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 D2.1:

Use Case and Requirements

Deliverable Type:	Report
Deliverable Number:	D2.1
Contractual Date of Delivery to the EU:	31/12/2018
Actual Date of Delivery to the EU:	31/12/2018
Title of Deliverable:	Use Case and Requirements
Work package contributing to the Deliverable:	WP2
Dissemination Level:	Public
Editor:	Antonio Skarmeta (OdinS), Kenichi Nakamura (PAN)
Author(s):	Juan A. Martinez (OdinS), Wenbin LI (EGM), Giuseppe Tropea (CNIT), Kenji Kanai (WAS), Hidenori Nakazato (WAS), Hiroaki Mukai (KIT)
Internal Reviewer(s):	Andrea Detti (CNIT)
Abstract:	This document summarizes the work carried out during Task2.1: Use cases and

	requirements. We have identified a series of use cases and applications motivating the need for Fed4IoT technologies. Use cases spawn from partners' experience and current deployments, and report how those scenarios will benefit from Fed4IoT technologies such as virtualized IoT deployments and context information sharing. Moreover, these use cases have guided us to make a first approach at sketching the functional architecture of the project. We also derive a set of high-level requirements that are going to guide our implementation.
Keyword List:	Virtualized sensors, Interoperability, Federation, Smart Cities, Use Cases.

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
GUI	Graphical User Interface
RPZ	Regulated Parking Zone
NGSI-LD	Next Generation Services Interface – Linked Data
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology

Table 1: Abbreviations

Fed4IoT Glossary

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

FED4IOT GLOSSARY TERM	DEFINITION
IoT Slice	A virtualized environment specific to IoT applications
Federated Data Domain	Collection of contextual/IoT information gathered from an heterogeneous set of IoT systems
Virtual Thing	Virtual entity producing virtual IoT information derived from real IoT information available in the Federated Data Domain
Tenant Data Domain	Collection of contextual/IoT information generated by i) instances of virtual things allocated by the tenant and/or ii) real things own by the tenant
IoT Slice	A virtualized isolated IoT environment of a tenant whose data composes the Tenant Data Domain and is exposed through a standard IoT framework (e.g. oneM2M, FiWARE, etc.)

Table 2: Fed4IoT Dictionary

1 Introduction

1.1 Deliverable Rationale

This deliverable presents the work carried out during Task2.1 where we have identified a selection of five use cases showcasing application of our Fed4IoT framework to different domains such as parking in the city, person finding or wildlife monitoring.

1.2 Quality review

The internal reviewer responsible of this deliverable is: Andrea Detti (CNIT).

VERSION CONTROL TABLE			
VERSION N.	PURPOSE/CHANGES	AUTHOR	DATE
0.1	Added Smart Parking Use Case	Juan A. Martinez, Antonio Skarmeta (OdinS)	2018/10/27
0.2	Improved contribution to Smart Parking Use Case	Juan A. Martinez, Antonio Skarmeta (OdinS)	2018/11/27
0.3	Waste Management use case	Wenbin Li (EGM)	2018/12/12
0.4	Cross Border Person Finding	Kenichi Nakamura (PAN)	2018/12/12
0.5	Wildlife Monitoring	Hiroaki Mukai (KIT)	2018/12/12
0.6	Citizen-made IoT Applications	Hidenori Nakazato (WAS)	2018/12/12
0.7	Overall revision	Andrea Detti (CNIT)	2018/12/12
0.8	Requirements clean-up	Giuseppe Tropea (CNIT)	2018/12/20
0.9	Harmonization	Juan A. Martinez (OdinS), Giuseppe Tropea (CNIT)	2018/12/21
0.10	Internal revision	Giuseppe Tropea (CNIT)	2018/12/21
0.11	Final internal revision	Andrea Detti (CNIT), Kenichi Nakamura (PAN)	2018/12/27

1.3 Executive summary

1.3.1 Deliverable description

This deliverable firstly introduces two key concepts of the Fed4IoT project: the concept of federation of heterogeneous IoT systems and the concept of thing virtualization (Section 2).

Additionally, according to the work defined in Task 2.1, we have included five use cases: Smart Parking (EU), Cross Border Person Finder (EU,JP), Waste Management (EU), Citizen-Made IoT Applications (JP) and Wildlife monitoring (JP).

For each of the sections describing the five use cases we have defined a common structure to provide a unified vision of them, in terms of the different elements and aspects that have been considered. We have identified the following sections: Application and Use Case description, Architecture, Data Sets, Deployment View and FG-DPM Unified Use Case Template.

Finally, we conclude this deliverable by reporting the requirements coming out from these use cases, and then by summarizing the results.

Structure of Each Use Case

Each use case is organized into several sections that describe its details. All of the section names are self-explanatory and all of the use cases follow the same structure. Some additional explanation is, however, necessary about the FG-DPM section of each use case.

Within the ITU-T, several project partners, including NEC, EGM, PAN and KIT, belong to the Focus Group on Data Processing and Management (FG-DPM). Accordingly, the Fed4IoT project plans to contribute to the focus group with inputs on data model, security, interoperability and reliability. In order to meet this objective, already at this stage, the project considered some alignment to ITU-T FG-DPM activity as an approach that can ease possible ITU-T standardization. FG-DPM WG1 agreed (6 Feb 2018) on a template version for recommended usage by DPM use cases contributors. Thus, we decided to also use this official FG-DPM template in use cases' descriptions within this deliverable in order to facilitate a possible submission to ITU-T FG-DPM.

Structure of the ITU-T FG-DPM Template

The general guidelines (not field-specific) for filling in the template are the following ones:

- Each grey box is a use case entry
- Each area and bullet represents something the use case contributor may want to consider when describing the use case. NOTE - It is not mandatory to fill in all fields of the template.
- It should be paid attention to have as much as possible terminology (and taxonomy) alignment across the whole use case description.
- NOTE - The current version of this document does not contain definitions of relevant terms used in the template, but an updated version of this document (targeted as output of next 1-3 May FG-

DPM meeting) will provide a set of definitions associated to the template, according to the definitions under development in FG-DPM D0.1 “DPM Terms and Definitions, Taxonomies”.

- It is suggested to consider the different natures of experts required to complete the template

Concerning the field-specific guidelines, they are directly provided in the master template which is provided in the Annex 1.

1.3.2 Summary of results

The main result we highlight in this deliverable (under the umbrella of Task 2.1) is the identification of five use cases that can showcase the application of our Fed4IoT system, thereby producing related system and application key requirements.

Another important result of this document is the definition of the two fundamental concepts that are the baseline of our Fed4IoT project, which are the concept of data federation (aka context information sharing) and the concept of virtualized things. We foresee that this result, as the project further develops, will in turn lead to a more powerful representation of the information which can be shared among different users or tenants.

In this document we show how the Fed4IoT concepts may be applied to the use cases, and how the scenarios benefit from those concepts.

By collecting key requirements, we are able to see what layers the project’s challenges mainly impact, and we start to understand how some challenges are similar across use-cases.

2 Key Concepts

Central to the Fed4IoT project are two high-level technological objectives:

- To **federate** heterogeneous IoT systems, in order to form a cross-domain set of shared data (contextual information).
- To **design virtualization** techniques that can provide IoT systems "as a service", implementing the concept of an *IoT slice*.

As shown in Figure 1, we have a *federated data domain* formed of contextual information that is exposed by real things in heterogeneous IoT systems. IoT virtualization techniques developed by the project will fetch information from the *federated data domain* in order to create virtual things, which generate virtual data items. Exemplary virtual things are: a virtual thermometer that generates virtual temperature values that are an exact copy of the values of a real thermometer available in the federated data domain, a virtual thermometer that aggregates federated data from four differently located temperature sensors and produce virtual temperature values with higher accuracy, a virtual camera producing pictures captured by a federated video stream of a flying drone, which surveys premises from many different viewpoints, etc. Generally speaking, a virtual thing may be either a copy of a real thing, or an entity producing new information coming out from the elaboration of data generated by federated real things.

As shown in Figure 1, Fed4IoT offers *IoT slices* to tenants. An IoT slice is a virtual isolated IoT environment, solely dedicated to a specific *tenant* for running her applications. The IoT data set available in the slice forms the *Tenant Data Domain* which is made up of i) data items generated by instances of virtual things "rented" by the tenant; ii) data items coming from connected real things already under the tenant's ownership. For instance, a tenant that want to develop a watering system for her house, can have own thermometers and watering devices (actuator) and can rent a virtual hydrometer because she hasn't a real one.

The data items of the Tenant Data Domain are exposed by the IoT slice through a standard IoT framework (such as FIWARE or oneM2M, for example) that can be chosen by the tenant. To make a comparison with the classic cloud world we could say that: as in a cloud a tenant rents virtual ICT resources (virtual CPU, virtual storage, virtual network etc.), as in Fed4IoT a tenant rents virtual things; as in a cloud a tenant choses the operative system of its virtual server, as in Fed4IoT a tenant choses the IoT framework handling its virtual/real things.

An *orchestration* capability (Figure 1) is needed in order to manage resources, slices and tenants.

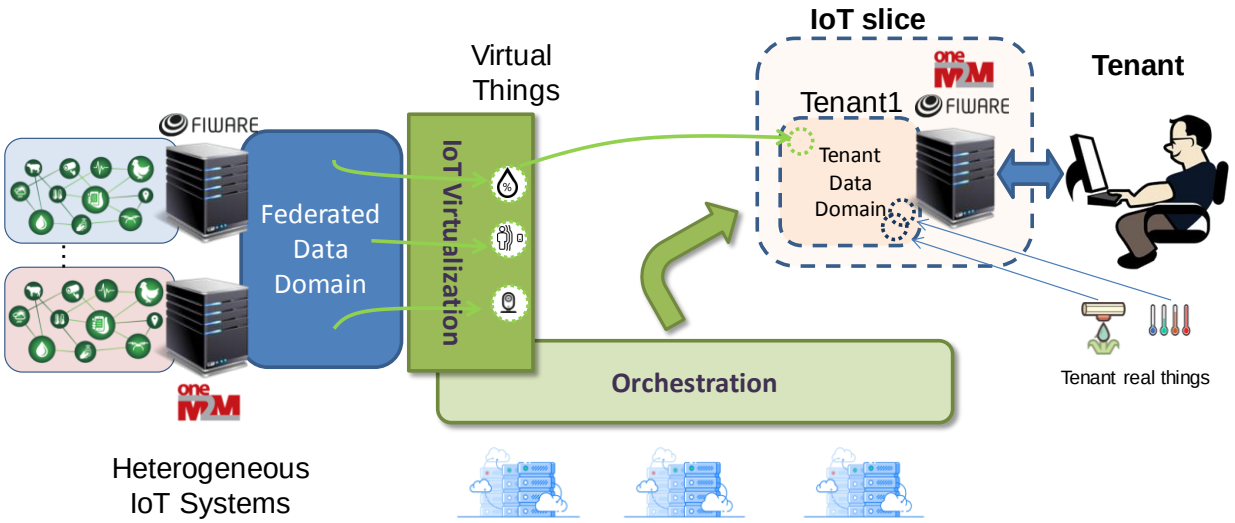


Figure 1. The concept of IoT slice

Broadly speaking, the above concepts can be applied to two different generic usage profiles: public and private. Just like there is a dichotomy (and a debate on which is preferable when) between public cloud and private cloud for the virtualization of servers and applications, the same may be envisioned of IoT services' virtualization.

In public cloud, your business is able to deploy services without owning any hardware resources or operating systems, by being assigned and exploiting a dedicated slice via Internet, which is offered to you by a "public" provider owning the resources. In private cloud, which may be viewed as an extension to your business' data centre, resources are not shared with others, and are owned by your company and operated on a local communications infrastructure.

Similarly, the Fed4IoT slicing can be applied both to public and to private scenarios.

In a public scenario (see Figure 2) we have a **decoupling between the owner of the IoT systems and the owner of the IoT applications**. This is going to be crucial in large-scale environments, such as smart cities, where the city owns several arrays of sensors and sells the raw data streams to a public provider of IoT slices, which acts as intermediary between the huge amount of raw resources and the applications. Designers of smart-city Apps can rent IoT slices as a service from the public provider, having access to perhaps thousands of selected virtual things and to an operating environment of choice (e.g. oneM2M) to develop the desired App.

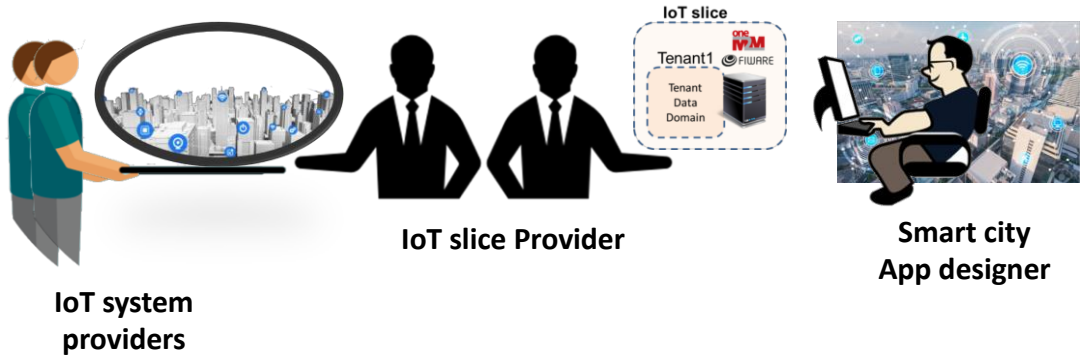


Figure 2. Public Scenario

Conversely, in a private scenario (see Figure 3), where the same actor owns both the sensors and the applications running on top of them, we aim at **decoupling the newly designed IoT applications from the IoT services that are already running in production**. For instance, a company operating a smart harbour system may have a robust solution in place, where numerous real sensors are exploited by the existing application. A novel version or an enhancement can be safely tested in a slice on virtual sensors, prior to deployment in the production environment. Such a decoupling of production from test environments allows for easy upgrading to newer releases of the oneM2M or FIWARE middleware, for instance, or it enables creation of mini environments for testing new services or fast prototyping. Security-wise, a choice can be made as to what expose to attacks from the outside. In short, a private approach to IoT virtualization offers the same advantages a private cloud is nowadays offering to companies deploying their servers in virtual vs. bare-metal.

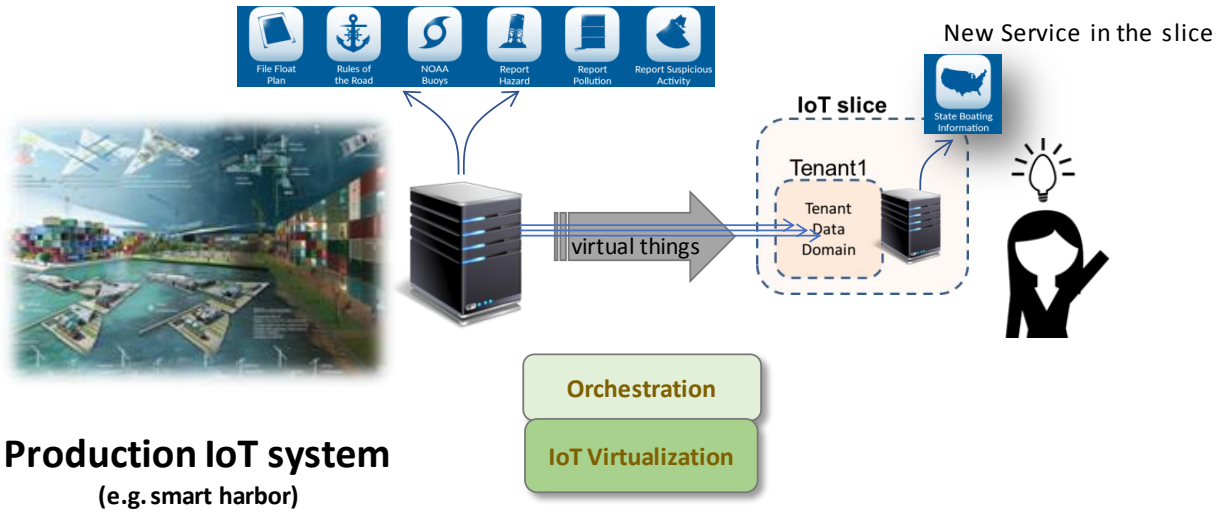


Figure 3. Private Scenario

The parallelism that can be established between virtualization specific to the IoT world and generic virtualization as we know it today is clearly between the sensors and the CPU/peripherals, and between the OS and the M2M frameworks such as oneM2M and FIWARE Orion.

In the following we will try to estimate, for each use case we present, what the most appropriate strategy (whether public or private) would be.

3 Use Case – Smart Parking

3.1 Application and Use Case Description

One of the most common problems that big cities cope with is daily traffic congestion caused by commuting citizens and daily activities carried out by the own citizens inside the city. One of the main causes for having a large number of vehicles wandering along the city is no other than finding a free parking spot in a certain destination area. For this reason, we have considered this concern by putting it as a relevant use case that our Fed4IoT project must address, providing a solution to allow them to reduce the amount of time for such a difficult activity.

In this sense, Murcia is a medium-size city in the south east of Spain with a population of 450.000 residents. This number increases by about 5.000 residents per year, with a noteworthy distribution of people among the city districts. Because of this and the fact that Murcia is the capital of the Murcia Region, it has been detected a dramatic rise of accesses to the city centre in the last years. Day by day, commuters, tourists and families traveling by car collapse the core of Murcia with the intention to park at commerce, financial and historical areas.

The aim of this Smart Parking use case is that of taking advantage of Fed4IoT framework to provide a service that tracks the state of the parking spots to provide the drivers with this information beforehand, so they can better plan where they will park. This will also lead to a more fluid traffic in the centre of the city, because there will be less drivers wandering, looking for a parking spot in areas that do not have any available.

This solution will provide a GUI, as depicted in Figure 4, allowing the user to specify both the destination location, as well as the time when she will arrive by presenting a map-based web interface/App. Once the selection is made, our Smart Parking solution will make a complex reasoning to generate an informed recommendation about the best destination area to park the vehicle.

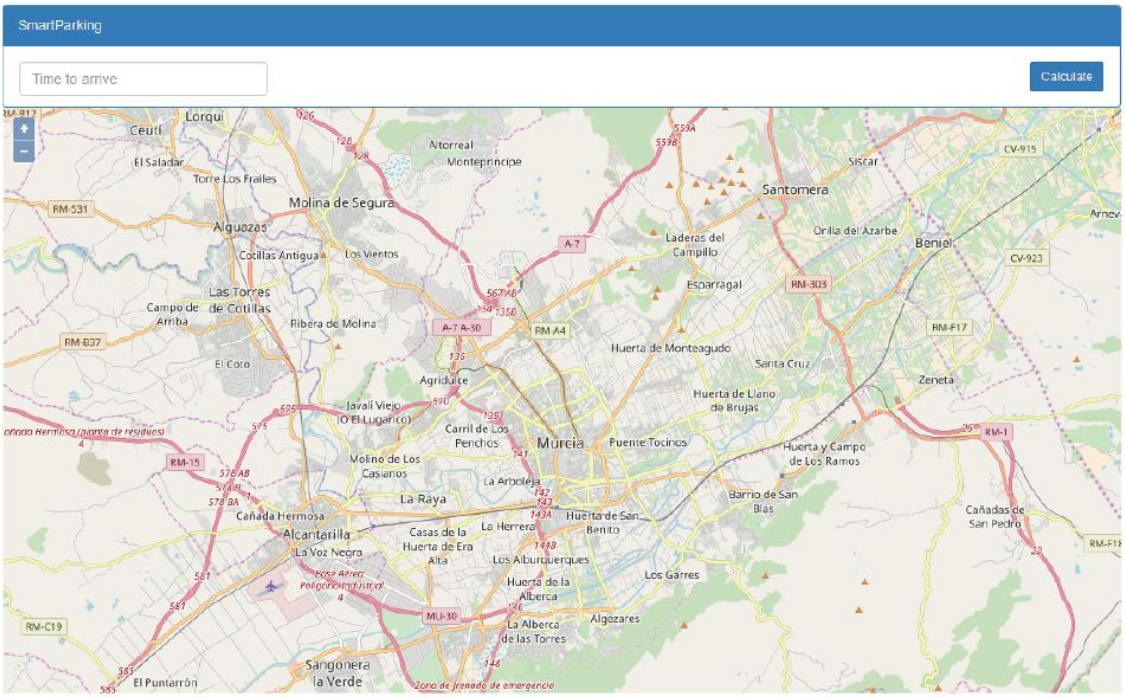


Figure 4. Smart Parking map-based GUI

3.2 Architecture

The Smart Parking solution provided by our Fed4IoT framework integrates the information coming from the FIWARE-based Mi-Murcia platform. Figure 5 presents the most relevant component of the envisioned platform according to this concrete use case.

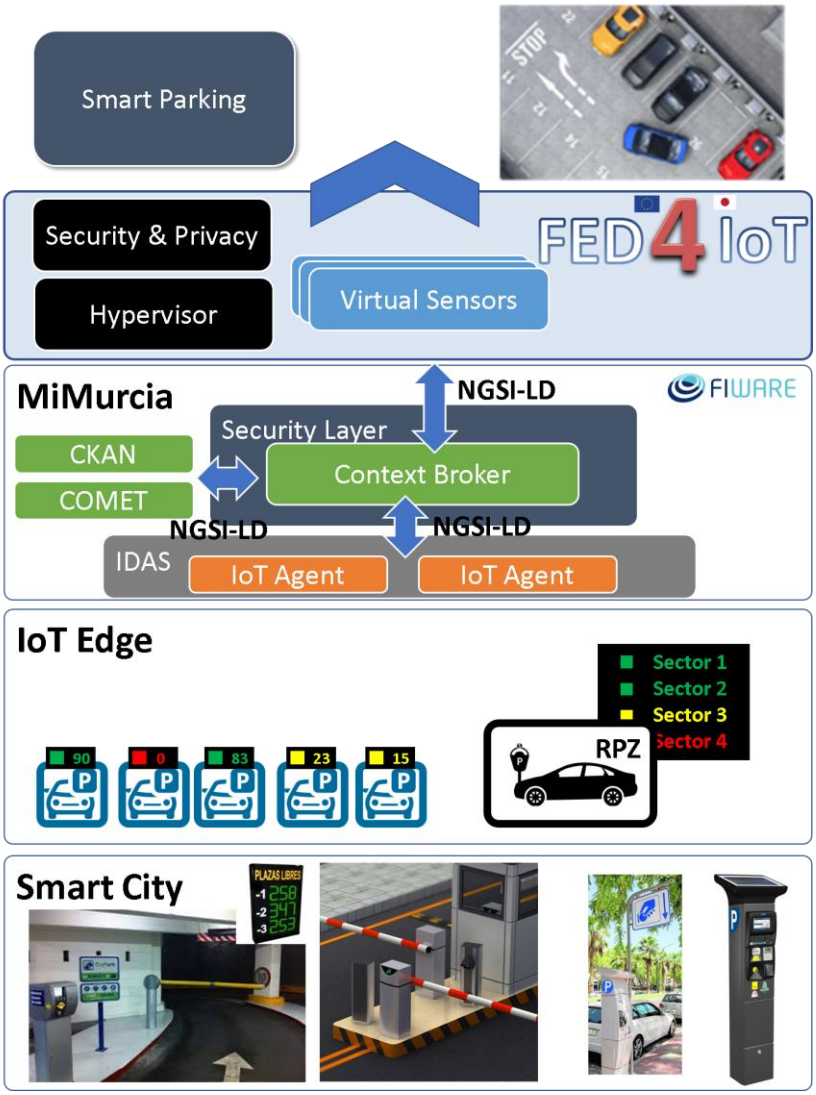


Figure 5. Smart Parking architecture

Our Smart Parking use case requires two kinds of context sources: the availability of private parking sites in terms of free parking spots, and the information coming from the Regulated Parking Zone (RPZ) regarding the daily expended tickets. This information is provided thanks to the sensors deployed in each private parking site that decrease the number of free parking spots as a vehicle enters the parking site and increase it when a vehicle exits it. Usually, an IoT gateway is required to transmit this information to an IoT platform too. For the case of RPZ, the expenders provide two different operational modes depending on their age. Old-fashion expenders are equipped with highly-constrained CPU, so during the day, they are completely dedicated to the ticket issuance task. These devices take advantage of the nights to perform the transmission of the activity of the whole day providing detailed information regarding the expended tickets. On the other hand, the modern ones are equipped with a more powerful CPU which allows for a real-time exchange of information. Therefore, when the expenders issue a new ticket, it communicates this fact to its corresponding platform.

Considering the previous information, the integration with a city platform requires a second iteration. This one is focused on the information provided by both IoT platforms (private parking site and RPZ) and how it is integrated into the city platform. Since we are integrating heterogeneous information from two different IoT platforms, we need open and standard protocols to unify this information into a single repository and platform, for this reason the adoption of NGSI-LD is a key aspect.

A more generic Smart Parking use case processing information from different cities or countries will raise the previous requirements to a distributed or even federated environment where each source of information can be provided by a different instantiation of our IoT city platform. This is one of the reasons for integrating this information into our Fed4IoT framework. In addition, this framework will also allow us to provide a virtualized view of the current sensors. This approach lets us define subsets of sensors, virtual slicing which can be assigned to different clients or tenants depending on their application (as for public cloud). For instance, for the specific case of smart parking the Fed4IoT provides the power to define virtualized data for the availability of private parking sites. Regarding the RPZ, an enhanced representation can be produced: providing the whole battery of tickets, the tickets associated only to a specific region, an aggregated value of it, etc.

3.3 Data Sets

Since this use case is new, in this section we detail the different involved datasets. The application needs the information provided by the concessionary companies responsible for the management of private parking sites and the regulated parking zones (RPZs). This information is therefore divided in two groups.

On the one hand, the information regarding private parking sites:

- Id: An identifier of the parking site
- Type: The current value for these entities is *Sensor*
- Attributes:
    - nombre: The name of the parking site
    - geoposicion: Geographical position of the parking site
    - forzado: This attribute can be set to establish several free parking spots publicly available.
    - libres: The number of free parking spots.
    - totales: The number of total parking spots.

On the other hand, regarding the RPZs information we receive in our platform the information about the tickets issued the day before. Such information comprises:

- id: identifier of the ticket, e.g. Ticket:123456789.

- type: for these entities the value is set to Ticket.
- Attributes:
    - duracion: duration in seconds.
    - expendedor: expender
    - fechaemision: issued date.
    - horaemision: issued time.
    - importe: price
    - pago: Payment option (e.g. Efectivo = Cash)
    - tarifa: rate
    - validez: booked period of time e.g.12:09 – 12:54
    - zona: Sector or area.

3.4 Deployment View and Interoperability Aspect

The previous picture of the architecture, Figure 5, also sheds some light onto a deployment view, highlighting interoperability aspects too.

In that architecture diagram we included not only logical components but also the physical ones, i.e. sensors, gateways, as well as software components encompassing the different enablers for the use case. We can highlight the following ones:

- IoT sensors deployed in each Private Parking sites that will transmit in real-time their occupancy state. Usually this process is carried out by decreasing the number of free parking spots each time a new vehicle enters the parking site and increasing it each time a vehicle leaves it.
- IoT gateways to forward readings at the entrance and exit of the parking site to an IoT platform.
- Regulated Parking Zone expenders which issue tickets valid for a specific period of time.
- Integration of this information into its corresponding RPZ IoT Platform.
- IoT platform which usually exposes an API to allow the interoperability with third-party platforms.
- IoT Agents to tailor this information, acting as intermediaries, to integrate this information into the third-party platforms.

In addition, from a functional point of view we can have different approaches to provide a Smart Parking solution. The more simplistic one could be to provide all the information to a cloud service which does all the performance to provide a result. Nevertheless, such approach provides a specific solution applicable to only one scenario because the cloud service would expect the information represented in a determined manner (provided by the sensors or IoT platform) and it will not be possible to export or migrate to another city.

Since interoperability and federation are, among other characteristics, the focus of this project, we must follow a not so straightforward approach. To start with, we must agree on a common representation notation. This way, despite having different sensors, actuators or sources of information, all of them will be represented following the same structure. For this reason, we

have selected NGSI-LD, since it is an open and standard information representation which can also include references to other represented entities thanks to its linked-data (LD) feature.

Additionally, for a federation scheme we take advantage of the virtualization concept but applied to the sensors that can feed our Fed4IoT platform. Thanks to this virtual sensor, we can create slices over them which are provided to the application depending on their needs. Thanks to this approach, the smart parking application can consume the availability information of the private parking site, but only when it is relevant for its processing, for instance when a certain number of vehicles have entered or left. In addition, the same concept is applied for the information coming from the RPZ, where information over a concrete area or region is provided, instead of the raw values of the tickets, or the totally available information.

Thanks to this approach, we must count with the components that were also depicted in Figure 5.

- NGSI-LD IoT Platform: This platform is fundamental to integrate the information into our federated scheme. Otherwise, a connector/adaptor is required to integrate the information.
- A hypervisor which is able to provide virtual entities/sensors based on the information integrated by the NGSI-LD IoT Platform. This component will define these definitions, and they will also be able to provide an end-point for applications to consume the aforementioned information.
- Information broker: A broker is required to store the information and to provide an API to access to it.
- Interfaces: NGSI-LD is selected as the most appropriate interface for providing the information in a homogeneous and unified way. It also let us to provide semantic annotation to the stored information.

3.5 FG-DPM Unified Use Case Template

FG-DPM D1.1 unified use case template		
Use case title	Name	Smart Parking
	Domain – Cross domain	Smart City – traffic
	Version	0.1
	Source	Murcia Smart City Platform (Mi Murcia)
Objective	This scenario exploits the needs for a more environmentally friendly traffic management by integrating the information about different sort of parking areas (private parking lots and regulated parking zones) into a service that will allow the users to reduce the time spent for a free parking spot in a specific destination.	
Background	Current practice	Murcia is a medium-size city in the south east of Spain with a population of 450.000 residents. This number increases by about 5.000 residents per year,

		with a noteworthy distribution of people among the city districts. Because of this and the fact that Murcia is the capital of the Murcia Region, it has been detected a dramatic rise of accesses to the city center in the last years. Day by day, commuters, tourists and families traveling by car collapse the core of Murcia, wanting to park at commerce, financial and historical areas.
	Rational for the use case	To improve the situation, the aim of this use case is that of providing a service that tracks the state of the parking spots to provide the drivers with this information beforehand, so they can better plan where they will park. This will also lead to a more fluid traffic in the city center, because there will be less drivers wandering, looking for a parking spot in areas that do not have any available one.
Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	Main stakeholder: City Council and concessionary companies responsible for managing regulated parking zones (RPZs from here on) and private parking sites. Responsibility: reduce the time spent by the citizens searching for a free parking spot. Secondary stakeholders: Sensor owners. Responsibility: making devices discoverable.
Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	A driver wants to attend a meeting or do some shopping in the city center. He/she wants to park as close as possible to the intended destination, and having no other alternative, plans on using private parking or RPZs. The driver has a cell phone with internet connectivity, and access to the Smart Parking app. The driver also knows the destination's location and the expected time of arrival. By entering the desired destination and time of arrival, the Smart Parking application will emit a parking recommendation, based on the information provided by the user and estimations of free parking spots. In order to emit such a recommendation, the Smart Parking system uses real time information and estimations of free spots in private parking, as well as an approximate estimation of free spots on parking regulated zones, both based on information provided by the companies/organizations responsible for those services.
Detailed scenarios	(same structure than "Scenario")	
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics ○ Data accessibility ○ Data availability	All the information exchanged in this system follows the NGSI-LD standard, by which the format and structure of data and metadata exchange is established in the form of context entities which hold information in the form of attributes. As such, common metadata regarding time of update of context entities and attributes is readily available, as well as typing information and other metadata. User inputs consists of individual requests for parking allocation, whereas the parking availability and occupancy can be introduced as an aggregation with

	○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability Data format, incl. standard, structured	varying time scales depending on the source (from some minutes to full days). Intermediate data, consisting of predictions and forecasts for free parking spots in the different places integrated in the system, is generated on a timely basis by forecasting software agents, based on historical and current parking availability information existing on the system. This information is represented in the form of forecasts for different time scales (depending on the source) and horizons (for example, on a 5minute basis for the next 5 days). Output data consists of recommendations for parking sites in which free parking spots could be found given the user's premises. This information is available through NGSI-LD queries and is temporarily stored in the IoT Broker. Once again, metadata regarding the user temporal ID that triggered the recommendation, as well as user's given premises and the time of recommendation is also stored in the context entity.
DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl.	Historic and current parking occupancy information is transferred from RPZ concessionary companies and private parking sites to the system. Parking occupancy information is processed and analyzed, generating forecasts of parking availability in different time scales and time horizons. Users generate parking recommendation requests in the form of entities in the system. Users request trigger the computation of parking recommendations, which make use of parking forecasts and current status. Parking recommendations, once created, trigger the response to the user. Parking recommendations and user requests can be further analyzed to compute KPIs regarding system use and performance. All information on the system is distributed in IoT Brokers along several nodes. Centralized index of the available information stored along the IoT Brokers is stored in the IoT Discovery component. Computations in the system, take place in nodes, which can be edge physical components or cloud elements. Other elements external to the Smart Parking system, can be connected (thanks to the NGSI interface) to further enhance and extend its performance and capabilities, providing components for OpenData sharing and BigData analysis among others. The inherently distributed nature of the system makes it resilient and flexible, adapting to changes in structure, hardware availability and user demand.

	5Vs of Big Data) ○ Data models and modelling ○ Data backup, archiving and recovery ○ Event management ○ System resilience ○ System sustainability	
	Data application to the different interests, incl. stakeholders' interests	
Governance and data life cycle considerations	Data accountability	
	Data isolation	
	Personal data (incl. sensitive personal data)	
	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	
	SLAs enforcement	RPZ concessionary companies and private parking sites, may enforce the use of SLA agreements regarding the manipulation and use of their data.
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	The Smart Parking service is a non-critical public service, with no private or sensible information stored or processed. As such, the main risks that concern us are the enforcement of SLAs regarding parking historical and current data, and keeping the availability of the service as high as possible. To that extent, cybersecurity risks will be faced by implementing secure communications and enforcing the use of secure certificates and PKIs. Availability of service will be assured by applying HA techniques and keeping track of and forecasting system use, to implement scalability policies according to demand.
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main	RPZ concessionary companies and private parking sites, should consent to give access to their current and historical data regarding parking spot occupancy for forecasting and current state of parking availability use. Personal in charge of and allowed to access and manage that information will exclusively be technical staff in charge of the management and support of the system.

	groups of internal employees, (2) external business partners, (3) general public)	
	Data value chain maintenance, incl. data asset management (data asset value appraisal, identification, registration and disposition)	Once user request and parking recommendation data has been used (by issuing user response), it will remain stored in the system for as long as it's needed for analytical computations and/or its collection by OpenData external components. After that period, data may be safely discarded from the Smart Parking system, which should only store the snapshot of data required to perform current and new recommendations. This process is performed by automatic tasks triggered at specified intervals.
	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	The system issues parking recommendations based on user requests in which desired destination and time of arrival are specified. Parking recommendations take into consideration destination, time of arrival and parking availability based on current status and forecasts. Forecasting is computed based on current and historical information of PRZs and private parking sites.
	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	System should be available and reliable, and perform its recommendations with enough agility as to avoid users desisting from their request and making for an enjoyable experience. System should be interoperable with other smart city modules and components.
	Other requirements	
	Available International Standards supporting the requirements (if any)	
	References (related to above standards or other useful information (e.g. on regulatory aspects))	
Architecture considerations	○ Communication infrastructure (incl. connectivity) ○ Data consistency across systems involved in the use case ○ Deployment	The distributed nature of the Smart Parking system, call for a wide range of alternatives regarding communication technologies applied in edge nodes. Although there is no intrinsic need for high data bandwidths, a reliable and performing connection is required for the successful operation of the system. Public internet addresses are needed for all the systems involved, as task management and context data subscription and notifications require so.

	considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	
General remarks	The impact on society can be summarized in the next advances: reduce the time spent looking for a free parking spot; reduce the CO2 emissions associated to the wandering vehicles; improve the traffic management because of the reduction of the time in pursuing a free parking spot	

4 Use Case – Cross Border Person Finder

4.1 Application and Use Case Description

This application aims 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, represented, for testing purposes, by our cities.

Real cameras offered by the Fed4IoT federated infrastructure in the involved cities will be used for **creating a cross-border (for instance between two different EU cities) IoT slice of virtual cameras**. This means that a single slice dedicated to the person finding application can leverage virtual cameras from both the cities of Grasse and Murcia, as an example.

More importantly, the **same real cameras can, at the same time, be exploited for another, totally different application, within a different IoT slice**. So that, for instance, the same real cameras are used to support both the Cross-Border Person Finder and the Waste Management (see next use case) applications, and they are virtualized differently within different slices.

The related streams will be offered at the Fed4IoT contextual information sharing layer and then processed by specific algorithms on the edge computing resources within Fed4IoT, which will be selected by application-specific criteria.

Person detection algorithms will carry out matching operations using photos stored in a Person DB operated by an Identity document issuing authority. The identity document issuing authority assures the integrity of photo data and credentials of the users.

A subset of photos from the Person DB is 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**), if the person is not to be found in the original city after a specified amount of time. Photos are going to be matched with streams coming from the right virtual camera.

End users can enter images of the person to be searched and be notified upon relevant matches.

4.2 Architecture

The cross-border person finder system consists of the following components.

i. **Surveillance camera system**
   The surveillance camera system consists of cameras distributed in the smart city. Each camera supports APIs for face detection and face recognition.

The surveillance camera system supports functions to publish data and also provides APIs for other service providers to use their surveillance camera system, based on subscription requests. The surveillance camera system may also support functions to identify the location (e.g., GPS) of the camera.

ii. **IoT Slice**

It is the virtual environment hosting the virtual cameras and the data broker (e.g. oneM2M or FiWARE) for this specific application, assigned to the application owner/tenant.

iii. **Edge application for Person Recognition**

The IoT Slice supports execution of all parts of the application that are needed for the cross-border person finder system, running at the edge of the network infrastructure.

The application is connected to the Person DB operated by the Identity document issuing authority, and to the Front-End Service and user devices.

The application should support optimized relocation of photos, to be used for cross border person finder, according to explicit user consent.

iv. **Front End Service and user devices**

End user devices are used to access to cross border person finder service through a front end. End user devices should have an instance of identity document of the user to be identified. Such instance of identity document is used to authorize the service.

End user may be asked to verify the service access privilege by the authority.

v. **Person DB**

Person DB supports a function to store the photos and credentials (including privilege and roles) of users. Person DB is operated by identity document issuing authority and the stored data should be derived from primary Identity documents such as ePassports, eIDs and e-driving licences.

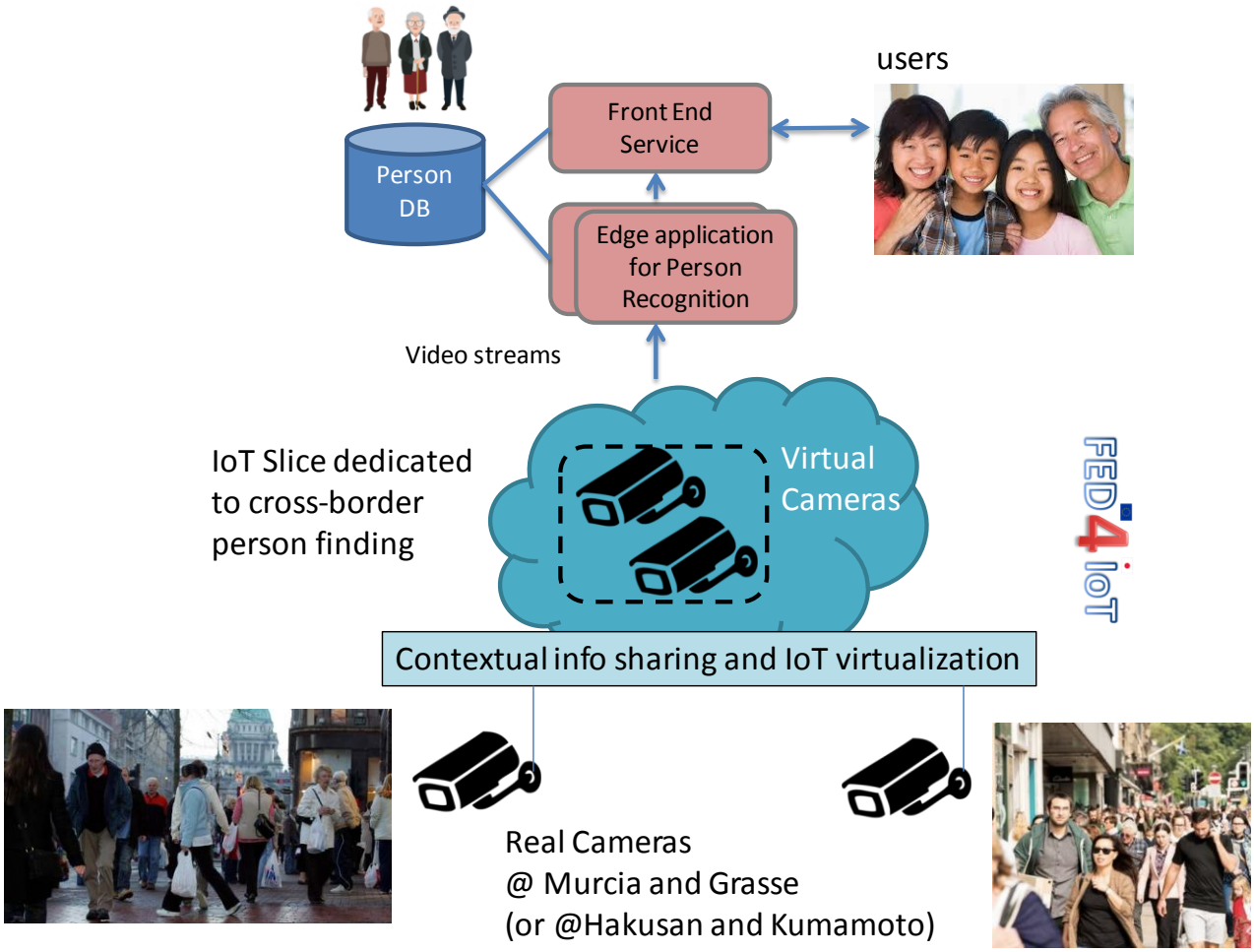


Figure 6. Overview of the Cross Border Person Finder System

4.3 Data Sets

The following data sets are to be produce and consumed in IoT part of this system.

i. **Person finding requests**
   Produced by the front end and sent to surveillance camera system via IoT Slice.
   The data set is described as JSON-RPC request that should include consumer information, a list of service requests and authorization for service requests

ii. **Person finding response**
   Produced by surveillance camera system and sent to front end via IoT Slice.
   The data set is described as JSON-RPC response that may include notification, feature of face image and image/movie of the target person. The JSON-RPC response may also include additional information for identifying location (e.g. GPS data) and time stamps.

4.4 Deployment View and Interoperability Aspect

Usually person finding systems are thought to be deployed within a limited area/city because surveillance camera systems are set by local governments. Fed4IoT enables to interwork such systems with other smart cities.

Thanks to the IoT slice concept, real cameras can be shared between very different applications. Thanks to the interoperability, provided by Fed4IoT concept of a federation of context brokers, the person finding system can be used throughout multiple smart cities. In application level, we also will use international standard based personal identification technologies and cryptographic technologies, thus this system will be interoperable across the different technologies.

4.5 FG-DPM Unified Use Case Template

FG-DPM D1.1 unified use case template complemented by field-specific guidelines		
Use case title	Name	**Cross border person finder**
	Domain – Cross domain	**Smart City – Cross domain**
	Version	**0.1**
	Source	**Panasonic Corporation, Waseda University**
Objective	**This application is aimed to notify 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.**	
Background	Current practice	Usually, a person finding system can be deployed within a limited area/city because surveillance camera systems are set by local governments.
	Rational for the use case	**This use case seeks to enable extending such systems throughout other smart cities in an interoperable way.**
Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	Surveillance camera system operator: Operating surveillance camera system in a smart city. They also provide APIs for other service providers to use their surveillance camera system. Person finder service provider: Providing person finder service across smart cities. They provide their services to the end user upon request. They require to identify the end user by identity document issued by authority and authorize the service according to explicit user consent. End user: A person who wishes to find a person remotely. He/she shall have a smart city app on a smartphone, and the smart city app is included the photo and credentials, including relation with a person to be found, derived from identity document issued by authority. Authority: An entity to issue and manage identity document and information. They have a database (Person DB) that stores the photos that is requested to be found by end users. They provide a photo and associated attributes to person finder service provider according to end user's consent.

Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	Real cameras offered by the Fed4IoT federated infrastructure in the involved cities will be used for creating a cross-border IoT slice of cameras. The related streams will be offered at the Fed4IoT contextual information sharing layer and then processed by specific algorithms on the edge computing resources, which will be selected by application-specific criteria. 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 subset of photos stored in the Person DB is 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 relevant matches.
Detailed scenarios	(same structure than "Scenario")	
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics ○ Data accessibility ○ Data availability ○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability Data format, incl. standard, structured	i. Person finding requests Produced by front end and sent to surveillance camera system via IoT Slice. The data set is described as JSON-RPC request that should include consumer information, a list of service requests and authorization for service requests Each data item is based on credentials/attributes and some set of data items that could be gathered as "data group". The sender of the message should be verified to confirm the service authorisation. ii. Person finding response Produced by surveillance camera system and sent to front end via IoT Slice. The data set is described as JSON-RPC response that may include notification, feature of face image and images/videos of the target person. The JSON-RPC response may also include additional information for identifying geolocation (e.g. GPS data) and time stamps. The data shall be signed by the IoT device to authenticate that the data is created by the authorised IoT device, and also ensure the data integrity.

DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl. 5Vs of Big Data) ○ Data models and modelling ○ Data backup, archiving and recovery ○ Event management ○ System resilience ○ System sustainability	In data processing capabilities, for handling the user requests, a person finding system requires to authenticate the users' requests by certifying the users' IDs to protect the data from illegal access and illegal data usage. To certify the users' IDs, the system requires a capability of accessing the data base stored the face images, matching the face images, and transmitting to the matching results to the users' devices. In addition, for processing the images/videos, the system requires to capture the images/videos from city surveillance cameras, collects the images/videos in the edge and cloud servers, and performs image processing to detect the humans, including their faces, and extract their attributes. In addition, the system requires to connect the human information to other environmental information, such as geolocations and time stamps in order to track a human's historical movement. In data management capabilities, to protect the privacy for the users, the system requires to encrypt every data exchange between the servers and the users. In addition, to reduce the video traffic volume and server processing loads, the captured images/videos are processed and stored to distributed regional edge servers. These images/videos should not be stored as raw data but intermediate processed results, such as extracted feature points.
	Data application to the different interests, incl. stakeholders' interests	
Governance	Data accountability	Integrity of system should be assured.

and data life cycle considerations	Data isolation	The person finding requests and responses should be isolated among users.
	Personal data (incl. sensitive personal data)	All collected users' face images should not be stored as raw data but just as intermediate processing results. In addition, the users' information should be non-linkable to users' identity.
	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	Person finding system should be composed the license free applications because the system is provided by the municipalities. All collecting/processing data should be private.
	SLAs enforcement	
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	The users and data base of person finding system should be protected from illegal access and illegal data usage.
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main groups of internal employees, (2) external business partners, (3) general public)	Data access rights and data authorization considerations according to the different stakeholders: in general, the system providers have the access to all details of data processing results, while the users only have the access to the authorized data by the system providers.
	Data value chain maintenance, incl. data asset management (data asset value appraisal, identification, registration and disposition)	
	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	Edge and cloud processing, streaming data, and visualization the person finding responses

	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	
	Other requirements	
	Available International Standards supporting the requirements (if any)	oneM2M
	References (related to above standards or other useful information (e.g. on regulatory aspects))	WiFi alliance, OpenID Connect
Architecture considerations	○ Communication infrastructure ○ Data consistency across systems involved in the use case ○ Deployment considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	The captured images/videos by the surveillance cameras are mainly transmitted via Wi-Fi and gigabit Ethernet. To acquire the images/videos from cross-border smart cities, surveillance cameras should be virtually shared and equipped with common APIs.
General remarks		

5 Use Case – Waste Management

Waste management, including the separation, collection and logistics of daily wastes in different garbage sites across the city, has become an important aspect of urban life. Effective and efficient waste management strategies are able to largely improve living experiences, protect environment, and promote garbage reuse.

With the Grasse Smart City project, EGM is managing the whole network deployment and supporting the deployment and adoption of the waste management use case. The Grasse Smart City project is a regional collaborative project with local authorities and associations as partners. It aims to provide more digital facilities and applications to the citizens to make life greener and more efficient using state-of-the-art IoT technologies. The main interest of the public authority managers is to understand the way IoT technologies can benefit to citizens in urban, peri-urban and rural areas and identify the sustainability model of such deployments at a time of reduced budgets and increasing constraints on data management (such as GDPR or open-data regulations).

5.1 Application and Use Case Description

The waste management use case will deploy smart camera in the waste collection sites to detect uncivil behaviours such as throwing bulky objects outside garbage bins or putting in the normal garbage bin dangerous objects or objects harmful to the environment such as batteries. Through the automatic analysis and detection, the objective is to detect on time the uncivil behaviours and send related notifications to raise awareness for follow-up processing. The rapid reaction to uncivil behaviours improves the urban life experiences by preventing the continuous bad consequences.

The different actors are identified with the following roles related to the use case.

- Data provider: smart cameras. Smart cameras will be deployed over different garbage sites to frequently collect images and/or video streams of the area with garbage bins, and then send the data to be processed by data processor.
- Data processor: fed4iot edge agent/software component. The collected data will be processed in the edge side to identify the uncivil behaviours. Generally, the data are processed by data processor associated with the camera in the gateway level, while only the processing results (the detected uncivil behaviours) are sent back to Fed4IoT central platform for further use, while the collected raw data stay locally anonymous.
- Data user: city authorities, garbage management company, citizens. The processing results about uncivil behaviours are available via the Fed4IoT platform interfaces for query and subscription/notification. The relevant users with authorization can access the data, and the city

authorities or garbage management companies can define the follow-up actions to deal with the uncivil behaviour consequences.

Following the list of actors in the use case, we specify here two main scenarios including the high-level interactions between actors.

Scenario 1. Data Collection and Information Sharing.

- Step 1. The data providers collect data from garbage sites across the city.
- Step 2. The data processors conduct data processing and send results to Fed4IoT platform.
- Step 3. The data users (e.g., authorities and companies) query the Fed4IoT platform and get related information follow-up actions.

Scenario 2. Uncivil Behaviours Improvement.

- Step 1. The data users (e.g., citizens) query the Fed4IoT platform and get related information of detected uncivil behaviours related to each garbage site.
- Step 2. The environment protection awareness is raised due to the detection results and the citizens autonomously and continuously improve their behaviours.

5.2 Architecture

The preliminary architecture of the use case deployment in Grasse is presented in Figure 7. In edge level, an IoT slice will be used to setup a virtual camera infrastructure and to deploy related analytics on edge computing resources. The slicing technology allows the same real camera to be used for different applications (e.g. for cross-border person finding and for waste management), by deploying different smart analytics on different slices. As regards the shape, colour, volume, etc., the virtual camera and its analytics are able to recognize trash thrown outside of the garbage bin, and possibly a non-matching of the type of trash to the type of garbage bin will trigger an anomaly event to be transmitted to the Fed4IoT platform. The data users can query and retrieve different information for references and decision making, and the statistics of uncivil behaviours will also be made public to raise citizens' awareness. In particular, the collection of certain data can be sensible to citizens, the use case will focus on the edge processing of the data by the devices or local gateways, so that only results will be sent back to Fed4IoT platform, while the collected raw data (e.g., person images) will stay anonymous locally.

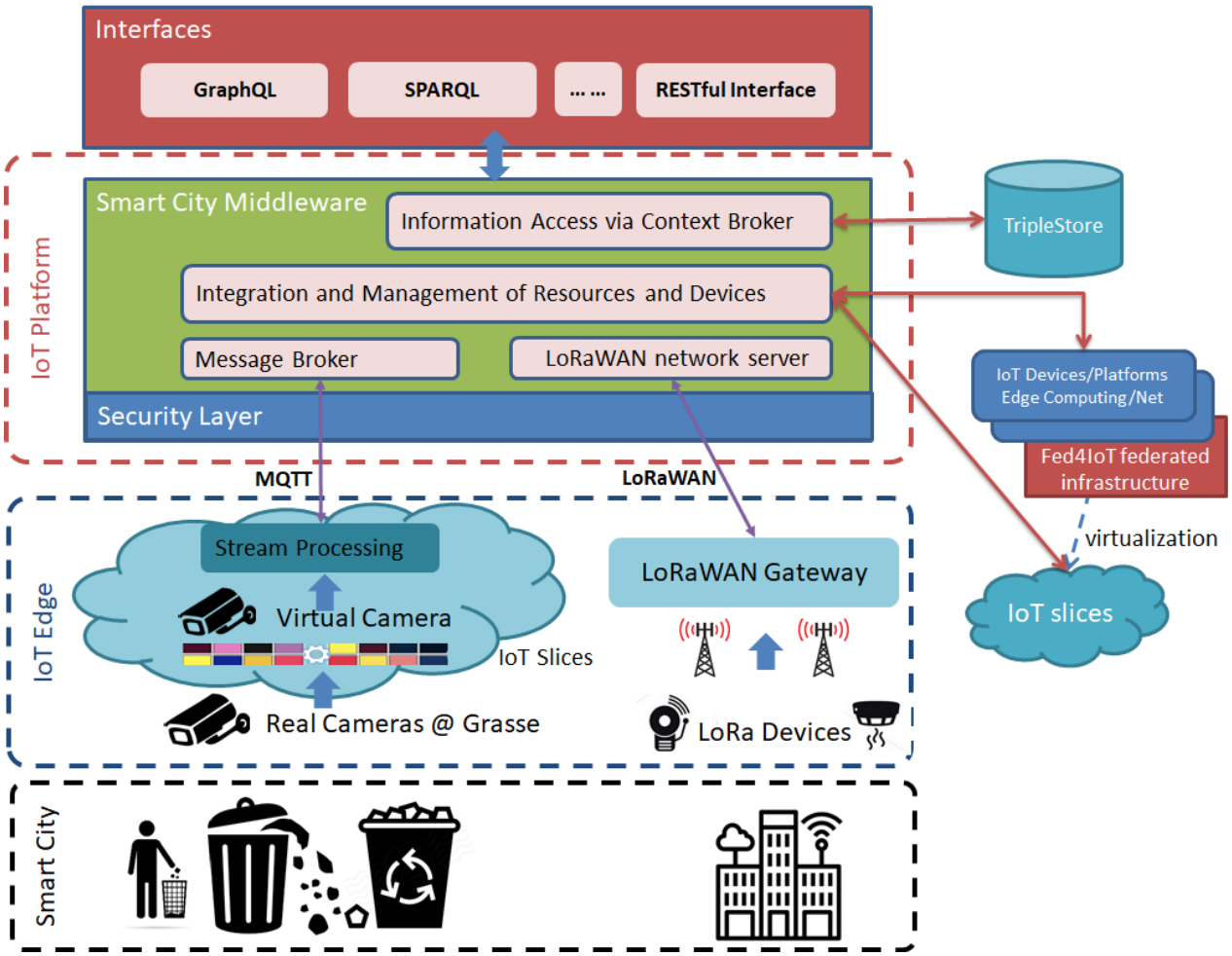


Figure 7. Waste Management Deployment Architecture

This use case will use all technologies developed in Fed4IoT projects with the following particular points:

IoT Virtualization: To achieve the smart multi-tenancy technologies and easy the use of IoT services, the IoT virtualization will form a federated pool of real IoT/cloud resources that can be used for providing IoT slices through virtualization services. The investigation and the application of this technology will be among VMs, containers and functions provided by IoT service platform such as oneM2M Application Entity.

Stream processing: The continuous collection of data and the privacy requirement of citizens require the data processing efficiency and anonymity, and thus the data will be processed locally via edge computing and stream processing technologies. The objective is to ensure the detection of relevant events and inform the IoT platform only the data processing results. The related components to achieve this function are Fogflow, Apache Flink, and so on.

Message broker: The message broker connects different components following the microservice paradigm and serves as an efficient data transfer service with multiple messaging protocols support in distributed and federated configurations.

Integration and Management of Resources and Devices: Besides the data communication, the deployment in Grasse will enable the automatic management of deployed devices (such as device provisioning and device diagnostics) and the management of IoT resources by use of the functions provided by the deployed IoT platforms, such as oneM2M, LightweightM2M.

Information Access via Context Broker: Upon the management and integration, an information access component will be deployed to provide users with efficient information access APIs and federate data from different sources. A target API is the NGSI-LD one, leveraging the triple store used to store the meta information (e.g., devices and resources) with semantic information to support advanced query and additional knowledge discovery.

Interfaces: In order to achieve the specified scenarios, the deployment will provide users with interfaces for data analytics, data visualization and device management. The potential interfaces include GraphQL, SPARQL, RESTful interfaces of NGSI-LD.

LoRa Deployment: Furthermore, besides the waste management use case described in this section, the deployment in Grasse will continue to explore other possibilities with interesting use cases to improve urban life experiences. One focus point will be the deployment of LoRa devices (for example, air quality sensors) to encourage the necessary actions of environment protection and extend the use case deployment in Grasse.

5.3 Data Sets

The data sets exchanged and used in this use case mainly contain four parts:

- The raw data collected by smart cameras for processing,
- General information of the deployed devices and the garbage sites
- The detected garbage site status and uncivil behaviours

Except the raw data of the camera, not planned to be streamed out of the edge, the other two parts of data will be consolidated in JSON format, additionally with semantic annotations for metadata. Here we present two examples of the exchanged data, while the details of data specification will be defined during the project period.

Figure 8 and Figure 9 respectively present the JSON-LD description examples of the information of a garbage site and a deployed camera. The JSON-LD combines the JSON description with a field “@context”, in which all concepts used in the JSON are mapped to a specific ontology concept with semantics to support data interoperability and advanced reasoning.

```
{
  "@context": "http://vocabulary.example.org",
  "id": "urn:example:Camera1",
  "type": "Camera",
  "location": {
    "type": "Property",
    "value": 15 Chemin des Gardes
   },
 "monitoringSite": {
    "type": "Relationship",
    "object": "urn:example:GarbageSite2"
  }
}
```

Figure 8. JSON-LD description for Device Information

```
{
  "@context": "http://vocabulary.example.org",
  "id": "urn:example:GarbageSite2",
  "type": "GarbageSite",
  "location": {
    "type": "Property",
    "value": 16 Chemin des Gardes
   },
"managedBy": {
    "type": "Relation",
    "object": "urn:example:Company1"
  } ,
 "status": {
    "type": "Property",
    "value": "empty"
  }
}
```

Figure 9. JSON-LD description for Garbage Site

5.4 Deployment view and Interoperability aspect

Smart cameras will be deployed mainly over different garbage sites across the cities, connected to a gateway for the communication of detection results to Fed4IoT platforms. Mainly taking

advantage of the edge technology from Fed4IoT project, the data processing will be achieved by leveraging the edge resources and functions.

The interoperability will be ensured in two main aspects:

Inter-deployment interoperability: the metadata of different devices and garbage sites along with the detection results will rely on the information model of Fed4IoT to achieve the information exchange between sites and information federation over Fed4IoT platform; moreover, the interoperability of deployments over different cities can also be realized by Fed4IoT platform interfaces upon the information sharing federated architecture.

Inter-use case interoperability: The deployed cameras for waste management can also be used to implement the functions of the person finder use case. This is achieved by the device virtualization technology of Fed4IoT to virtualize the same physical camera into several virtual cameras to support different use case functions. Ideally, by adjusting direction and zooming of the same physical camera, the camera can change their surveillance areas and focus points to eventually apply associated functions of waste management or person finder.

5.5 FG-DPM Unified Use Case Template

FG-DPM D1.1 unified use case template complemented by field-specific guidelines		
Use case title	Name:	Waste Management
	Domain – Cross domain	Smart City – Urban life
	Version: 1.0	1.0
	Source:	Easy Global Market
Objective	**The objective of this use case is to provide easy deployment of garbage site monitoring devices and services to detect uncivil behaviours for necessary actions and improve urban life experience**	
Background	Current practice	The Grasse Smart City project is a regional collaborative project with local authorities and associations as partners. It aims to provide more digital facilities and applications to the citizens to make life greener and more efficient using state-of-the-art IoT technologies. The main interest of the public authority managers is to understand the way IoT technologies can benefit to citizens in urban, peri-urban and rural areas and identify the sustainability model of such deployments at a time of reduced budgets and increasing constraints on data management (such as GDPR or open-data regulations). With the Grasse Smart City project, EGM is managing the whole network deployment and supporting the deployment and adoption of the waste management use case.
	Rational for the use case	Effective and efficient waste management strategies are able to largely improve living experiences, protect environment, and promote garbage reuse.

Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	The stakeholders are identified as: ● IoT device owners. They provide IoT devices to deploy. ● Technology providers, who provide edge processing, communication, virtualization, and management platforms to support the running of use case. ● Garbage site owners. ● Data users, who query and/or get notified the uncivil behaviours from platform. Technology providers deploy the provided IoT devices along with necessary technologies over garbage sites; the technologies will be used to realize the identified scenarios, and data users get results from the platform for further reactions.
Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	Contextual illustration: In order to better protect environment and improve urban life experiences, necessary devices and technologies are deployed to detect uncivil behaviours such as throwing bulky objects outside garbage bins or putting in the normal garbage bin dangerous objects or objects harmful to the environment such as batteries. Pre-requisite: smart cameras deployed over garbage sites with uncivil behaviour detection services. Pre-conditions: enough lighting conditions to support the data collection Trigger: the data will be continuously collected following a fixed frequency, and an event will be recorded once an uncivil behaviour is detected. Typical operational procedure: smart camera collects data and process the data locally; once an uncivil behaviour is detected, the results are sent to central platform and users to query. Post-conditions: data users can query the results and react to different behaviours Information exchange: the collected raw data of and the detected uncivil behaviours Considerations on publicity of results: the detections results can be made publicly available to raise citizen awareness for environment protection
Detailed scenarios	(same structure than "Scenario")	○ Step 1. The deployed devices collect data from garbage sites across the city. ○ Step 2. The configured technology conducts data processing and send results to Fed4IoT platform. ○ Step 3. The data users (e.g., authority and companies) query the Fed4IoT platform and get related information follow-up actions. ○ value-added services: the same physical camera can be virtualized to provide extra service of person finder use case such as to find a missing person in the city.
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics	Data input characteristics: continuous collected image and/or video streams Data output characteristics: only the detection results are accessible with the access control Data formats: the processed data are communicated in JSON with semantic annotation.

	○ Data accessibility ○ Data availability ○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability Data format, incl. standard, structured	
DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl. 5Vs of Big Data) ○ Data models and modelling ○ Data backup,	Data processing capabilities ○ Cleaning and filtering: the collected raw data are cleaned regularly if no pattern has been detected ○ Classification and indexing ○ De-identification, anonymization and pseudonymization: the collected raw data are always anonymous in the edge side ○ Transfer: only data processing results are sent to platform ○ Pre-processing and processing: the collected raw data are processed in the edge side ○ Reading and query: the processing results are shared via platform interfaces for query and subscription/notification Data management capabilities ○ Access and use: the processing results are shared via platform interfaces for query and subscription/notification ○ Administration: the platform will control the access rights to results ○ Acquisition and collection: the data are collected via smart cameras following a fixed frequency ○ Storage: the collected raw data are cleaned regularly if no pattern has been detected, and the storage is updated frequently Considerations on system capabilities ○ Functions and operations: device virtualization, image/video processing, edge computing ○ Data models and modelling: the data include semantic annotation for interoperability purpose. ○ Event management: an event will be recorded once an uncivil behaviour is detected

	archiving and recovery ○ Event management ○ System resilience ○ System sustainability	
	Data application to the different interests, incl. stakeholders' interests	Different stakeholders can call the different virtualized service provided by the same camera for different purpose.
Governance and data life cycle considerations	Data accountability	
	Data isolation	
	Personal data (incl. sensitive personal data)	All collected personal data stay in the edge side anonymous, the detected results contain only the behaviour description but do not associate them with persons.
	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	All collected data are private while the processing results are public.
	SLAs enforcement	
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	The collection of citizen data involves privacy issue, and the corresponding management solutions is to keep all collected data anonymous in the edge while only communicate the data processing results without any personal information.
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main groups of internal employees, (2) external business partners, (3) general public)	Data access rights and data authorization considerations according to the different stakeholders: generally, the city authorities have the access to all details of data processing results, while the general public has access of the global information from a statistical and geographical view.
	Data value chain maintenance, incl. data asset management (data asset value	

	appraisal, identification, registration and disposition)	
	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	Edge processing, device virtualization and stream processing
	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	Non-functional requirements, incl. ○ Interoperability: the data processing results are semantically annotated and interoperable with other deployment of the same use case or different use cases. ○ Safety: the data processing results is accessible via access control ○ Security and privacy: the personal data should always stay anonymous
	Other requirements	
	Available International Standards supporting the requirements (if any)	NGSI-LD model and API
	References (related to above standards or other useful information (e.g. on regulatory aspects))	
Architecture considerations	○ Communication infrastructure ○ Data consistency across systems involved in the use case ○ Deployment considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	○ Communication infrastructure: the communication is mainly achieved by LoRa network ○ Data consistency across systems involved in the use case ○ Deployment considerations: deployment of cameras will be made mainly over garbage sites, while in the meantime considering the possible coverage of preserved city areas when adjust directions and zooming to realize other functions such as person finder. ○ Interface requirements, incl. user interfaces and APIs: query interfaces including SPARQL, REST, GRAPHQL, etc. ○ Performance criteria: near real time data processing and information sharing

General remarks	Effective and efficient waste management strategies are able to largely improve living experiences, protect environment, and promote garbage reuse.

6 Use Case – Citizen-Made IoT Applications

6.1 Application and Use Case Description

As IoT becomes widespread, people will be surrounded by many types of IoT devices with variety of functions. For example, an owner of a house selects, purchases, and places variety of IoT devices in and around the house. The combination of IoT devices used in an IoT system has wide variety from a system to a system. Even in such a diverse environment, IoT devices must cooperate to make a good use of the system and to deliver IoT services required by the users.

How IoT devices cooperate and collaborate varies from a system to a system due to the combination of IoT devices, the environment where the system is installed, and the service demand by users. Since the potential combination of IoT devices is limitless and the environment varies, the applications can have infinite variety and configurations.

Programmability of the system is a solution to counter the variability of applications. Basically, an IoT system is different from the others due to the variety in combination of IoT devices and as stated above. Each IoT system calls for a different program from another IoT system even to provide a similar IoT service. To this end, "user" programmability is required to solve the problem. If provided, user programmability can act as a catalyst to proliferate IoT systems in daily life of ordinary people.

The objective of this use case is to make the IoT systems programmable so that the owners and the users of IoT systems can integrate different IoT devices and let them work together to implement variety of services in a specific environment.

The main users to be concerned in this use case are the owners and users of IoT systems with no prior experience of computer programming.

The actors of this use case are:

- the owner of an IoT system with variety of IoT devices,
- users of the IoT system,
- IoT device suppliers,
- IoT platform provider, and
- software developers

The roles involved in this use case are:

- component function suppliers,
- IoT application programmers,
- IoT device wrapper suppliers, and
- IoT application users

"Component function suppliers" and "IoT device wrapper suppliers" are software developers. IoT devices are virtualized through "IoT device wrapper" software (aka IoT Hypervisor). The virtualization creates replicas of one physical IoT device to isolate IoT Slices serving different IoT applications as well as adding some functionality to the raw physical interface of the IoT device. The owner and users of the IoT system act as "IoT application programmers" and create IoT applications. They create an IoT application program by simply connecting and configuring function components supplied by "component function suppliers" and "IoT device wrapper suppliers" using GUI. Typically, IoT device suppliers provide IoT device wrapper for their products and act as "IoT device wrapper suppliers." Some software developers act as the "component function suppliers" and develop general purpose component functions to be used in this use case. Owners and users of an IoT system act as "IoT application users."

Here, function components are software that implement a certain processing of input data and generate some data. The input data may come from IoT devices and output data may be fed into IoT devices to control the devices.

6.2 Architecture

The architecture is made of the following components (see Figure 10):

- IoT device wrappers that virtualize physical IoT devices,
- an IoT network,
- computing resources that can be either in a cloud or in the network edge,
- Functions that perform processing of data from IoT device wrappers and produce output to be passed to other Functions or to IoT device wrappers,
- Routing function to route data to/from IoT devices and functions,
- Coordinator that deploy functions in computing resources and route data among Functions and IoT device drivers by means of Routing function
- Programming facility to interface between the Coordinator and programmers by providing GUI to the programmers, interpreting the programs, and transferring appropriate control information to the Coordinator

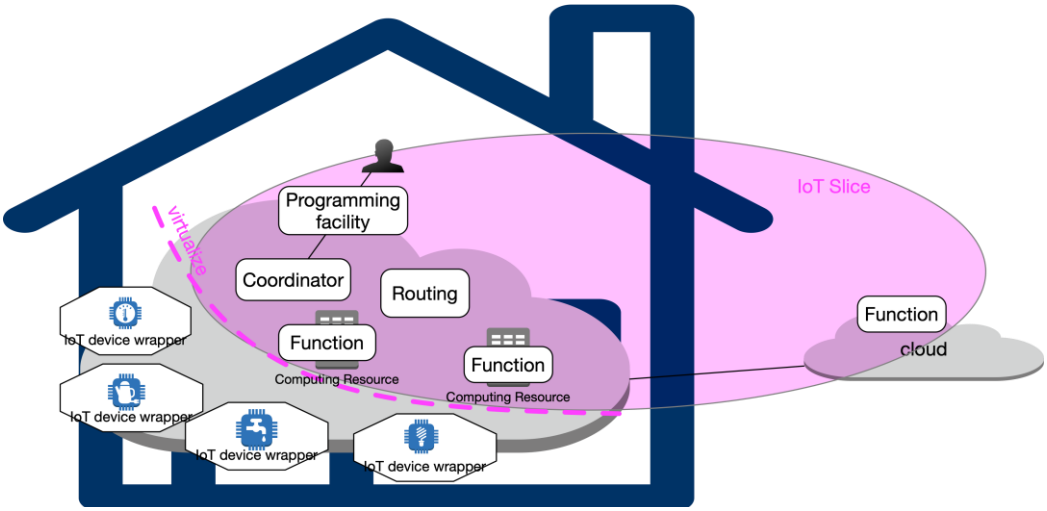


Figure 10. Architecture of Citizen-made IoT Applications

6.3 Data Sets

Data involved in this use case is:

- semantic information of interfaces of IoT device drivers and Functions to be referenced during program composition
- program itself including
    - information of IoT device drivers and Functions to be used in the program
    - information about data flow among IoT device drivers and Functions
    - parameter settings of IoT device drivers and Functions

6.4 Deployment View and Interoperability Aspect

A house owner constructs an IoT application that fit well with his/her needs. Component function suppliers develop Functions as the components of the house IoT systems and trade the Functions that can be used in different IoT platforms. The house owner purchases the Functions from component function suppliers, install them in his/her IoT system, and create his/her IoT applications by making use of the Functions.

The citizen-made IoT applications are not only used within a house but can be applicable in other application domains. Local municipality may let the citizens access the IoT infrastructure to invent a creative usage of their IoT system. For example, the city of Kumamoto, Japan, is building an event hall called "Kumamoto-Jo Hall" next to Kumamoto Castle (Kumamoto-Jo) and the hall is planned to be equipped with IoT devices. Kumamoto city plans to open up the IoT services to citizens to develop their applications.

The operability of the Functions in different IoT platforms may be realized by executing Functions in the platform where the Functions can run and communicating the input and out data of the Functions exploiting the facility provided by Fed4IoT (please refer to Figure 9).

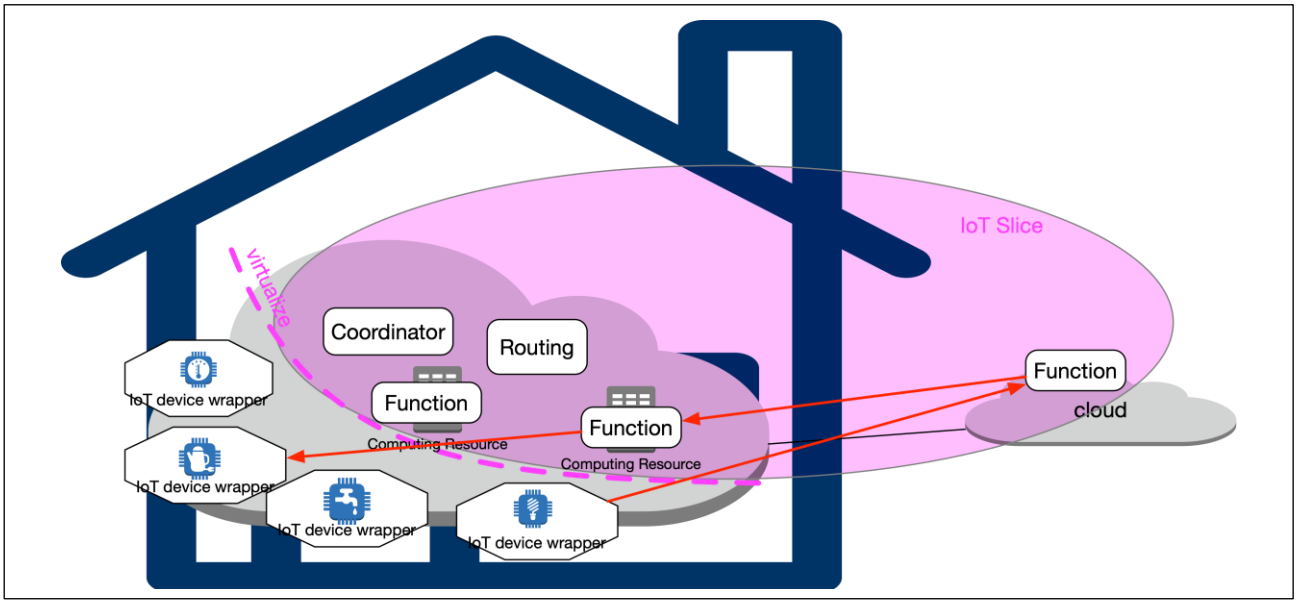


Figure 11. Interoperable Functions

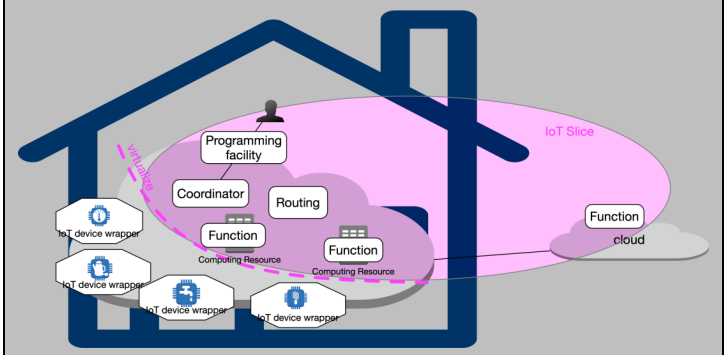
6.5 FG-DPM Unified Use case Template

FG-DPM D1.1 unified use case template complemented by field-specific guidelines		
Use case title	Name	**Citizen-made IoT Applications**
	Domain – Cross domain	**Home and Regulated Smart City Environment**
	Version	**0.1**
	Source	**Kumamoto-jo Hall**
Objective	**The objective of this use case is to make the IoT systems programmable so that the owners and the users of IoT systems can integrate different IoT devices and let them work together to implement variety of services.**	
Background	Current practice	The city of Kumamoto, Japan, is building an event hall called "Kumamoto-Jo Hall" next to Kumamoto Castle (Kumamoto-Jo) expected to open in 2019 and the hall is planned to be equipped with IoT devices. The IoT infrastructure to be federated is not yet finalized; the project will likely reuse the deployment strategy of FiWARE over oneM2M, and then the infrastructure will be federated in the Fed4IoT platform. The final deployment strategy will be decided during the project lifetime.
	Rational for the use case	Kumamoto city plans to open up the Fed4IoT services to citizens to

		develop their applications in addition to the ordinary usage such as guiding visitors and surveillance. The idea for citizens to develop their own application is part of the activities to re-vitalize the city by attracting more international visitors as well as the visitors from other part of Japan. In order to ensure the accessibility by citizens to the system, the city plans to develop IoT application components to be combined to form citizen-made IoT applications. Fed4IoT platform provides IoT device sharing capability among citizens to the system.
Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	The following stakeholders exist in the system. ● Owners and uses of an IoT system with variety of IoT devices. They may act as programmers of IoT applications as well as IoT application users. ● IoT device suppliers. They may also provide IoT device wrappers to virtualize their IoT devices. ● An IoT platform provider who is responsible to the infrastructure on which an IoT system is constructed. ● Software developer who supplies Functions that compose programs. Programmers of an IoT service purchase Functions and compose IoT applications by connecting the Functions and IoT device wrappers.
Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	The owner, which is a local government, of an event hall equips the event hall with variety IoT devices. The IoT system is constructed in an IoT platform. The government decided to open the IoT system to its citizens to develop IoT applications to find good applications of the system. The government purchases software called Functions that perform processing on IoT data in addition to IoT devices with their device wrappers. The citizens sit at a terminal that gives GUI to create IoT application programs. They can dispatch the created program to the system and test its behaviour.
Detailed scenarios	(same structure than "Scenario")	
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics ○ Data accessibility ○ Data availability ○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability Data format, incl. standard, structured	Input data except IoT device readings comes from programmer input through GUI. The input data includes the specifications of ● the Functions and IoT device wrappers to be used in the created program, ● the parameters to be set at the Functions and IoT device wrappers, and ● the information flow among the Functions and IoT device drivers. The input data need to be stored in the system as a program. There will be no output data except the one from IoT devices.

DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl. 5Vs of Big Data) ○ Data models and modelling ○ Data backup, archiving and recovery ○ Event management ○ System resilience ○ System sustainability	The system has processing capability to execute the Functions and transfer data among Functions and IoT device wrappers. In addition to the processing capability, the system provides the capability to store programs, interpret the programs, place Functions, route data among Functions and IoT device wrappers, and interface with programmers.
	Data application to the different interests, incl. stakeholders' interests	Different stakeholders can create their IoT application that fit with their interest using the programming capability
Governance and data life cycle considerations	Data accountability	Integrity of programs should be assured
	Data isolation	Programs should be executed by themselves and isolated from each other by means of IoT slices though they are shared among users in the system.
	Personal data (incl. sensitive personal data)	Handling of personal data can be avoided by properly preparing IoT device handlers and Functions

	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	Citizen-made applications should be license free since the IoT system is provided by the municipality.
	SLAs enforcement	No SLA exists for citizen-made applications
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	The users of citizen-made IoT applications should be protected from hazardous behaviour of the applications due to malfunction or ill-behaving programs.
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main groups of internal employees, (2) external business partners, (3) general public)	Data acquired or used in citizen-made applications should be confined within the system and should not be directly visible by the citizens creating the applications unless the visibility is explicitly permitted by the system.
	Data value chain maintenance, incl. data asset management (data asset value appraisal, identification, registration and disposition)	Citizen-made applications are kept in the system to be executed later by requests. The system provides the capability to manage the applications.
	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	● GUI to provide graphical programming capability ● Capability to link Functions and IoT device wrappers by data flow ● Capability to place Functions in processing facilities ● Capability to route data among Functions and IoT device wrappers
	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	The programming facility should be usable by users with no prior experience of programming.

	Other requirements	
	Available International Standards supporting the requirements (if any)	
	References (related to above standards or other useful information (e.g. on regulatory aspects))	
Architecture considerations	○ Communication infrastructure ○ Data consistency across systems involved in the use case ○ Deployment considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	The system to support citizen-made IoT applications is composed of IoT devices and the software to provide interface to them, a network, computing resources, "Functions" that perform processing of IoT data and produce output to be passed to other Functions or to IoT devices, "Routing" function to route data to/from IoT devices and Functions, "Coordinator" that deploys Functions in computing resources and route data among Functions and IoT devices, and "Programming" facility to interface between the Coordinator and programmers by providing GUI for programming, interpreting the created programs, and transferring control information to implement the program to the Coordinator. 
General remarks		

7 Use Case – Wildlife Monitoring

7.1 Application and Use Description

In recent years, damage to field crops by wildlife such as wild boars, deer, monkeys and the like is increasing in Japan. Currently, 60% of agricultural workers are over 65 years old; moreover, population's decline in regional cities is progressing. Under these circumstances, possible labour-saving achieved by exploiting IoT and the revitalization of local cities are discussed at the COCN (Council on Competitiveness-Nippon), which is one of the measures of Society 5.0 [see reference link 1 at the end of the section]. In this project, we aim at helping to revitalize local cities.

Wildlife damage is serious also in Hakusan city (750 km^2, population 110 thousand people) in Ishikawa prefecture, which is a local city on the side of the Sea of Japan. In addition to damage to field crops, elderly people refrain from going out because they are scared of wild animals attacking them, which may result in serious problems to their health. On the other hand, wildlife is more and more valuable as products (wild meat (gibier), for instance) [see reference link 2], and an efficient usage of wildlife-derived products, by using ICT technology, is desired. Fed4IoT is going to conduct technical pilot experiments related to detection of wildlife at Kanazawa Institute of Technology Hakusan Campus, followed by field trials near the residency of Hakusan City, with the goal of creating value by sharing the data, also with other smart cities, in the future.

Examples of specific wildlife damage prevention measures by municipalities are as follows:

- Catch the wildlife: wild boars and deer caught in a trap are processed into gibier, furs, etc.
- Scare wildlife away by the residents: as for monkeys, it is impossible to catch them, contrary to wild boars and deer. When monkeys appear, people scare monkeys away and let the group of monkeys learn that they cannot get food in human residential areas.

However, the cost of wildlife damage prevention measures is a problem because of the following reasons.

- After animals are caught in a trap, if time passes, they cannot be used for meats and furs anymore: they struggle to escape from the trap and, as a result, their bodies are damaged and consequently disposed of. In such a case it is impossible to obtain profits. In addition, disposal costs are incurred.
- It takes a lot of effort to constantly monitor for wildlife to show up.

These problems can be solved with the IoT technology. By attaching a communication function to surveillance cameras, sensors, etc., people can know the status of traps, remotely and in real time without the need to physically go find each one of them. It is possible to obtain profits by

quickly handing wild boars and deer to the processors. For monkeys, it is possible to scare the monkeys away, before the monkeys steal farm products, by real time monitoring their approaching to human residential areas. By sharing information such as the type, and number, of captured animals among the processors, restaurants, etc., it is possible to much better utilize materials such as gibier and furs. By matching the demand and the supply, it is possible to avoid wasting such resources, and it will be possible to more responsibly manage the wildlife.

Reference links

[1] http://www8.cao.go.jp/cstp/english/society5_0/index.html

[2] http://www.maff.go.jp/e/data/publish/attach/pdf/index-93.pdf

7.2 Architecture

The IoT system for wildlife monitoring we want to realize consists of sensor devices, networks and clouds. The following functions are implemented in each component.

Sensor devices

Remotely notify the number and type of animals captured in the trap by the surveillance camera.

Notify only opening / closing information of electronic cages.

Notify the existence of animals with infrared sensors.

Network

Information on sensor devices installed outdoors is sent to the IoT-GW via the wireless sensor network and transferred to the Cloud via the Internet.

For the wireless sensor network, LPWA (Low Power Wide Area) is the most promising technology. Since usually there is no power line outdoors, battery operation is required for the end devices.

We aim at using ICN (Information Centric Network) technology to efficiently collect information from many sensor devices.

Cloud

Information on the type, number and place of captured animals is provided to meat processors and restaurants.

Information about distribution of animals within the territory is provided to hunters and farmers.

IoT slice and Context Information Sharing

Capabilities of the specific sensing technologies are provided to other applications, in order to develop other services by using the same real sensors/cameras, by means of virtualization techniques.

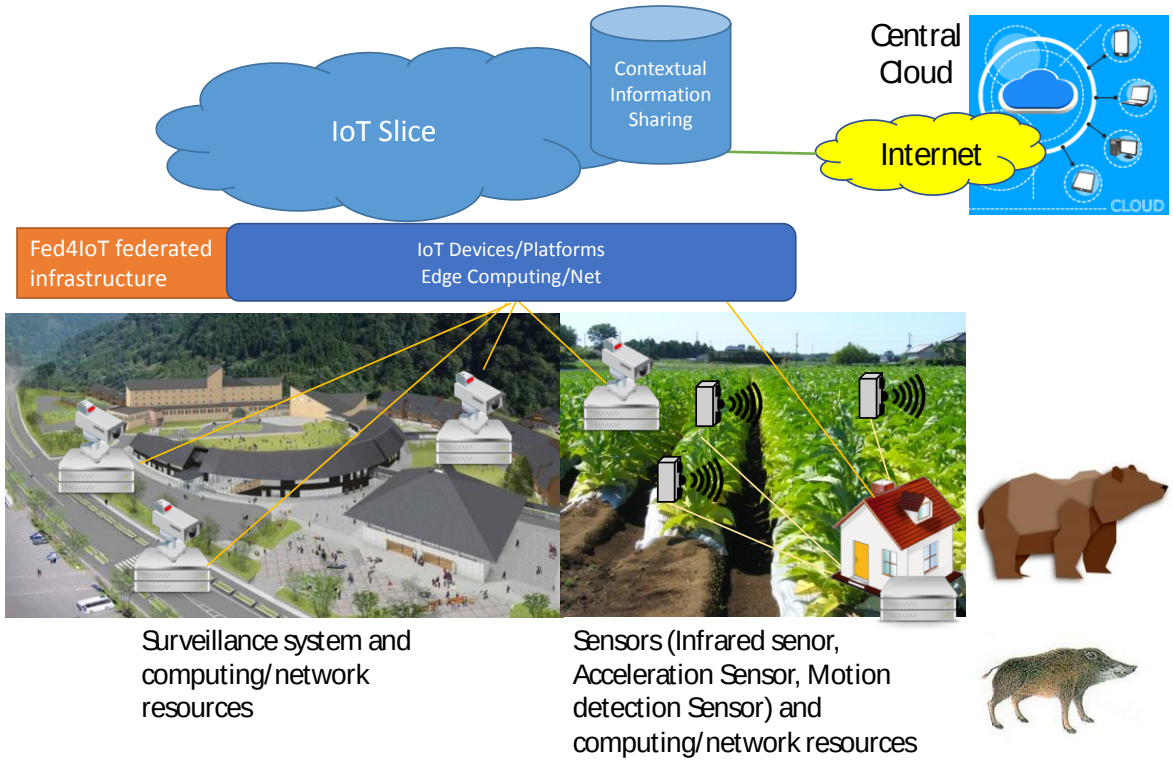


Figure 12. Overview of the Wildlife Monitoring System

7.3 Data Sets

(1) Information obtained from a sensor device attached to traps.

- Types of trapped animals.
- Number of trapped animals
- Places of trapped animals
- Time of trapped animals

(2) Information obtained from sensor devices installed in houses, farms.

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

(3) Information on sensing technology for diverting to other applications.

- Sensing device type
- Sensor network configuration
- Data collection method
- Virtualized devices

7.4 Deployment View and Interoperability Aspect

The long-term goal is to share the above data with other smart cities in a standardized data structure, and to use it widely as a sensing technology applicable to wildlife damage prevention measures and other applications. Towards interoperability, we aim to comply with the oneM2M specification and to define the data model in a format conforming to the international standard. System verification of the wildlife monitoring will be compliant with oneM2M guidelines. Based on the result of verification, a contribution to the global standard will possibly be submitted.

An important aspect is virtualization of sensors and cameras, so that the same real hardware can be shared among different applications and use cases. This is going to be achieved through the concept of IoT slices.

7.5 FG-DPM Unified Use Case Template

FG-DPM D1.1 unified use case template complemented by field-specific guidelines		
Use case title	Name	**Wildlife monitoring (Hakusan City)**
	Domain – Cross domain	**Smart City**
	Version	**0.1**
	Source	**Fed4IoT**
Objective	**Real-time monitoring of the wildlife intrusion to farms. Sharing of the information of trapped wildlife among farmers, food/furs processors, restaurants, hunters, and other smart city platforms.**	
Background	Current practice	Wildlife damage is serious also in Hakusan city (750 km 2, population 110 thousand people) in Ishikawa prefecture which is a local city on the side of the Sea of Japan. In addition to damage to field crops, elderly people refrain from going out scared of wildlife showing up. As a result, there is serious problem regarding their health. On the other hand, wildlife is valuable as wild meat and fur products, and more efficient usage is desired by utilizing ICT technology. However, the cost of wildlife damage prevention measures is a problem because of the following reasons. (1) After animals are caught in a trap, if time passes, they cannot be utilized for meats and furs. Since they struggle to escape from the trap, as a result, their bodies are damaged and must be disposed of. In such case it is impossible to obtain profits. In addition, disposal costs are incurred. (2) It takes a lot of effort to constantly monitor for wildlife to show up.
	Rational for the use case	These problems can be solved with the IoT technology. By attaching a communication function to surveillance cameras, sensors, etc., people can know the status of traps remotely in real time without having to physically reach each one of them.

Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	Examples of wildlife damage prevention measures by municipalities are as follows. (1) Catch the wildlife Wild boars and deer caught in a trap are processed into gibier, furs, etc. (2) Scare wildlife away by the residents As for monkeys, it is impossible to catch them as we do with wild boars and deer. When monkeys appear, people scare monkeys away and let the group of monkeys learn that they cannot get food in human residential areas.
Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	By attaching a communication function to surveillance cameras, sensors, etc., people can know the status of traps remotely in real-time.
Detailed scenarios	(same structure than "Scenario")	By attaching a communication function to surveillance cameras, sensors, etc., people can know the status of traps remotely in real time without looking around the traps. It is possible to obtain profits by quickly handing wild boars and deer to the processor. For monkeys, it is possible to scare the monkeys away before the monkeys take farm products by real time monitoring of the approach to human residential area. By sharing information such as the type and number of captured animals among the processors, restaurants, etc., it is possible to utilize materials such as gibier and furs.
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics ○ Data accessibility ○ Data availability ○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability	(1) Information obtained from a sensor device attached to traps. • Types of trapped animals. • Number of trapped animals • Places of trapped animals • Time of trapped animals (2) Information obtained from sensor devices installed in houses, farms. • Where animals are • Types of animals • Number of animals • Time when animals were detected (3) Information on sensing technology for diverting to other applications. • Sensing device type • Sensor network configuration • Data collection method

	Data format, incl. standard, structured	
DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl. 5Vs of Big Data) ○ Data models and modelling ○ Data backup, archiving and recovery ○ Event management ○ System resilience ○ System sustainability	By sharing information such as the type and number of captured animals among the processors, restaurants, etc., it is possible to utilize materials such as gibier and furs.
	Data application to the different interests, incl. stakeholders' interests	

Governance and data life cycle considerations	Data accountability	
	Data isolation	
	Personal data (incl. sensitive personal data)	
	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	
	SLAs enforcement	
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main groups of internal employees, (2) external business partners, (3) general public)	
	Data value chain maintenance, incl. data asset management (data asset value appraisal, identification, registration and disposition)	
	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	Outdoor sensing data must be collected. Real time monitoring capability is required. Connectivity between the Cloud and the sensor device must be guaranteed. Collected data is accessible by other IoT systems.

	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	**Under discussion among municipalities.**
	Other requirements	
	Available International Standards supporting the requirements (if any)	ITU-T Y.3071 (03/2017): Data aware networking (information centric networking) – Requirements and capabilities ITU-T Y.4113 (09/2016): Requirements of the network for the Internet of things
	References (related to above standards or other useful information (e.g. on regulatory aspects))	LoRa alliance
Architecture considerations	○ Communication infrastructure ○ Data consistency across systems involved in the use case ○ Deployment considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	The IoT system realizing wildlife monitoring consists of sensor devices, networks and clouds. The following functions are implemented in each component. (1) Sensor devices • Remotely notify the number and type of animals captured in the trap by the surveillance camera. • Notify only opening / closing information of electronic cages. • Notify the existence of animals with infrared sensors. (2) Network • Information on sensor devices installed outdoors is sent to the IoT-GW via the wireless sensor network and transferred to the cloud via the Internet. • Utilize ICN (Information Centric Network) technology to efficiently collect information from many sensor devices. (3) Cloud computing • Information on the type, number and place of captured animals is provided to meat processors and restaurants. • Distribution information on animals is provided to hunters and farmers. • Providing information on sensing technologies for application to other services. (4) IoT slice and Context Information Sharing • Capabilities of the specific sensing technologies are provided to other applications, in order to develop other services by using the same real sensors/cameras, by means of virtualization techniques.
General remarks	Fed4IoT is going to conduct technical pilots related to detection of wildlife at Kanazawa Institute of Technology Hakusan Campus, followed by field trials near the residency of Hakusan City, with the goal of creating value by sharing the data with other smart cities in the future.	

8 Requirements

This is a collection of key requirements, drawn from each use case. For each requirement we have tried to identify the main layer it impacts, broadly distinguishing between: i) Fed4IoT, when it mainly impacts the technologies we are going to develop within the scope of the project; ii) IoT Layer when it impacts the technology and networking of sensors/actuators already in place or to be deployed; iii) Infrastructure when it mainly impacts the data communications network; iv) Application when it mainly impacts the design of the parts that are visible to end-users.

8.1 Requirements for Smart Parking

ID	Description	Rationale	Fit Criterion	Layers it impacts
SP-1	Data collection infrastructure	Private parking site info must be provided in real time. RPZ info is provided at the end of the day	Data collection stations and network services operating in real-time	IoT Layer, Infrastructure
SP-2	Cloud-centric	Complex tasks should be achieved by virtualized services in the cloud	Modular cloud services, offering APIs for complex tasks	Fed4IoT, Application
SP-3	Virtualization	Deployment of virtualized services in a specific slice	The same data can be presented to different applications and/or differently aggregated	IoT Layer, Infrastructure, Fed4IoT
SP-4	Reliability	Provided data must be reliable for a right performance of the use case	Fed4IoT solutions must not impact IoT Layer reliability	IoTLayer, Fed4IoT
SP-5	Interoperability & Data Federation	All components must support NGSI-LD data format	Provide a common representation	IoTLayer, Fed4IoT

8.2 Requirements for Cross Border Person Finder

ID	Description	Rationale	Fit Criterion	Layers it impacts
CBPF-1	Real-time	The required data must be provided in time to obtain the right performance and exact time (stamp) management. Request/response for person finding must flow in real time	In-time information	IoT Layer, Infrastructure, Fed4IoT
CBPF-2	Event-driven model	Person found info is provided upon match events	Design allows event-driven or pub/sub throughout the whole system	IoT Layer, Infrastructure, Fed4Iot
CBPF-3	Privacy	Intermediate features extracted and stored by the system must not be linked back to the specific person	Pseudonimity/ anonymity is provided transparently to upper layers	IoT Layer
CBPF-4	IoT Slice on the edge	The same logical slice has to be distributed over several distinct edge resources	Design allows for distribution of IoT resources	Fed4IoT
CBPF-5	Camera virtualization	The same real camera is to be used by different applications at the same time (e.g. for Person Finder and Waste Management)	Design allows for abstraction of IoT resources	Fed4IoT
CBPF-6	Device security	Data shall be signed by the IoT device/camera to authenticate that the data is created by authorised devices only, and also ensure the data integrity	Design allows for digital signing	IoT Layer, Fed4IoT

8.3 Requirements for Waste Management

ID	Description	Rationale	Fit Criterion	Layers it impacts
WM-1	Real-time monitoring infrastructure	Garbage sites data should be collected in real time.	The deployment shall include an infrastructure composed of smart	IoT Layer, Infrastructure

			cameras	
WM-2	Edge processing	In the edge, processing capability is needed to carry out the local data processing.	The collected data shall be processed locally in the device or gateway level	IoT Layer
WM-3	Message brokers	Connectivity between different components via message brokers must be guaranteed	At least one message broker shall be included in each deployment for communication	IoT Layer
WM-4	Virtualized services	One physical device should be able to support virtualization to different services	Resources can be shared among different applications	Fed4IoT
WM-5	Data interoperability	The different deployments over sites should be able to automatically communicate with each other under the right configuration	Standardized information model to achieve interoperability is used	IoT Layer, Fed4IoT
WM-6	Data security	Data integrity and data provenance verification should not be endangered by data being copied and replicated throughout the different message and context brokers	Fed4IoT's security mechanisms are designed for compliance with the concept of data federation	IoT Layer, Fed4IoT

8.4 Requirements for Citizen Made IoT Applications

ID	Description	Rationale	Fit Criterion	Layers it impacts
CMIA-1	Storage	Data collection may be performed as a part of the application-level functionality. Storage to hold collected data is required if historical data is needed to implement applications	A solution must be designed, deciding at which level/component storage of historical IoT data must occur	Application, Fed4IoT
CMIA-2	Edge processing	Home IoT system is an edge system at its minimum configuration	Processing capability must exist within an IoT system located at the edge	IoT Layer, Fed4IoT
CMIA-3	Cloud functions	Functions running on different IoT platforms may be required	Processing capability in fog, cloud, or other IoT domains	Fed4IoT

CMIA-4	Local communication	IoT devices and Functions must communicate	Communication that stays local to an IoT system is supported	Infrastructure
CMIA-5	Virtualized functions	Programming components including functions and IoT device drivers are virtualized functionalities	The system must cope with virtualization of functions	Fed4IoT

8.5 Requirements for Wildlife Monitoring

ID	Description	Rationale	Fit Criterion	Layers it impacts
WM-1	Outdoors data collection infrastructure	Outdoor sensing data must be collected	Existence of outdoor sensors and wired communication functionalities.	IoT Layer, Infrastructure
WM-2	Communication	Connectivity between the Cloud and the sensor device must be guaranteed	Continuous online availability of services and devices.	Infrastructure
WM-3	Resources virtualization	The monitoring infrastructure may be used by other applications (such as the cross-border person finding one)	Techniques for lightweight duplication of real-time data flows	Fed4Iot

8.6 Considerations about requirements

All use cases, obviously, require that a robust and real-time communications infrastructure be in place between all sensors, devices, and servers.

Many cross-scenario requirements clearly emerge. Specifically:

- There is a clear indication of the benefits of **virtualization** strategies that allow the same resources to be used by different applications. At the same time, virtualization techniques must seamlessly span both the edge and the cloud.
- There is a strong indication that the business logic and data processing capabilities of each application are going to be **distributed** among cloud, edge, and local resources.
- There is a strong need for data **interoperability** and standard formats.
- There is a requirement for **privacy** and pseudonimity techniques that ensure sensible data stays confined near the local nodes (or at edge) and private information leaking to the cloud is non-linkable back to the specific individuals.
- There is a requirement for **security** techniques that are able to ensure integrity and authenticity of data regardless of the various processing steps and of the different nodes it crosses.

We think that the major challenges the project is going to face are thus related to the ability of implement virtualization and federation strategies specific for the IoT eco-system, that are able to cope with local/edge/cloud distribution, function virtualization, data-centric security, real-time data flows and interoperability with existing systems.

9 Summary

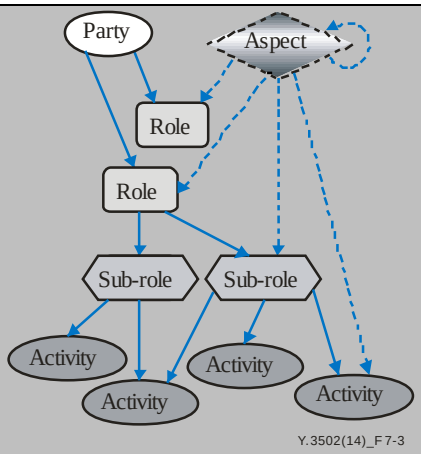
This document addresses the work done during Task 2.1 where we have identified five use cases for the application of our Fed4IoT framework to different domains. These use cases are well-grounded on partner's concrete needs and availability of data and infrastructure. Additionally, each use case clearly shows how the current situation is going to be improved by the novel concepts put forward by Fed4IoT.

The narrative of each use case description is unified to highlight the elements involved in each use case, the flow of information, and even the requirements in term of context representation where the virtualization and slicing of devices is presented. Additionally, we have used the ITU-T FG-DPM template to provide a thorough view of each use case description.

Finally, another result worth-highlighting of this deliverable is the depiction of a first and rough view the architecture of our framework and the application to each use case.

10 Annex 1 – FG-DPM Template

FG-DPM D1.1 unified use case template complemented by field-specific guidelines		
Use case title	Name	
	Domain – Cross domain	
	Version	
	Source	
Objective		
Background	Current practice	
	Rational for the use case	
Ecosystem	○ Stakeholder roles and responsibilities ○ Stakeholder relationships	It is required to consider – if and as useful – the distinction between "**business roles**" and "**actors**" (an actor can play one or more roles) **Stakeholders: roles and actors** The following understanding of "role (business role)" is adopted in several ITU-T Recommendations in different technical areas (Cloud Computing, IoT, Big data, NGN etc.). It is largely inspired by ISO guidelines for designing architectures (which also inspired – among others – some Cloud Computing Recommendations jointly developed by ITU-T and ISO). NOTE 1 - In some specs, the term "business role" is used instead of "role". NOTE 2 – Roughly, activities can be associated to functionalities. Role [Y.3052]: A set of activities that serves a common purpose. ([ITU-T Y.3502] Recommendation ITU-T Y.3502 (2014), *Information technology - Cloud computing - Reference architecture)* For a more general view, see below some text and a figure from Y.3502 (the "party" - alternatively called "player" or "actor" - can play multiple roles). *"Activity, role and sub-role are defined in [ITU-T Y.3502] as follows:* ***3.2.1*** ***activity****: A specified pursuit or set of tasks.* ***3.2.7*** ***role****: A set of* ***activities*** *that serves a common purpose.* ***3.2.9*** ***sub-role****: A subset of the* ***activities*** *of a given* ***role****.*

		 Ecosystem illustration with role, sub-role, activity and party NOTE – Three examples of ecosystems (referring ITU-T Y.2060/4000, ITU-T Y.4114 and ITU-T Y.3502) - with related actors and roles - are provided for information in Appendix I.
Scenario	○ Contextual illustration ○ Pre-requisites ○ Pre-conditions (if any) ○ Triggers ○ Typical operational procedure ○ Process flow diagram ○ Post-conditions ○ Information exchange ○ Considerations on publicity of results (if any)	
Detailed scenarios	(same structure than "Scenario")	With respect to the above basic scenario, presenting typical work patterns for different stakeholder roles. In this field (or in the above "Ecosystem" field), it is also suggested to describe– if applicable - "value-added services" which might be associated to the use case. Value-added services are intended as services which might be added at little or no extra cost to existing IoT infrastructures with application of new software acting upon existing data sources.
Data characteristics, quality and formats	Data input characteristics ○ Data granularity ○ Characteristics of meta data Data output characteristics ○ Data accessibility ○ Data availability	It is suggested - where applicable - to separate the different pieces of information to be provided for this field (as an example, a part to be filled in by users of data in output, another part to be filled in by data processing experts).

	○ Data traceability Data quality considerations ○ Data authenticity ○ Data reliability ○ Data integrity ○ Data usability Data format, incl. standard, structured	
DPM capabilities considerations	Data processing capabilities ○ Aggregation and grouping ○ Cleaning and filtering ○ Classification and indexing ○ De-identification, anonymization and pseudonymization ○ Transfer ○ Pre-processing and processing ○ Analysis and analytics ○ Reading and query ○ Visualization Data management capabilities ○ Access and use ○ Administration ○ Acquisition and collection ○ Creation ○ Preservation incl. protection ○ Sharing ○ Storage ○ Update Considerations on system capabilities ○ Functions and operations ○ Service Level Agreements (SLAs) ○ Performance (incl. 5Vs of Big Data) ○ Data models and modelling ○ Data backup,	The description should focus on considerations about the DPM capabilities, not on the solutions adopted in the use case for the implementation of the DPM capabilities Where applicable, it is suggested also to separate the requirements-related text of the DPM capabilities from the text related to the solution adopted for the implementation of the DPM capabilities.

	archiving and recovery ○ Event management ○ System resilience ○ System sustainability	
	Data application to the different interests, incl. stakeholders' interests	
Governance and data life cycle considerations	Data accountability	
	Data isolation	
	Personal data (incl. sensitive personal data)	
	IPR and Licensing ○ Open data vs private data ○ Licenses of data use and reuse	
	SLAs enforcement	
	Risk management, incl. different concerns and dimensions and of risks (cybersecurity, privacy, safety, risks assessment, change management)	
	Data distribution ○ Technical management considerations on data distribution ○ Data access rights and data authorization considerations according to the different stakeholders (e.g. in a smart city scenario, (1) main groups of internal employees, (2) external business partners, (3) general public)	
	Data value chain maintenance, incl. data asset management (data asset value appraisal, identification, registration and disposition)	

	Incident management process	
	Continuous improvement process, incl. data minimization	
Requirements	Functional requirements (with respect to the different DPM capabilities indicated above)	
	Non-functional requirements, incl. ○ Availability ○ Data continuity ○ Flexibility ○ Interoperability ○ Reliability ○ Safety ○ Security and privacy ○ Trust (incl. traceability)	
	Other requirements	
	Available International Standards supporting the requirements (if any)	
	References (related to above standards or other useful information (e.g. on regulatory aspects))	
Architecture considerations	○ Communication infrastructure ○ Data consistency across systems involved in the use case ○ Deployment considerations ○ Interface requirements, incl. user interfaces and APIs ○ Performance criteria	
General remarks	It is suggested to include information on reference implementations where applicable, as well as add other information such as reporting on incidents on security or privacy of the use case if it has been already implemented. It is also suggested that the expected impact of the use case (social, economic or environmental aspects) could be also highlighted here.	