be determined.	Recommended
numberOfPictures	Properties	Indicates the number of pictures which have been stored	Mandatory.

Table 17: Camera attributes

Example of Camera Entity The following is an example instance of the Camera entity.

Listing 12: Example of camera instance

```
{
    "@context": [
        "https://forge.etsi.org/gitlab/NGSI-LD/NGSI-LD/raw/master/coreContext/ngsi-ld-core-context.json",
        "https://model.eglobalmark.com/NGSI-LD-Entities/general/Camera-context.jsonld"
    ],
    "id": "urn:ngsi-ld:Camera:ba2dd9-f57f",
    "type": "camera",
    "createdAt": "2017-01-01T01:20:00Z",
    "modifiedAt": "2017-05-04T12:30:00Z",
    "entityVersion": 2.0,


    "monitors"{
        "type": "Relationship",
        "object": "urn:egm:SensitiveSiteModel:c1be2e-d9e7"
    },


    "detects"{
        "type": "Relationship",
        "object": "urn:egm:WildDepositModel:c1be2e61-k9e7"
    },

    "Model": {
        "type": "Relationship",
        "object": "urn:panasonic:CameraModel:d1be2e61-d9e7"
    },

    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
```

```
            "coordinates": [
                -104.99404,
                39.75621
            ]
        }
    },

    "batteryLevel": {
        "type": "Property",
        "value": 0.7,
        "observedAt": "2017-05-04T12:30:00Z"
    },

    "numberOfPictures": {
        "type": "Property",
        "value": 167,
        "observedAt": "2017-05-04T12:42:00Z"
    },

    "operationSpace": {
        "type": "Property",
        "value": 0.7,
        "observedAt": "2017-05-04T12:30:00Z"
    },


}
```

9.2.2 Sensitive Site

This entity contains a description of a Sensitive site.

Sensitive site attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity.	Mandatory
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD	Recommended
name	Property	Indicates the name of the monitored site.	Recommended
location	GeoProperty	The installed geo location point of the monitored site.	Recommended

Table 18: Sensitive site attributes

Example of Sensitive site Entity The following is an example instance of the Sensitive Site entity.

Listing 13: Example of wastemanagement:SensitiveSite

```
{
    "@context": [
        "https://forge.etsi.org/gitlab/NGSI-LD/NGSI-LD/raw/master
           /coreContext/ngsi-ld-core-context.json",
        "https://model.eglobalmark.com/NGSI-LD-Entities/waste/
           SensitiveSite-context.jsonld"
    ],
    "id": "urn:egm:Camera:ba2dd9-f57f",
    "type": "SensitiveSite",
    "createdAt": "2017-01-01T01:20:00Z",
    "modifiedAt": "2017-05-04T12:30:00Z",
    "entityVersion": 2.0,
    "name": "Chemin des Canebiers",
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                -104.99404,
                39.75621
            ]
        }
    },
}
```

9.2.3 Wild deposit

This entity contains a description of a wild deposit.

Wild deposit attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity.	Mandatory
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD	Recommended
observedAt	TemporalProperty	Timestamp of the wild deposit event detection.	Mandatory
registrationPlate	car plate number of the offender	Mandatory	

Attribute Name	Attribute Type	Description	Constraint
wastetype	Property	Type of waste detected by the system	Recommended

Table 19: Wild deposit attributes

Example of wild deposit Entity The following is an example instance of the Sensitive Site entity.

Listing 14: Example of wastemanagement:SensitiveSite

```
{
    "@context": [
        "https://forge.etsi.org/gitlab/NGSI-LD/NGSI-LD/raw/master
           /coreContext/ngsi-ld-core-context.json",
        "https://model.eglobalmark.com/NGSI-LD-Entities/waste/
           WildDeposit-context.jsonld"
    ],
    "id": "urn:egm:WildDeposit:b219-f57f",
    "type": "SensitiveSite",
    "observedAt": "2017-01-01T03:48:00Z",
    "entityVersion": 2.0,
    "registrationPlate": "CH-208-DL",
    "wasteType": "Washing Machine",
  }
```

9.3 Hakusan

Video contents and/or pictures by cameras and data by environmental sensors are collected using the formats reported in the following examples. The model for data is proprietary.

Common service data			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
Temperature	String	Temperature of when the event occurred	Recommended
Humidity	String	Humidity of when the event occurred	Recommended
Moisture	String	Moisture of the day (It shows whether it is raining or not)	Recommended
Amount of Sunlight	String	Amount of sunlight in the day	Recommended
Location	Strings	Location of Camera	Mandatory
Video Contents	Picture or Movie	Picture or movie captured by the camera	Mandatory

Table 20: Common service data

Data of specific services is produced according to specifications and functions of deployed devices. Data consists of these types as shown in Figure 18.

- Monitoring data without request, i.e. periodical monitoring, trigger base monitoring, etc.

- Monitoring data by request
- Control to devices

For example, in wildlife monitoring, the following data is transferred across this infrastructure. The model for data is proprietary.

- Direction of moving animals (Alert of animal approaching)
- Animal detection
- Monkey detection
- Trap status
- Control data to devices

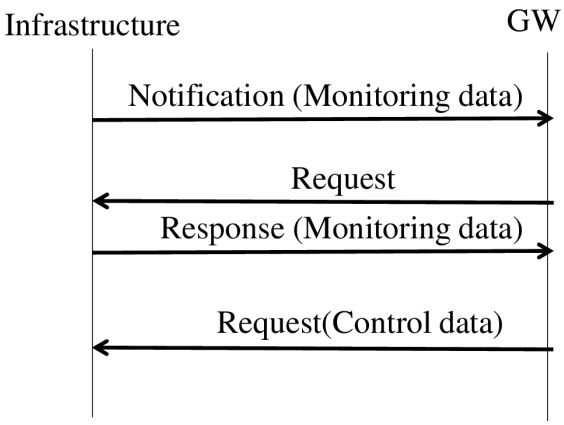

Figure 18: Type of data from/to an Infrastructure

9.4 Kumamoto

The data model for the information presented here for Kumamoto is NGSI-LD.

9.4.1 Camera

This entity contains a harmonised description of a generic camera. This entity provides an essentially static description of a generic camera and is therefore applicable to all IoT segments and related IoT applications.

Camera attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity.	Mandatory
type	@type	Defines the type of the entity.	Mandatory

Attribute Name	Attribute Type	Description	Constraint
location	GeoProperty	The installed/last known geo-location (point/ polygon) of the camera.	Mandatory
createdAt	TemporalProperty	Indicates the date/time that the instance of the entity was created in ISO 8601 format. The value of this will be set by the server when the entity was created.	Mandatory
Source	Property	Specifies the URL to the source of this data (either organisation or where relevant more specific source)	Recommended
dataProvider	Property	Property Specifies the URL to information	Recommended
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD.	Recommended
deviceModel	Relationship	A reference to the associated Device Model for this device.	Mandatory
description	Property	An optional description of this device.	Recommended
owner	Relationship	Reference to the owner or owners of the device as either a Schema.org person or organization. https://schema.org/Person or https://schema.org/Organization	Optional
supportedProtocols	Property	A list of communications protocols supported by the device.	Option
online	Property	The communication status of this device. A logical representation of Offline (false) or Online (true).	Option
BatteryLevel	Property	Battery level. It must be equal to: 1.0 When the battery charge is full; 1.0 When the battery charge empty; Null when it cannot be determined.	Optional
FileName	Property	Indicate the file name and extension of the still picture or movie captured by the camera	Mandatory

Table 21: Camera attributes

9.4.2 Human Detector

This entity contains a harmonised description of a virtual human detector. This entity provides an essentially static description of a virtual human detector and is therefore applicable to all IoT segments and related IoT applications.

Human Detector attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory
type	@type	Defines the type of the entity.	Mandatory
location	GeoProperty	The installed/last known geo-location (point/ polygon) of the camera.	Mandatory
createdAt	TemporalProperty	Indicates the date/time that the instance of the entity was created in ISO 8601 format. The value of this will be set by the server when the entity was created.	Mandatory
Source	Property	Specifies the URL to the source of this data (either organisation or where relevant more specific source).	Recommended
dataProvider	Property	Property Specifies the URL to information.	Recommended
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD.	Recommended
description	Property	An optional description of this virtual device.	Recommended
softwareVersion	Property	The (manufacturer specific) software version of this virtual device.	Recommended
NumberOfHuman	Property	The number of the human detected by the camera.	Optional
DetectHuman	Property	Notification of human detection by the camera.	Mandatory if the NumberOfHuman is not supported.

Table 22: Human Detector attributes

9.4.3 Face Feature Detector

This entity contains a harmonised description of a virtual face feature detector. This entity provides an essentially static description of a virtual face feature detector and is therefore applicable to all IoT segments and related IoT applications.

Face Feature Detector attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for an instance of the entity either in the form of a URI (i.e. either a publicly accessible URL or a URN).	Mandatory

Attribute Name	Attribute Type	Description	Constraint
type	@type	Defines the type of the entity.	Mandatory
location	GeoProperty	The installed/last known geo-location (point/ polygon) of the camera.	Mandatory
createdAt	TemporalProperty	Indicates the date/time that the instance of the entity was created in ISO 8601 format. The value of this will be set by the server when the entity was created.	Mandatory
Source	Property	Specifies the URL to the source of this data (either organisation or where relevant more specific source).	Recommended
dataProvider	Property	Property Specifies the URL to information.	Recommended
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD.	Recommended
description	Property	An optional description of this virtual device.	Recommended
softwareVersion	Property	The (manufacturer specific) software version of this virtual device.	Recommended
FaceLandmarInfo	Property	The encrypted face landmark information to be used for face matching	Mandatory
KeyInfo	Property	The key info to generate the key to decrypt the face landmark information.	Mandatory

Table 23: Face Feature Detector attributes

9.4.4 LiDAR

This entity contains a harmonised description of a LiDAR. This entity provides an essentially static description of LiDAR and is therefore applicable to all IoT segments and related IoT applications.

LiDAR attributes			
Attribute Name	**Attribute Type**	**Description**	**Constraint**
id	@id	Provides a unique identifier for LiDAR in the form of a URI.	Mandatory
type	@type	Defines the type of the entity.	Mandatory
location	GeoProperty	The installed/last known geo-location (point/ polygon) of the LiDAR.	Mandatory
createdAt	TemporalProperty	Indicates the date/time that the instance of the entity was created in ISO 8601 format. The value of this will be set by the server when the entity was created.	Mandatory

Attribute Name	Attribute Type	Description	Constraint
Source	Property	Specifies the URL to the source of this data (either organisation or where relevant more specific source)	Recommended
entityVersion	Property	The entity specification version as a number. A version number of 2.0 or later denotes the entity is represented using NGSI-LD.	Recommended
deviceModel	Relationship	A reference to the associated Device Model for this device.	Mandatory
description	Property	An optional description of this device.	Recommended
owner	Relationship	Reference to the owner or owners of the device as either a Schema.org person or organization. https://schema.org/Person or https://schema.org/Organization.	Optional
supportedProtocols	Property	A list of communications protocols supported by the device.	Option
online	Property	The communication status of this device. A logical representation of Offline (false) or Online (true).	Option
BatteryLevel	Property	Battery level. It must be equal to: 1.0 When the battery charge is full; 1.0 When the battery charge empty; Null when it cannot be determined.	Optional
DetectMovingObject	Property	Notification of moving object detection by the LiDAR	Mandatory

Table 24: LiDAR attributes

References

[1] A. Detti and H. Nakazato, "Fed4IoT - Deliverable D2.2 System Architecture - First Release," March 2019.

[2] Node-RED Web Page. [Online]. Available: https://nodered.org

[3] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1561572249627&uri=CELEX:32016R0679

[4] Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation 2016/679. [Online]. Available: https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=612053

[5] What does the GDPR say about automated decision-making and profiling? [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/automated-decision-making-and-profiling/what-does-the-gdpr-say-about-automated-decision-making-and-profiling

[6] Article 29 Data Protection Working Party, Opinion 01/2017 on the Proposed Regulation for the ePrivacy Regulation. [Online]. Available: https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=610140

[7] Processor and Controller obligations under GDPR: a cheat-sheet. [Online]. Available: https://blog.returnpath.com/processor-and-controller-obligations-under-gdpr-a-cheat-sheet

[8] The position of IT service providers (data processors) under the GDPR. [Online]. Available: https://creobis.eu/nl/the-position-of-it-service-providers-data-processors-under-the-gdpr

[9] European Commission - Press release EU News 176/2018 - The European Union and Japan agreed to create the world's largest area of safe data flows. [Online]. Available: https://eeas.europa.eu/delegations/japan/48487/european-union-and-japan-agreed-create-worlds-largest-area-safe-data-flows_en