Deliverable D3.6

Enhancement and Configuration of Open Platforms and Reusable Components

WP 3

November 15th, 2019

Additions to existing components required by the IoF2020 use cases, aiming at the realisation of modules that are also applicable for future usage.

IoF2020 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731884. Visit iof2020.eu for more information about the project.

DOCUMENT IDENTIFICATION

Project Acronym	**IoF2020**
Project Full Title	Internet of Food and Farm 2020
Project Number	731884
Starting Date	January 1st, 2017
Duration	4 years
H2020 Call ID & Topic	IOT-01-2016
Date of the DoA	23.08.2018
Website	www.iof2020.eu

File Name	D3.6-OpenPlatform Enhancement and Configuration of Open Platforms and Reusable Components.docx
Date	November, 15th 2019
Version	121
Status	Final Draft
Dissemination level	PU: Public
Authors	José M. Cantera, Harald Sundmaeker, Rita Campos, Francisco de la Vega, Isabel Neira, Daoud Urdu, Ignacio Gómez, Francisco Maroto, Jason Fox, Klaus Förderer
Contact details of the coordinator	George Beers george.beers@wur.nl

PROJECT SUMMARY

The internet of things (IoT) has a revolutionary potential. A smart web of sensors, actuators, cameras, robots, drones and other connected devices allows for an unprecedented level of control and automated decision-making. The project Internet of Food & Farm 2020 (IoF2020) explores the potential of IoT-technologies for the European food and farming industry.

The goal is ambitious: to make precision farming a reality and to take a vital step towards a more sustainable food value chain. With the help of IoT technologies higher yields and better-quality produce are within reach. Pesticide and fertilizer use will drop and overall efficiency is optimized. IoT technologies also enable better traceability of food, leading to increased food safety.

Nineteen use-cases organised around five trials (arable, dairy, fruits, meat and vegetables) develop, test and demonstrate IoT technologies in an operational farm environment all over Europe, with the first results expected in the first quarter of 2018.

IoF2020 uses a lean multi-actor approach focusing on user acceptability, stakeholder engagement and the development of sustainable business models. IoF2020 aims to increase the economic viability and market share of developed technologies, while bringing end-users' and farmers' adoption of these technological solutions to the next stage. The aim of IoF2020 is to build a lasting innovation ecosystem that fosters the uptake of IoT technologies. Therefore, key stakeholders along the food value chain are involved in IoF2020, together with technology service providers, software companies and academic research institutions.

Led by the Wageningen University and Research (WUR), the 70+ members consortium includes partners from agriculture and ICT sectors, and uses open source technology provided by other initiatives (e.g. FIWARE). IoF2020 is part of Horizon2020 Industrial Leadership and is supported by the European Commission with a budget of €30 million.

EXECUTIVE SUMMARY

This report is intended to develop and extend the main pillars, revolving around the universal concept of “Open Platforms”, that have to be considered, so that **interoperability** and **replicability** are achieved, allowing for the building of an Internet of Things (IoT) ecosystem around the IoF2020 project and ultimately around Smart Farming in Europe. Those main pillars have been identified using as main input the analysis of use cases made by **D3.3** (in architectural terms) and **D3.7** (in terms of requirements) (and updated by this deliverable):

- **Minimal Interoperability Mechanisms (MIMs)**, which can be used as a “common architectural ground” to establish new services in the Agri-food domain, thanks to a common set of technologies and standardized interfaces. This deliverable provides an update of the MIMs, formerly identified by **D3.3**. as Interoperability Points.
- **Reusable Components**, which are general purpose software components, with open and well-defined interfaces which can be used off-the-shelf across solutions. Reusable components are the backbone of Open Platforms and the most salient ones are described.
- **Semantic Interoperability**, which enables, through common information models, a shared understanding of concepts and relationships within the Agri-food domain, paving the way towards universal data sharing.
- **Service and Data Monetization**, a means to extend Open Platforms by providing advanced data and services marketplaces that allow different stakeholders to harness all the powerfulness of data sharing and service reutilization.

For realising the IoF2020 vision of replicable and interoperable Digital Farming solutions, it is necessary to develop and foster **integrated Open Platforms** (supported by the pillars described above) that can solve major development challenges, while offering an architecture compliant with the MIMs. The Open Platforms described by this technical report are:

- **FIWARE**, a result from the former EU FI PPP, has now turned into an open source platform for implementing Smart Solutions, including but not limited to Smart Farming. FIWARE includes different components gravitating around the Context Broker, a “data broker” which offers standardized interfaces to publish, store, and consume (through NGSIv2 and NGSI-LD) data tagged with associated temporal and spatial metadata (context data).
- **365FarmNet**, a popular Farm Management Information System (FMIS) software that offers open and pluggable interfaces, enabling the creation of a multiple stakeholder marketplace of Digital Farming solutions.

Going beyond this, **Farm Management Information Systems (FMIS)**, built on top of Open Platforms, are intended to provide a holistic view of farms, often offered in a Software as a Service (SaaS) schema. FMIS enable the realization of the System of Systems view in Agri-food, where solutions offered by different vendors can be integrated into a common framework for the benefit of the different users and stakeholders.

With regards to semantic interoperability, the availability of shared, well-adopted information models is a key interoperability mechanism for enabling a global market for IoT-enabled Digital Farming. Such models provide an essential element in the common technical ground needed for standards-based innovation, by making replicability and portability of Smart Farming solutions practical. This allows for a sector-specific focus in a procurement or development process, while maintaining cross-domain consistency. At this respect the work reported on this deliverable includes a:

- Simple yet-powerful **framework** for defining new information models in agri-food. NGSI and **NGSI-LD** (standardized by ETSI) are information meta-models through which concrete, domain-specific data models can be expressed in a coherent way across domains.

- Summary of **recommended design guidelines**, documentation guidelines and compliance rules for new data models, with a view to further upscale these to new trials and use cases.
- Set of **concrete Data Models**, based on the NGSI and NGSI-LD meta-models that address the needs of certain trials in IoF2020.

Once an Open Platform is consolidated as the substrate of a Smart Farming solution or FMIS, new needs might arise. For instance, how to monetize data or services. Different IoF2020 partners are offering open software components that can be instantiated to experiment with these innovative services. This report finalizes with a description of them, which are based on open interfaces and APIs defined by TMForum.

The final aim of this work is to help use case stakeholders, product owners and developers to identify configure and extend a set of common technology enablers, software components, open platforms and related architectures that guarantee the creation of **a sustainable ecosystem of portable solutions, FMIS or System of Systems** for the farm and food sector. In the end, that will foster:

- the flourishment of a marketplace composed by different vertical solutions capable of interoperating and integrating into a broader system of farm management,
- the identification and development of IoT reusable components and reference configurations and compositions in the framework of a common architecture.
- the deployment of smart agri-food solutions within a common infrastructure, “IoF2020 Lab”, enabling a collaboration space aimed at resource sharing, continuous improvement and knowledge exchange.

Finally, this IoT marketplace, enabled and empowered by standard technologies, will turn into the ideal space for collaboration and incubation of further innovations in the agri-food sector.

TABLE OF CONTENTS

ABBREVIATIONS

AEF	Agricultural Industry Electronics Foundation
AgGateway	Non-profit organization for industry's transition to digital agriculture
API	Application Programming Interface
COTS	Commercial off the Shelf Product
D	Deliverable
DSS	Decision Support System
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
EU	European Union
FIWARE	Future Internet Software Initiative
FMIS	Farm Management Information System
GS1	Global Standards One
ICT	Information and Communication Technology
IoF2020	Internet of Food and Farm 2020
IoT	Internet of Things
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LSP	Large Scale Pilot
MIM	Minimal Interoperability Mechanism
MVP	Minimum Viable Product
NGSI-LD	Next Generation Service Interfaces Linked Data
OMA	Open Mobile Alliance
SDO	Standard Development Organisation
SME	Small and Medium sized Enterprise
UC	Use Case
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
WP	Work package

1 Introduction

1.1 Introduction

One of the main objectives of the IoF2020 project, and particularly WP3, is to provide a robust technological offer around Open Platforms (realising a reference architecture) for standards-based, interoperable and replicable solutions that deliver added-value functionalities to various stakeholders in the food and farms domain. To that end, this deliverable is aimed at convincing and demonstrating use case and IoF2020 Open Call developers the advantages of relying on open platforms, to facilitate and trigger collaboration and synergies and as input for further awareness with external stakeholders. The final goal is enabling an Internet of Things (IoT) *ecosystem* around the IoF2020 project and ultimately around Smart Farming in Europe.

This document identifies and summarizes the main pillars, revolving around the concept of Open Platforms, that have to be considered by individual solutions (particularly new use cases resulting from the IoF2020 Open Call), Farm Management Information Systems (FMIS), or composable System of Systems configurations. The final aim is to implement the different use cases in a way that **interoperability** and **replicability** are achieved.

Several main pillars have been identified using as main input a former analysis of use cases (requirements, reusable components, etc.) made by **D3.7** (and updated by this deliverable) and previous work on interoperability described by **D3.3**:

- **Minimal Interoperability Mechanisms (MIMs)**, which can be used as a "common architectural ground" to establish new services in the Agri-food domain, thanks to a common set of technologies and standardized interfaces. Open Platforms are enabled by these MIMs. This deliverable provides an update of the MIMs formerly identified by **D3.3**. as Interoperability Points.
- **Reusable Components**, which are general purpose software components, with open and well-defined interfaces which can be used off-the-shelf across solutions. Reusable components are the backbone of Open Platforms and the most salient ones are described.
- **Semantic Interoperability**, which enables, through common information models, a shared understanding of concepts and relationships within the Agri-food domain, paving the way towards universal data sharing.
- **Service and Data Monetization**, a means to extend Open Platforms by providing advanced data and services marketplaces that allow different stakeholders to harness all the powerfulness of data sharing and service reutilization.

The recommendations made by this report provide a "common ground" to leverage the IoF2020 IoT-based innovations based on Open Platforms in the upcoming phases of the project, both within each use case, spawning across multiple use cases.

To better understand the context around this deliverable, it is encouraged to read **D3.2** which contains the technological analysis and the architecture of the different solutions proposed to address the IoF2020 use cases. Besides, **D3.9** should be read so that a proper understanding of the use case *synergies* is gained. The recommendations made by **D3.3** have served as a valuable input for *Task 3.3, Open Platforms*. In fact, it has guided the definition and selection of platforms (some of them described by this deliverable) and components during the upcoming implementation of the use cases to be developed.

Chapter 2 makes a summary and update of the fundamental requirements identified by IoF2020. *Chapter 3* provides an updated reference architectural view as a pillar to guide use cases and open call applicants. Such chapter provides a summary of the architectural layers and the Minimal Interoperability Mechanisms (MIMs). For each MIM one or more relevant standard technologies have already been identified by **D3.3**. *Chapter 4* describes and updates **D3.7** about *IoT reusable components* to be assembled within Open Platforms. *Chapter 5* provides an overview of existing Information Models with a view to enable semantic interoperability. *Chapter 6* deals with new developments made by WP3 towards *Semantic Interoperability* in the agri-food domain: new developments in standards for the semantic description of devices (Web of

Things), Reference Common Information Models developed by WP3 in collaboration with use cases in different trials and new libraries to make the integration of digital farming solutions easier.

Chapters 7 and 8 describe two consolidated Open Platforms for agri-food: FIWARE (open source) and 365FarmNet (which aims at defining a marketplace of integrable solutions around the concept of Farm Management Information Systems (FMIS)). Chapter 9 is devoted to data and service monetization in such Open Platforms by leveraging data marketplaces and open tools based on standard APIs (TM Forum).

Finally, the last chapter captures conclusions, outlook and action plan during next phases of the project.

1.2 Approach and Methodology

In order to provide a robust technological offer around Open Platforms (realising a reference architecture) for standards-based, interoperable and replicable solutions, WP3 has been working on identifying the **main pillars for Open Platforms** (introduced by the previous section), following the methodology described by figure 1.

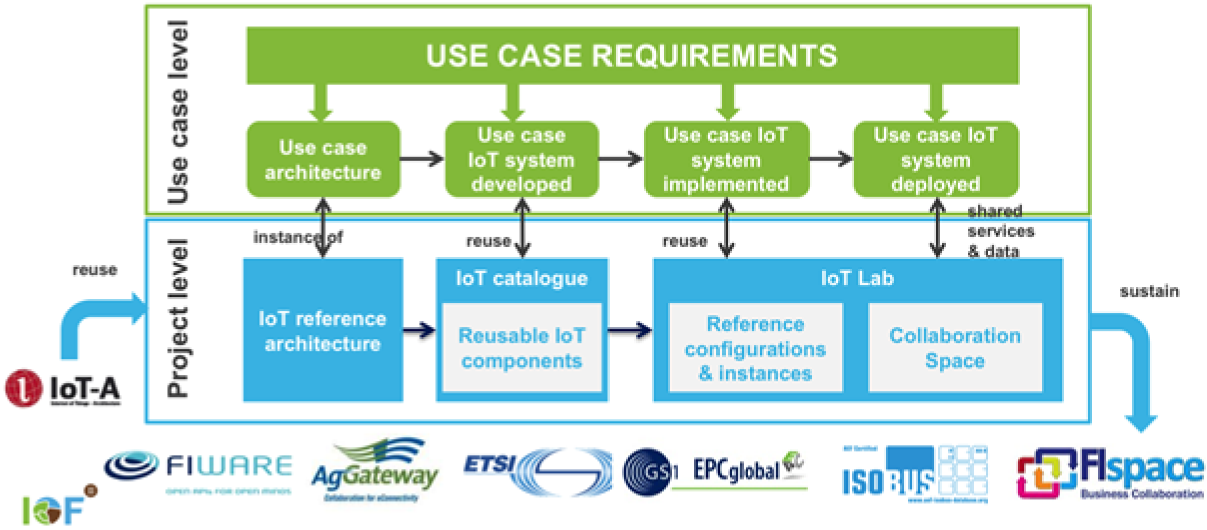


Figure 1: IoF2020 Large Scale Pilot vision and approach.

To realize the IoF2020 vision (depicted by figure 1) the following assets have to be taken as starting point:

- the requirements (both functional and non-functional) of the use cases and trials (**D3.7**), gathered in close collaboration with WP2.
- the list of potential reusable components resulted from the analysis already made by **D3.7**.
- the key and common Minimal Interoperability Mechanisms, MIMs (named as Interoperability Points by **D3.3**) that allow the different use cases and trials to rely on a common Reference Architecture.

Last but not least, a common framework for collaboration has been identified revolving around the System of Systems approach, where multiple and complementary Agri-food solutions can be integrated to provide a holistic view to users and stakeholders. This new approach is going to be trialled by some of the new use cases that are beneficiaries of the IoF2020 Open Call.

Once the main pillars have been identified, they have been developed in more detail by providing an extended analysis and guidelines for their realization. Particularly, the availability of shared, well-adopted information models is paramount to realize the key pillar of Semantic Interoperability, enabling a global market for IoT-enabled Digital Farming. Such models provide an essential element in the common technical ground needed for standards-based innovation, by making replicability and portability of smart agri-food solutions practical. This allows for sector-specific focus in a procurement or development process, while maintaining cross-domain consistency. At this respect the work reported on this deliverable includes:

- a simple yet-powerful **framework** for defining new information models in agri-food. NGSI and **NGSI-LD** (standardized by ETSI) are information meta-models through which concrete, domain-specific data models can be expressed in a coherent way across domains.
- a summary of recommended **design guidelines**, documentation guidelines and compliance rules for new data models, with a view to scale further to new trials and use cases.
- a set of concrete **Data Models**, based on the NGSI and NGSI-LD meta-models, that address the needs of certain trials in IoF2020.

In addition, as an update and extension to the previous work described by **D3.3** and **D3.7**, and given the importance of semantic interoperability at the IoT Service Layer in the agri-food domain, the new **W3C Web of Things** Thing Description (TD) standard has been presented, together with a proposed architecture of configurable and reusable components for seamless integration with the Information Management Layer (based on NGSI-LD).

For realising the IoF2020 vision of replicable and interoperable Digital Farming solutions, it is necessary to develop and foster **integrated Open Platforms** (supported by the main pillars) that can solve major development challenges, while offering an architecture compliant with the MIMs. The Open Platforms described by this technical report are:

- **FIWARE**, which includes different components gravitating around the Context Broker, a "data broker" which offers standardized interfaces to publish, store, and consume (through NGSIv2 and NGSI-LD) data tagged with associated temporal and spatial metadata (context data).
- **365FarmNet**, a popular Farm Management Information System (FMIS) software that offers open and pluggable interfaces, enabling the creation of a multiple stakeholder marketplace of Digital Farming solutions.

Once an Open Platform is configured and consolidated as the substrate of an individual Smart Farming solution or a whole Farm Management Information System (FMIS) integrating multiple solutions (System of Systems), new needs might arise. For instance, how to monetize data or services. Different IoF2020 partners are offering open software components that can be instantiated to experiment with these innovative services. This report finalizes with a description of them, which are based on open interfaces and APIs defined by TMForum.

The final aim of this work is to help use case stakeholders, product owners and IoF2020 Open Call developers to identify, configure or even enhance Open Platforms, leveraging common technology enablers, software components, and related architectures that guarantee the creation of a sustainable ecosystem of portable solutions for the Farm and Food sector. In the end, that will foster (as depicted by figure 1):

- the flourishment of a marketplace composed by different vertical solutions capable of interoperating and integrating into a broader ecosystem of farm management
- the identification and development of IoT reusable components (described by a catalogue) and reference configurations and compositions in the framework of a common architecture.
- the deployment of smart agri-food solutions within a common infrastructure, "IoF2020 Lab", enabling a collaboration space aimed at resource sharing, continuous improvement and knowledge exchange.

2 Requirements for Open Platform Components

2.1 Introduction

During the analysis of use cases realised in the first 18 months of the IoF2020 project, the WP3 team was looking for “additional reusable components” that could be offered to the IoF2020 use cases as well as to a wider stakeholder target audience. The underlying objective of using reusable or generic software components is to reduce efforts for implementing IoT based systems, increasing quality of systems as well as shortening the duration required for delivering such solutions. As previously outlined in deliverable D3.7 on the compilation of use case requirements, the IoF2020 team was focusing on the identification of reusable components that comply with the following criteria:

- Relevance to Use Cases – assessing how many of the use cases and a potential wider stakeholder target audience could use the envisaged additional reusable component.
- Innovation – analysing the originality of the component and its contribution to a relevant advance for realising IoT based solutions.
- Feasibility and Impact – identifying the viability of the implementation in the time frame of the use cases.

At the same time, the WP3 team had to have in mind the overall context of the IoF2020 project. One has to keep in mind that the “IoF2020 Large Scale Project” means to realise “*not just one*” research, technological development and innovation project, but more a kind of ***farm and food innovation programme that assembles over 100 organisations and materialised by a community around 33 use case projects***. These are 19 initial use cases starting from the beginning of IoF2020, and 14 use cases selected via an open call procedure, starting their operation in month 24 of the project lifetime. The following Figure 2 presents an overview of those 33 use cases, highlighting their number, title, topic and countries of involved organisations. The colour indicates the grouping by the five IoF2020 trial areas (i.e. yellow = arable, blue = dairy, purple = fruits, green = vegetables, red = meat).

Figure 2: The 33 use cases projects, structured in the 5 trial areas arable, dairy, fruits, vegetables and meat. See also www.iof2020.eu/trials

When aiming at the identification of open platform components, one must not forget the heterogeneity of the different trial areas with regards to different aspects.

Arable solutions are faced by harsh conditions in open fields with a mobile operation of equipment like tractors, implements and combines. Transportation and usage of seeds, fertilizer, pesticides, or even production can be planned in advance. However, they are often facing changing situations, due to effectiveness in planning, weather conditions and availability of machinery.

With respect to fruits and vegetables, there also diverse characteristics that cannot be compared, for instance, in the scope of machinery that can be deployed for treating and e.g. a vineyard or for olive trees. The former is characterised by highly mobile equipment that is rather brought to the plants than bringing plants to the infrastructure, as it is done in greenhouse-based farming. Greenhouses are often highly automated and controlled, usually not lacking energy for operating the environment, but often asking for specific installations, due to an indoor situation.

The dairy sector combines an out- and indoor situation in itself. Use cases deploying collars for e.g. monitoring the grazing time and location of cows or also fixed installations in the stables like milking robots for an automated process, nowadays also used for gathering additional data for analysing the health status of cows. Last but not least, the meat sector implies quite some interaction of animals and finally the meat with the human operators in the process. This offers opportunities to use not only animal data, but also data related to the process operation by the employees during animal handling or processing of meat.

As a consequence of these scenarios, it is obvious that also the envisaged and developed IoT based solutions are of diverse kind, while asked to comply with diverse quality requirements. Sensors that are highly recommendable for one use might be too inaccurate for another. Costs of IoT solutions that are very meaningful for operating them in combination with a tractor, might be rather inacceptable for locally analysing yield, temperature, moisture, or existence of parasites.

At the same time, we must not forget the preparedness of the end-user organisations. IoF2020 is including diverse stakeholders from large industrial players that benefit from a mass effect of implementations, while smallholder farms are rather challenged by the number of promising solutions and efforts to manage the basic infrastructure, while scaling effects cannot be capitalised.

Therefore, the team considers the IoF2020 project as an excellent opportunity to get a grip on those requirements and being able to aggregate all the lessons learnt while being aware of the overall complexity. However, to gather and communicate experience gained in tools like the IoT catalogue (www.iot-catalogue.com), we had to analyse the use cases with respect to different dimensions as also presented in deliverable D3.10 about the selected use cases via the open call. The following Figure 3 is presenting the structure used for differentiating reusable and individual components.

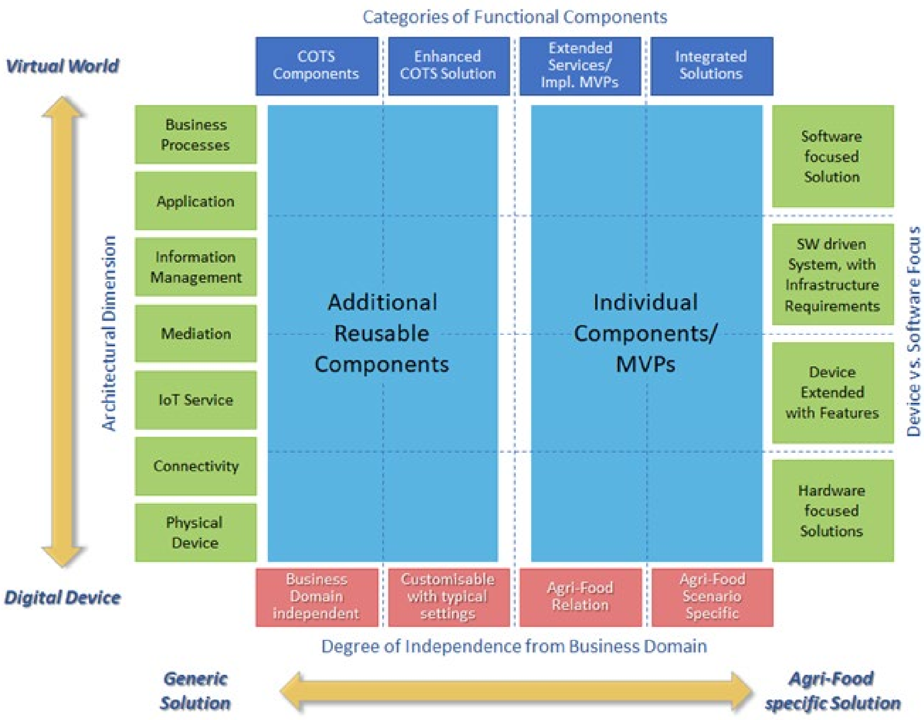


Figure 3: Structure for use case analysis.

The following sections 2.2, 2.3, and 2.4 are summarising key requirements that were identified in the collaboration of WP3 with the different use cases in WP2 of the IoF2020 project.

2.2 Openness of Access to Data and Services

A general topic to be taken into account is the access to data and services. This was already highlighted in the very early stage of the IoF2020 project and specifically discussed with the trial chairs in the IoF2020 meeting hosted in Almeria in February 2018. Subsequently, the trial related meetings were taking up that topic, discussing the specific data and functionality required and on how to facilitate access and exchange. Besides, the IoF2020 meeting in Prague in March 2019 enabled a follow-up by group work in task forces with respect to specific data models (e.g. in relation to ISOBUS-FMIS data exchange, EPCIS based tracking of events or animal data models relevant in dairy and meat use cases).

At the same time, WP3 was supporting the collaboration with standard development organisations, specifically facilitated by the FIWARE Foundation, as a key non-profit organisation promoting open source towards the realisation of a platform that shall help to digitise processes and collaboration based on standards for breaking information silos as well as helping to make the realisation of IoT based solutions simpler. From the SDO side, specifically the AEF was supporting work on the ISOBUS related matters, GS1 was facilitating work with EPCIS and AgGateway was involved to discuss potentials for facilitating access to relevant data for Farm Management Information Systems. At the same time, project partners were collaborating with the ETSI to progress on the NGSI-LD development.

Therefore, when analysing and discussing the requirements with respect to and open access to data and services, we need to highlight the following requirements:

- ***Interoperability:***
  - From an end-user perspective, interoperability is a kind of universal requirement. In certain areas, like arable farming, it is even a prerequisite for being able to sell farming equipment like tractors, implements and combines as the farmers are expecting of being able to combine farm equipment produced by different manufacturers. Therefore, farmers are not restricted to buy just one brand of equipment required for e.g. tilling, seeding, spraying, or harvesting. This has a direct impact on the technical infrastructures of the equipment manufacturers. New costs are caused by hosting related cloud solutions for connecting farm equipment with the FMIS used or operated by farmers. In addition, mobile devices are often used, while also accepting the usage of smartphones as multi-purpose devices in combination with diverse applications.
  - There are still diverse ICT based solutions used at farms, like e.g. milking robots that are rather closed systems, usually only providing a limited access to the data that could be made available for a potential usage by other systems and solutions. Extracting data is usually costly and in most cases requiring an explicit interaction with the solution providers. This situation is rather frustrating for end-users that expect full access to “their data”.
  - The definition of standards for interfaces does not necessarily result in the development of fully interoperable solutions. Current solutions like the ISOBUS require tangible plugtests to assure full interoperability. While other implementations, like connectors to weather stations or governmental reporting systems are regularly implemented by several stakeholders repeatedly.
  - Standardised data models and related reference data are available to a certain degree, while there are challenges with respect to the access to reference data. This is due to different type of regional implementations (e.g. reporting somehow the same content of pesticide usage data but requiring different forms and interfaces in different regions, states or countries) or limited access to data, caused by organisational groupings of end-users.
  - Work on interoperability is usually discussing interface conditions or specific data models. The usage of standard based open source implementations for reusable components is rather limited. Especially large stakeholders are struggling with competing business models or limited possibilities for harmonising their solutions in the scope of competition law.
- ***Availability of Data:***
  - There is a limited willingness of system providers to share data that is either generated by their solutions or an implicit information/knowledge that characterises their solution. System providers

are afraid of sharing, since they are often fearful or unaware if such data need to be considered as competition-critical data, information or knowledge. Providing such data could jeopardise their competitive advantage or even damage specific business models as well as options on future business models.

- Another challenge is to facilitate the access to available open data that is fragmented in terms of stakeholders maintaining it, the format available as well as the procedure or system offering the tangible access.
- However, end-users have a clear interest on being able to make the most out of their ICT infrastructure, and require new approaches that protect the system providers and enable a maximum exploitation of the systems at hand.

- ***Costs of open platform components:***
  - Depending on the type of end-users, there are quite heterogeneous expectations. There are open platform components that can be used in the development of solutions, particularly the FIWARE Generic Enablers. WP3 has been supporting the use cases in using these components in their individual solutions.
  - Obviously, large stakeholders and solutions providers are able to invest serious amounts of resources in the usage, acquiring, adaptation, evolution or set-up of solutions. However, smaller stakeholders like farmers are usually not able to invest large amounts in complex ICT infrastructures and prefer ideally turn-key solutions.
  - However, also open platforms shall consider requirements to avoid a vendor lock-in and aim at strategies that will facilitate portability of specific solutions on the medium to long-term.
- ***Connectivity in rural areas***
  - This is a classical requirement when implementing ICT based solutions in a rural environment. Usually, the bandwidth for accessing the internet via the available telecommunication providers in rural areas is in no balanced ratio compared to the bandwidth available in urban settings.
  - As a consequence, we could consider the upcoming solutions like LoRa, SigFox and NBIoT as promising potentials to enable at least a certain access by mobile solutions. Nevertheless, the underlying requirements with respect to e.g. edge computing, asynchronous communication as well as minimising the required overhead and payload of messages shall be taken into account.
- ***Accuracy of IoT devices***
  - Several use cases were highlighting the issue of accuracy of IoT devices. For certain solutions, e.g. sensors and communication components need to provide a high accuracy with respect to the measurement results or required latency. The level of accuracy is not necessarily clear in advance, or devices are not keeping initially stated quality characteristics.
  - As the accuracy is crucial for a working final solution as well as prerequisite to maintain a required cost-benefit ratio, such meta-data might be of significant importance when selecting and using related IoT devices. It was discussed on how to assess devices as well as maintain a practical approach. At first hand it is considered as not feasible for IoF2020 to start on such an initiative. However, initiatives of hardware providers could be supported by open platform components on the medium to long-term.
- ***Promotion and support of digital tools at governmental levels in Agri-food***
  - Traditionally, the agri-food domain is promoted and supported to quite some degree by public bodies on a regional, national and EU level. Even global trade agreements can have an impact on how to handle and report information. There are general requirements on how to realise solutions as well as specific requirements on what to report to public entities.
  - The agri-food stakeholders, like farmers, are aiming to use more and more automated procedures to fulfil their administrative reporting obligations. For instance, FMIS are offering features to aggregate data and communicate it accordingly. There are certain initiatives by local public entities to develop individual ICT solutions that are differing from other regions. This usually leads to additional effort for private ICT providers hampering the achievement of a digital single market. Fragmented

initiatives shall be discussed and shall find rather a limited support or alternative solutions that are facilitating portability of solutions.

- ***Ability to coordinate topics of joint industrial interest***
  - This is usually not restricted or limited by a technical perspective. With respect to most technical topics, solution providers could agree on a joint strategy to minimise implementation overhead for the sake of all and agree on strategies on how to serve the end-users in a most appropriate way.
  - However, the competition law is very sensitive on all attempts, which could enable an opportunity to initiate anticompetitive agreements. Therefore, initiatives triggered by IoF2020 need to be highly transparent and open to the different stakeholder communities.

2.3 Usability

The agri-food sector is a domain that requires significant amounts of manual work, tangible interaction with animals as well as sensitive planning and handling of plant growth. As in many business domains, the role of ICT was growing over the years, promising tangible benefits in terms of maximising yield, minimising resource usage, increasing efficiency of the working process as well as optimising quality. Therefore, working places specifically in agriculture are not office centred. Revenues are generated in stables, fields, and greenhouses. The operative usage of ICT solutions is often experienced as a burden rather than a facilitation. This is not necessarily due to the nature of ICT, but possibly a consequence of a bad usability.

This was analysed and discussed within the interaction of WP3 with the use cases, where specifically 365FarmNet was focusing on potential measures for improving usability of IoT based solutions, while also organising related webinars for related use cases. This was organised by 365FarmNet in close collaboration with WP4 that was focusing on the analysis of underlying business models and related key performance indicators, for being able to quantify the solutions' added value.

Therefore, as an aggregation of the discussion and cooperation with the use cases, the team was identifying the following requirements with respect to usability:

- ***Reducing complexity***
  - ICT based solutions shall avoid complex interactions that are not possible without prior knowledge of the end-users.
  - Data entry shall be limited to a minimum and possibilities to acquire implicit knowledge available in existing systems or accessible via IoT devices shall be preferred.
  - End-users shall be able to concentrate on making decisions, having only highlighted situations that require their attention. Usual system behaviour that does not need intervention shall be hidden from end-users as long as not required for the realisation of the process.
  - Focus on the added-value features of a system required to create benefits and revenue. All interactions that are neither a prerequisite nor contribute to the realisation of added-value features shall be revisited for deletion or refinement. Any technical solution that does not solve a problem or support a defined business model shall be avoided.
- ***Value of data versus provision of knowledge***
  - Numerous use cases highlighted their sensitivity with respect to the gathering and accessing data. Tangible sensitivity exists to make data available to a larger stakeholder audience and specifically to competitors as well as customers that might reach an unbalanced position for contract negotiation. At the same time, it is not necessarily clear on how and which knowledge can be generated with the data at hand.
  - Use cases have expressed increased willingness to share data if they are aware of who and how it will be used. IoT based solutions shall facilitate the management of data access or at least making it transparent of who is empowered to access data.
  - At the same time, it shall be carefully analysed, validated and communicated the value laying in the data, if at all, and which value is added by the usage of algorithms generating related knowledge. Usually, shared revenue models shall be taken into account for facilitating the access to data. Use

cases have analysed IoT data and knowledge monetization business models in the scope of WP4 (see D4.9 IoF2020 Use Case Business Models).

– The creation and usage of joint data models, see chapter 5, can help to generate critical mass of data usually required for the sake of validating the outcome of data analysis. Such initiatives can also help to share data of seasonal character to possibly limit the amount of seasonal tests and validation for making new solutions market ready.

- ***Technology push versus pull***
  - As promoted from the very beginning of the IoF2020 project and based on the lean multi-actor approach, the overall objective is to assure an early validation of developed solutions. As soon as a component can be tested in the scope of a user related solution, we shall practically implement an MVP and gather related end-user feedback.
  - The compilation of technical features that are not based on or not matching a specific end-user requirement need to be analysed very carefully in an interdisciplinary team.
  - Due to the developer-centric nature of reusable components, usually we need to combine development and demonstration of reusable components with related IoT based solutions in the use cases.

2.4 Security, Privacy and Trust

The proliferation of "Connected Things" driven by the IoT has prompted the need for "Secure Things" that are critical in preventing sophisticated system attacks. Given the security vulnerabilities of IoF2020 Use Cases, the UC owners are primarily concerned about the safest mechanisms to implement a secure data exchange among several systems and the most appropriate technology for identity management inside the company and the value chain. Based on those use case concerns, the WP3 team was providing a set of best practices to assure security, privacy and trust by design. These best practices were provided to the use cases in the IoF2020 deliverable D3.7, addressing the aspects of process, people and technologies. The support with respect to SPT by design will also be very soon integrated in the IoT Catalogue for assisting users and developers of IoT based solutions to increase their sensitivity about potential threads.

At the same time, WP3 discussed potential requirements for open platform components with the use cases and highlights the following aspects that could be covered accordingly by potentially reusable components.

- ***Authentication and authorisation***
  - The exchange of data in an often mobile and commercial environment requires a sensitive handling of data generated and acquired in the processes. The technical realisation can be considered as a necessity, while diverse technical alternatives are available for its realisation. However, a key challenge is the proper design of the solution that corresponds to the real-world process realisation.
  - A light-weight approach for managing authentication and authorisation is required helping to enable access to distributed services. End-users and developers shall be disburdened from complex implementations that are handling the individual steps for a secure and managed authorisation of access.
- ***Revocation of access rights***
  - There are different scenarios of using IoT based solutions and systems in agri-food. In rather simple scenarios, devices and systems are used and operated by one organisation that is also handling the related data. In other settings, IoT based solutions are handled and used by numerous end-users that are representing diverse stakeholder roles along the agri-food chain. The joint usage of agricultural equipment is one example, while IoT enabled devices to monitor and control the environmental conditions of the produce from farm to fork could be another scenario. Therefore, in any situation where an IoT based solution is handed over from one stakeholder to another, it is required to carefully decide which access rights need to be revoked for the previous owner.
- ***Selective access to data records***
  - IoT based solutions are usually recording a defined set of data based on the presence/ implementation of related sensors. As soon as the access or even ownership of an IoT based solution is

changing from one supply chain stakeholder to another, potentially all available information is accessible. As highlighted before, one could enable a revocation of access rights by previous owners. However, also the access to previously collected data presents a sensitive matter. Therefore, also the selective access to data shall be considered as a requirement. For instance, this can enable the access to the measured temperature curve over the complete life cycle of a produce (i.e. from farm to fork), but only allowing access to the location information for the duration when the IoT based solution was in the hands of the related owner asking for access to the data.

As highlighted before, these are rather requirements that shall be covered by reusable components. For general SPT guidelines with respect to the complete IoT based solutions, use cases shall have a look at Deliverable D3.7.

2.5 Related Development of IoF2020 Reusable Components

The analysis of use case requirements with respect to open platform components was done in close collaboration with WP2 activities. Representatives of WP3 are also direct members of individual use cases and joined meetings of the different IoF2020 trials. At the same time, WP3 was monitoring the progress of use cases w.r.t. the realised IoT based solutions. After analysing the latest progress of the initial 19 use cases in late 2018 and early 2019, representatives of WP3 were also giving feedback and even defining plans for advancing IoT related implementations in the different use cases. At the same time, a team from WP3 was analysing the work plans of the 14 new use cases starting their operation in early 2019. An overview of the latter analysis is provided in deliverable D3.10, outlining the key objectives and IoT related characteristics of those 14 use cases.

In parallel to this, WP3 was also taking care for a structured description of the use cases, representing the results in the IoT catalogue. This shall enable an external and internal stakeholder audience to learn about the IoT based solutions implemented, tested and validated in diverse agri-food settings. At the current moment, the initial 19 use cases are presented in the IoT catalogue. The intention is to also include the additional 14 use cases selected by the IoF2020 open call. On top of that, IoF2020 WP3 is currently working on an administrative interface that will enable other research and innovation projects to add their IoT related use case examples as well as finally opening this to a larger public community to add also those external activities to contribute to the IoT catalogue. This shall help to reach a critical mass of contributions, for having enough traction to facilitate a sustainable offering also after the end of the IoF2020 project. More information is provided in Deliverable D3.12 and, directly by the IoT catalogue, accessible via www.iot-catalogue.com.

Accompanying to WP2 that is coordinating the realisation of IoF2020 use cases, the IoF2020 partners involved in WP3 were structuring their work around the key requirements of the use cases, aiming at the elaboration, enhancement and evolution of reusable components. This is further described in the following chapters, detailing specific components and elements considered of a reusable nature beyond the realisation of a single use case. The motivation for working on the specific topics is explained as following:

- ***Minimal interoperability mechanisms (MIMs):***
  MIMs are representing a common ground that shall help developers as well as end-users to learn about the current state-of-the-art when aiming at the digitisation of agri-food processes as well as facilitating openness and access to data and services. The underlying idea is in principle the motivation to elaborate a blueprint for this openness. As the agri-food business domain is fragmented in diverse sectors and types of stakeholders, it can be rather an orientation than a prescriptive definition of components usage. However, the larger the number of aligned solution providers and agri-food end-users is, the easier it will become to agree upon tangible MIMs. From a strategic perspective, it could highly facilitate solution development by avoiding development overheads, without reducing competition or generating a vendor lock-in. Rather contrary, it would increase the openness of solutions, foster competition, and encourage solution providers to focus on their unique selling propositions, increasing quality, decreasing time to market and finally also aiming at a better cost-benefit ratio for both solution providers as well as end-users.

- ***IoT reusable components:***
  In the initial phase of IoF2020 WP3, all the use cases were analysed in detail and mapped on a reference architecture. This work also characterised the developed elements in the use cases with respect to their potential reusability. When looking at this information in combination with the purpose of the IoT based solutions, the idea is to initiate synergies between the IoF2020 use cases. This is of specific

importance when aiming at a consolidation of work towards open platform components, for avoiding to reinvent the wheel in different WP related initiatives. Extended information is also compiled in the deliverables D3.2 and D3.7 that were elaborated in an earlier stage of the project.

- ***Existing Information Models:***
  The IoF2020 partners are very aware about the tremendous body of knowledge available in the IoT related and agri-food domains. However, when aiming at a consolidation towards open platform components and an open access to data and services, it is of key importance to analyse and remind on specific data models where reuse can add specific value to the agri-food sectors. Therefore, an overview of relevant existing information models can be found in chapter 5.

- ***Open Interoperability:***
  Beyond individual agreements on interfaces, the IoF2020 WP3 is considering open interoperability as key enabler to achieve an open access to data and services. This effort in IoF2020 goes beyond the pure definition of specific data models, but to collaborate in different groups to ensure a critical mass of stakeholders aiming at similar objectives. The work presented in chapter 6 is reflected by the cooperation of the IoF2020 team with the related working groups especially in AEF, AgGateway, AIoTI, ETSI and GS1.

- ***FIWARE Open Platform:***
  IoF2020 WP3 was identifying FIWARE as a key enabler to promote the vision of a system of systems approach. This initiative is touching all three key requirement areas, as the FIWARE related components can have a tangible impact on the openness, the usability and SPT by design. The motivation is to facilitate development in combination with the usage of mature, validated and thoroughly tested software components that comply with the overall system of systems vision.

- ***Farm Management Information System as a Service:***
  FMIS are bringing all information together required for a proper planning and control of farm related activities. It will rather facilitate the daily work of farmers. Therefore, the IoF2020 WP3 team considers FMIS at the heart of usability for classical smallholder farms. 365FarmNet is offering such a system on the basis of a System as a Service, via web-based front-ends. The focus of 365FarmNet in IoF2020 was therefore twofold. Firstly, to work on an FMIS that will be opened to external developers, empowering them of making use of a larger framework of features and services, by being able to integrate their services into 365FarmNet. Secondly, at the same time, 365FarmNet was coordinating tangible support for use cases on usability webinars and one-to-one support, helping for the design of systems that are considered of being straight forward, easy understandable and allowing to access the appropriate information at the right time.

- ***Service and Data Monetization:***
  Finally, the work in the IoF2020 use cases is consolidating in the availability of IoT based services and data. For avoiding a rather limited scope of scenarios, the idea is to promote also the offering and reuse of use case specific services and data. As this requires a basic portfolio of functionalities, WP3 was developing related frameworks for being able to realise related offerings. This is also including the idea of reusing components that were already shaped along the requirements with respect to usability and SPT by design.

On top of that, we would like to highlight that the SPT related analysis of the use cases was continued and is also addressing the new use cases selected via the open call. Further guidelines are also included in Deliverable D3.8, while WP3 will also embed an interactive approach with SPT guidelines and best practices in the IoT catalogue that is available to a public audience.

3 Minimal Interoperability Mechanisms (MIMs) for Digital Farming

3.1 Introduction

In **D3.3** *Opportunities and Barriers in the Present Regulatory Situation for System Development*, we identified eight interoperability points (IOPs). This chapter has reviewed that work towards Minimal Interoperability Mechanisms (MIMs), using a denomination common to the SynchroniCity LSP.

Building on the experience being generated on the field, **D3.2** has established a common architectural view, for each of the UCs, which can be used as a “common ground” to establish IoT- enabled synergies and new added-value services, together with the technologies which can act as the common enablers for the different interoperability points. The figure below shows an updated, generalized architecture (from a functional point of view) of the IoF2020 solutions and the main standards that enable each interoperability point. It leverages the IoT-A Architectural Reference Model. The main layers are:

- *Physical Device Layer*. This layer is composed by different IoT devices and agricultural machinery deployed in the field, that are capable of sensing their environment and generating data of interest for smart farming applications.
- *Connectivity Layer*. This layer enables the transmission of the data produced by devices to upper layers, and vice versa.
- *IoT Service Layer*. The IoT Service Layer exposes the raw data generated from IoT Devices to upper layers in the architecture through different application-level transport protocols based on different paradigms (publish / subscribe, request / response, etc.). In addition, it offers interfaces that allow to communicate with devices for management or actuation purposes.
- *Mediation Layer.* The Mediation Layer transforms the raw data coming from devices or other external services, into curated, harmonized and possibly aggregated data that can be exposed to data processing algorithms or analytics. In addition, this layer is also capable of sending actuation commands to the *IoT Service Layer*. The IoT-A Reference Model subsumes this layer into the “Virtual Entity” and “IoT Services and Resources” layers.
- *Information Management Layer*. The main component of this layer is usually a data hub (which could be incarnated by a Context Broker) which enables the publication, consumption and subscription of all the information relevant to a smart farming solution. The information present at this layer, which can be current or historical, may have been aggregated from different sources, not only IoT. In addition, this layer may offer complex event processing, storage or analytics services, which can generate insights, prescriptions or predictions. The IoT-A Reference Architecture names this layer as “Virtual Entity and Information”.
- *Application Layer.* In this layer resides all the different smart farming applications that could be used by stakeholders, particularly farming professionals. They include, but are not limited to, systems related to decision support (DSS), farm management (FMIS), dashboards or enterprise resource planning (ERP systems).
- A cross-cutting layer on *Security and Privacy* aimed at guaranteeing secure access to information and devices, while respecting the privacy of farmers and exploitations.

In addition, other external entities play a relevant role, namely:

- *Open Data providers*. These could be incarnated, for instance, by databases offering open data in the agricultural domain (pests, disease, weather historical data, ...) or services publishing certain contextual data such as weather forecasts, weather alerts or weather observations. Satellite data/image publication platforms or geo-services which provide geospatial data are also under this scope.
- *Harmonized information models.* They define the structure and representation of the information to be managed. The final aim is that the different smart farming trials share common information models with a view to enabling interoperability and portability of solutions in a wider ecosystem.
- *Public GeoServices.* They offer public geospatial data related to agricultural assets (for instance parcels), frequently coming from geo-information systems owned by public authorities.

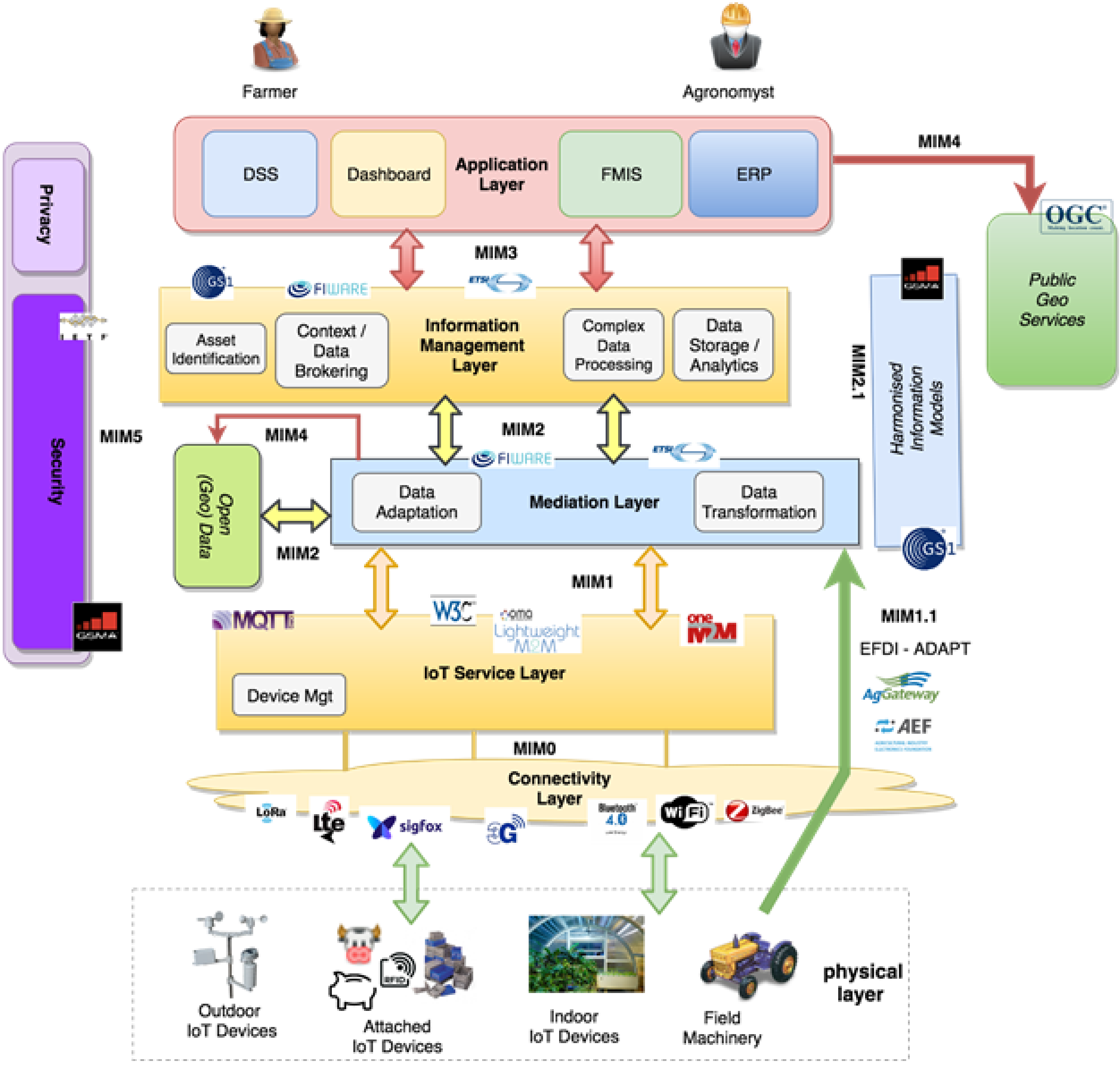


Figure 4: MIMs for Digital Farming Solutions.

3.2 Minimal Interoperability Mechanisms (MIMs) for Digital Farming

The interoperability mechanisms identified and depicted by the figure above are the following:

- *MIM0*: It is realized as a connectivity enabler for IoT Devices and agricultural machinery. Multiple communications technologies can be considered as its basis, including traditional wireless short range (WiFi, Bluetooth, IEEE 802.15.4, ...), Machine to Machine (M2M) powered by global telco networks (3G/4G/5G) or long range IoT networks specifically designed for IoT (**LPWA**).
- *MIM1*: It is situated in between the IoT Service Layer and the Mediation Layer, enabling the exposition of the data and services offered by IoT Devices through well-known programmatic interfaces. **MQTT**, **OMA Lightweight M2M, oneM2M** and **W3C Web of Things** are the main technology enablers available in the industry today.
- *MIM1.1*: It is situated in between the physical machinery (tractors, etc.) and the Mediation Layer, making it possible the bi-directional transmission of data between the agricultural machinery and

the upper layers that deal with information management. **ISOBUS**, **ADAPT**, **EFDI** and ETSI **NGSI-LD**[1] are key enabling and emerging technologies to realize this MIM.

- *MIM2*: It is situated in between the Information Management Layer and the Mediation Layer. On the one hand it enables the transformation, aggregation, harmonization and publication, as context information, of harmonized data coming from IoT Devices, agricultural machinery or other sources of information (open data portals, web services providing contextual data, etc.). On the other hand, it exposes a unified way to send commands and to mediate with IoT Devices or agricultural machinery, regardless the interface exposed by the IoT Service Layer or the Physical Machinery. **FIWARE**, **NGSI-LD** and **GS-1** are key enabling technologies when it comes to MIM2.

- *MIM2.1*: The main enablers of this interoperability point are *Harmonized Information Models* that allow to publish smart farming information following the same meta-model, data representation formats and conventions (units of measurement, etc.). MIM2.1 is key when it comes to portability of solutions at the data layer. **FIWARE Data Models**[2] (a superset of the **GSMA Harmonized Data Models**)[3] and **GS1** are specifications of common information models described by this document.

- *MIM3*: It is situated in between the Application Layer and the Information Management Layer. It is intended to provide access to all the data of interest to smart farming applications, including, but not limited to, real or right time data, historical data or analytics results. In addition, it allows the subscription to data changes and to publish new data coming from the application layer. **FIWARE**, ETSI **NGSI-LD** and **GS-1** are under the scope of MIM3.

- *MIM4*: This interoperability point enables the Application and Mediation Layers to consume public Geo-Services offering open geospatial data, enriching the smart farming applications with geospatial data and off-the-shelf visualizations. OGC **WFS** and **WMS** play a relevant role with regards to MIM4.

- *MIM5*: It is a cross-cutting interoperability point that facilitates the secure interchange of information between the different layers and actors. The **GSMA IoT Security Guidelines** and the traditional security technology stacks (TLS, DTLS, PKI, …) or protocols (particularly **OAuth2**) are under the scope of this interoperability point.

3.3 Integration of interoperable vertical solutions (System of Systems)

The sections above have presented the architectural view and interoperability points of single, vertical solutions intended to solve specific problems, as those posed by the different IoF2020 use case and trials. The Smart agri-food domain is very broad, specialized and diverse. In fact, it is usually needed to count with different solution developers to tackle each farmer's user story, which may involve multiple applications working at the same time. For instance, the technology, devices and applications needed for silo monitoring (sensors based in 3D cameras), is very different than those required for livestock control (collars worn by animals). Also, the graphical user interfaces to be devised are quite different. The latter needs maps and other intensive geospatial data, whereas the former needs a compelling graphical representation of silos and their filling levels.

Therefore, it is envisaged that no single company will be able provide the best solution for all agri-food challenges. Furthermore, there is an opportunity to integrate innovative solutions coming from different parties. It will be based on the integration of information generated by different solutions to build a holistic picture of what is going on a farm. As a consequence, Farm Management Information Systems (FMIS) will be able to provide users an integrated view, encompassing information from different verticals and mashing-up the best of breed user interfaces. In the end, Context Information associated to a farm will be enriched with the contributions coming from different vertical solutions (*System of Systems*), all of them able to share data between each other and enabling a further optimization of processes, saving time, money and resources.

[1] https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf
[2] https://github.com/FIWARE/dataModels
[3] https://github.com/GSMADeveloper/NGSI-LD-Entities

Figure 5 shows a picture of the envisaged architecture of integrated (System of Systems) smart farm solutions, sharing a common (Context) Information Management Layer. At the bottom of the picture there are different Smart agri-food vertical solutions devoted to solve specific problems and which, individually, may have been built using an architecture similar to the one it was depicted by figure 4.

There is also the possibility that vertical solutions could have been developed using proprietary approaches, i.e. custom APIs and data models. In that case an adaptor in the Mediation Layer would be needed. Such an adaptor will serve as a bridge between harmonized Information Management APIs and data models of IoF2020 and vertical-specific, proprietary artefacts.

The main layer in the referred picture is the Context Information Management Layer, enabled by *NGSI-LD and Harmonized Data Models*. It will expose (partially or totally) to the FMIS all the different Entities, Properties and Relationships that capture what is going on in the farm concerning the different verticals involved. Furthermore, each vertical could benefit from information generated by other integrated verticals, yielding to synergies that can bring new applications or insights to end users (farmers, agronomists, etc.). End users will get access to an integrated view of farming processes from a single stop-point, the FMIS. Such system, could offer, globally, different integrated applications, both horizontal or vertical (from the different vertical providers) such as alert management, dashboards, advanced map representations, analytics (prescriptive, predictive or descriptive), KPI monitoring or data mash-ups.

In addition, vertical solutions could also export their data to an Open Data Marketplace, or even benefit from existing data present in the marketplace. The available open data could also be exploited directly by FMIS. Open data could also be made available through standard Geo-services, adapted and mashed-up as Context Information or being consumed directly by ad-hoc adaptors.

This System of Systems approach is described in more detail in Deliverable **D3.8**.

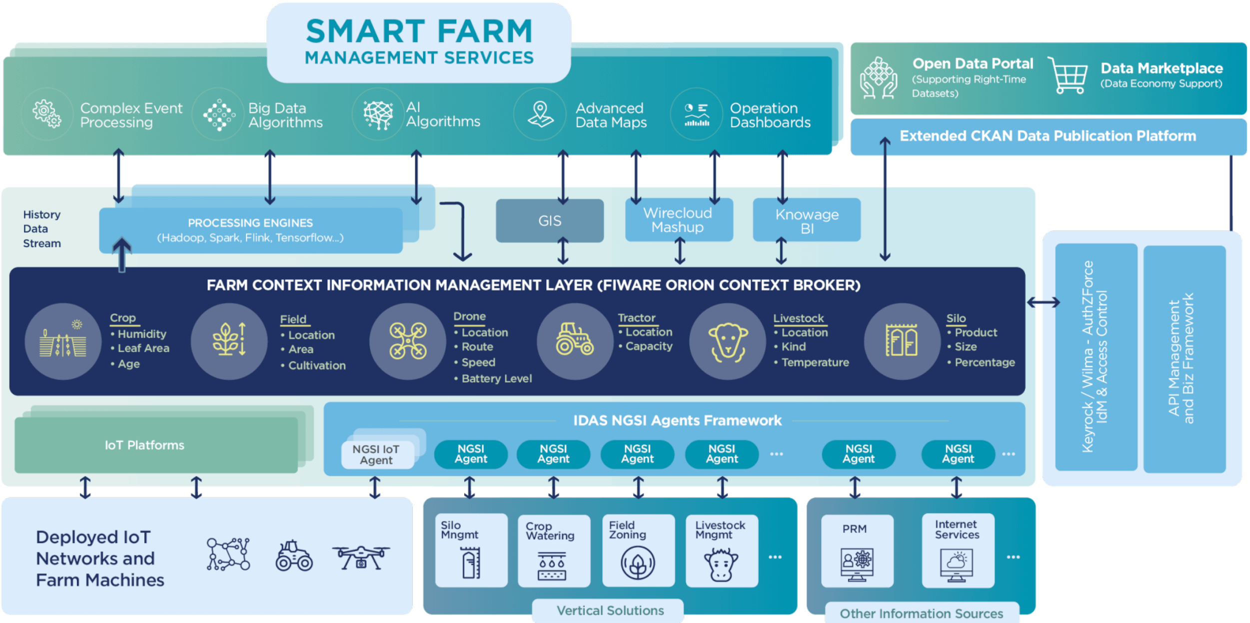


Figure 5: System of systems view for IoF2020 integrated Smart agri-food solutions

3.4 Harmonized Data Models as a key MIM

The availability of shared, well-adopted information models is a key interoperability mechanism for enabling a global market for IoT-enabled digital farming services. Such models provide an essential element in the common technical ground needed, by making replicability and portability of smart agri-food solutions practical. This report describes a simple framework for defining new information models in agri-food, taking a multiple baseline approach based on and extending the Minimal Interoperability Mechanisms (MIMs).

In particular with regards to data management, IoF2020 requirements state that the system has to support and reuse open data and metadata by providing pre-built taxonomies to describe different assets. The final aim is to facilitate replicability of solutions across agri-food subdomains and trials, while at the same time avoiding vendor lock-in.

MIM2.1 must be based on an information meta-model through which concrete, domain-specific data models can be expressed in a coherent way across domains. This allows for sector-specific focus in a procurement or development process, while maintaining cross-domain consistency. In addition to the information meta-model, our work provides recommended design guidelines, documentation guidelines and compliance rules.

Thus, data models play a crucial role because they define the harmonised representation formats and semantics that will be used both by agri-food applications to consume data (through the northbound interfaces) and by data sources at the southbound (sensors, existing information systems or open databases) to publish data. Furthermore, data models are one of the key “interoperability points”, enabling the participation in a digital single market. The implementation of the IoF2020 Data Models (as described by Chapter 5), together with other interoperability points will ensure that each of the trials and use cases can deploy interoperable IoT-enabled digital farming services.

4 Overview of IoT reusable components

WP3 has analysed the initial 19 use cases of IoF2020, representing a functional view of their respective IoT solutions in an IoT architecture reference model (see deliverable *D3.7 Compilation of Use Case Requirements*). This basic functional view was used to structure the use case specific key functionalities required for the envisaged IoT solutions. The objective was to identify key functional components, trying to reduce complexity, and conclude on reusable components used in several use cases.

WP3 in IoF2020 aims at identifying components of IoT based solutions that are reusable and facilitate the development of IoT based solutions. In order to achieve this, we used an IoT reference architecture (Figure 4) to map reusable IoT system components to an agreed approach, to facilitate the presentation and communication of project results and ease the assessment of reusability potentials by different stakeholders. All the use cases' solutions were represented using a unique functional view of IoT Architecture Reference Model, and the components were classified as:

- General commercial off-the-shelf (COTS) software or hardware components integrated as individual parts in the overall solution deployed in the use case. These might be interchangeable, while variants of different type or from different suppliers might provide similar features, but can deviate in terms of quality characteristics (e.g. accuracy, performance, size, costs). Such components are usually available from external parties, commercially offering their solutions at specific conditions.
- Components or also (minimum viable) products developed and implemented in the use cases themselves. These are either major functional components or use case related solutions that can also be customised or configured for use cases of other stakeholders from the agri-food domain or other verticals. The maturity and validation might be in an initial state and further effort might be required for a proper exploitation of such components.
- Reusable components not yet available for the IoF2020 use cases that would facilitate the realisation of the envisaged IoT based solutions. At the same time, they can facilitate the exploitation of the envisaged project results as well as increase the impact of the envisaged solutions with respect to non-functional requirements like e.g. adaptability, costs, robustness, maturity or usability.

As stated, deliverable D3.7 Compilation of Use Case Requirements included the functional view of the 19 initial use cases of the project, concluding with an overview of use case related reusable components, MVPs and additional developments. These additional developments were also represented in their own functional view of the IoT Architecture Reference Model, analysed and prioritised, to select a set of reusable components to be developed by WP3 and offered to all use cases as support for their IoT solutions.

This chapter extends this analysis to the 14 new use cases in the project, providing an update on the overview of use case related reusable components, MVPs and additional developments. The objective is to conclude whether the current offer of reusable components of WP3 is still valid or is missing some key component required by the use cases.

4.1 Overview of use case related reusable components, MVPs and additional developments

The following Table 1 provides an overview of all the 33 IoF2020 use cases with respect to their current development perspective and potentials for reusable and replicable components for a sustainable usage by the IoF2020 partners themselves as well as by a larger target audience.

Table 1: Summary of reusable components, useful additional developments and product perspective of the IoF2020 use cases.

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC1.1 Within-Field Management Zoning	• 365FarmNet • Information model • ISOBUS solution	• Configurable Dashboard • IoT Data Edge Consolidation • Identity Management • Authorisation, Authentication Cross system reuse of decision support modules/services could be supported by: • Service Monetization	• Soil map service • Variable rate application map service • Automation & Machine Communication • Modelling services
UC1.2 Precision Crop Management	• Orion Context Broker • Bosch IoT System • ImagesBank • Data model • Long term data storage • Data processing • Data transformation • IoT Broker	• Configurable Dashboard • IoT Data Edge Consolidation • Identity Management • Authorisation, Authentication	• Crop Monitoring Dashboard • Water & Nitrogen Manager
UC1.3 Soya Protein Management		• Cloud Data Storage The interoperability challenge could also be addressed by facilitating the context related provision of data, based on a standardised functionality offered by: • Context Broker • Data Marketplace	• Soya Production Advisor • European Premium Soya Label • Fusion Service Engine

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC1.4 **Farm Machine Interoperability**	• Sensors dashboards • Mobile app • Live Data Dashboard • Big Data Storage • Stream Processing • Batch Processing • Local Data Storage • Automation & Machine Communication	The nature of UC1.4 and its envisaged replication on a large number of farms operating diverse legacy systems, might be facilitated by making use of • Context Broker • Data Marketplace	• IoFieldGateway
UC1.5 **Potato Data Processing Exchange**	• Sensors • Aurea Imaging • AVR • IoT platform	• Configurable Dashboard • IoT Data Edge Consolidation • Identity Management • Authorisation, Authentication	• AVR Conect (web-based dashboards) • Dronewerkers (Aurea Imaging) • Farm Frites
UC1.6 **Data-Driven Potato Production**	• Gaiatron • Information Management Adapter • FIWARE CKAN • FIWARE Identity Management Access Control • Cygnus	• Configurable Dashboard • IoT Data Edge Consolidation • Data marketplace	• GAIA cloud and Gaiasense API • SF Application UI
UC1.7 **Traceability for Food and Feed Logistics**	• Silo tags • Silo IoT device • Trailer base station • Trailer PLC system • Wireless reader	• Context Broker • Identity Management • Authorisation, Authentication	• IoTrailer application • WiFi track & trace • Central control system
UC1.8 **Solar-powered Field Sensors**	• ML algorithms • Cloud data storage • AgriModule gateway • Sensors	• Context Broker • IoT Data Edge Consolidation • Identity Management • Authorisation, Authentication	• SolarVibes mobile application • Dashboard

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC1.9 Within-Field Management Zone Baltics	• AgroSmart platform • SpectroRadiometer • Hyperspectral Imager • Sensors	• Context Broker • IoT Data Edge Consolidation • Data marketplace • Service monetization	• FMIS ERP digital maps
UC2.1 Grazing Cow Monitor	• Sensolus tracking system (location and activity data)	Provision of pasture management data can be facilitated by: • Context Broker • Data Marketplace	• Cow Grazing Monitor
UC2.2 Happy Cow	• MEMS sensors: to monitor individual/group movements of cows. • Sub-1GHz RF radio: 6LoWPAN or Sig-Fox radio for each cow collar. • Bluetooth Low Energy: local interaction with smart phone app. • Farm Gateways: running integration/ communication layers. • Cloud service: for application hosting.	• Identity Management • Authorisation, Authentication	• Ida: Intelligent Dairy Farmers Assistant
UC2.3 Silent Herdsman	• Silent Herdsman: collars placed on the cow's neck to allow identifying them, which can be deployed on other farms	• Context Broker • Data Marketplace	• Herdsman+
UC2.4 Remote Milk Quality		The provision of reference data, anonymised average quality data and possibly also individual data could be provided by: • Data Marketplace	• Remote Dairy Quality Tools
UC2.5 Lameness Detection through Machine Learning	• IBM Watson IoT platform • Pedometer • Collar	• Context Broker • Data Marketplace • Service monetization	• Dashboard • Mobile application • Fog Node
UC2.6 Precision Mineral Supplementation	• Big data collection • Ear tags • Mineral feeder	• Context Broker • Identity Management • Autorisation, Authentication	• Pitstop+ Manager • Integrated Administration and Control System • Master unit

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC2.7 Smart Precision Cow and Cattle Monitoring	• Smart Rumen Bolus • Asset management	• Identity Management • Authorisation, Authentication • Service Monetization	• Web and mobile applications • Business logic algorithms • Complex Data processing and analytics
UC3.1 Fresh Table Grapes Chain	• BluLeaf DSS: commercial DSS solution for irrigation and fertilization management, which is suitable for every farm that requires a DSS to support their daily activities. • Agriculture analysis and control portal: web portal for viewing/analysing sensor data, specifying algorithms/models and automatic actuator control.	• Identity Management • Authorisation, Authentication Cross system reuse of decision support modules/services could be supported by: • Service Monetization	• Adaptive DSS • Kc Calculation • Blow
UC3.2 Big Wine Optimization	• STM32L0 Discovery kit LoRa, low-power wireless (B-L072Z-LRWAN1): LoRA connectivity module to connect any sensor/actuator to a LoRA network. • ST TESEO III of GNSS: Global Navigation Satellite System. • Lorrier-lr2-iot-lora-gateway 868 MHz: LoRA Gateway. • sensiNact ioT platform: IoT platform. • Process2Wine: Web and Mobile Software solution for the wine production.		• Process2Wine IoT: precision viticulture and winemaking supervision; • NIR handy spectrometer: advanced we-based wine analysis; and • JODYN Modem: NET test for wine shipping monitoring
UC3.3 Automated Olive Chain	• Orion Context Broker • IoT Agents	Provision of quality and traceability data could be provided by: • Data Marketplace	• Olive Production Manager • Olive Oil Quality Manager
UC3.4 Intelligent Fruit Logistics	• IoT Platform Vizix • IoT chip • EPS RTI • Mobile Scanner • EPCIS	Provision of added-value services for customers along the supply chain can be facilitated by • Service Monetization	• EPS Tray Radar • EPS Temperature Monitor

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC3.5 Smart Orchard Spray Application	• Mobile scan application • Context broker • Local data storage • Authentication and control	• Service Monetization	• FEDE Cloud • FEDE clients application • EPS staff reporting application
UC3.6 Beverage Integrity Tracking	• Sensors • Device management • Authentication & Authorisation • Cloud data storage	• Context broker • IoT Data Edge Consolidation • Data Marketplace • Service Monetization	• Mobile application • Track & Trace • Prescriptive Analysis Dashboard •
UC4.1 City Farming Leafy Vegetables		Provision of produce related quality data might be facilitated by • Data Marketplace	• GrowWise
UC4.2 Chain-Integrated Greenhouse Production	• OpenStack • FIWARE Engine	Provision of monitored data that might facilitate decisions in other systems might be provided via a • Data Marketplace	• Greenhouse DSS and OPtimization
UC4.3 Added Value Weeding Data	• Camera Data Processing Algorithm • Harvest yield monitor • Crop Growth Web Service • Yield Prediction Web Service		• CropMap
UC4.4 Enhanced Quality Certification System	• IoT Agents • Cygnus	• Identity Management • Authorisation, Authentication	• Enhanced audit • E-loyalty • E-wine making
UC4.5 Digital Ecosystem Utilization	• Cygnus • Context broker • IoT Agent • Sensors • Irrigation Controllers	• Data marketplace • Service Monetization	• Mobile application • Web dashboard

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC5.1 **Pig Farm Management**	• Water consumption sensor • RFID sensor • RFID tag • Farm and slaughterhouse Gateways • Cloud Service	• Identity Management • Authorisation, Authentication To facilitate provision of chain level data, potential usage of: • Data Marketplace	• Smart Pig Production
UC5.2 **Poultry Chain Management**	• Animal scale • Silo scale • Water consumption sensor • Environmental sensor node • Smart bracelets • Integral farm controller • Farm and slaughterhouse gateway • Slaughterhouse DB • Slaughterhouse Gateway • FIWARE Cloud Services • Cloud Services	To facilitate provision of e.g. animal welfare monitoring data, one can use: • Data Marketplace	• Poultry Growth & Health Manager • Transport & Logistics Monitor • Poultry Chain Manager
UC5.3 **Meat Transparency and Traceability**	• EPCIS • EPCIS Capture connector • FIWARE Adaptor to EPCIS (Orion Con-text Broker)	Adaptation towards other farming domains	• MITS: Meat Information Transparency System • Tool for proactive automated auditing • KPI and benchmark dashboard
UC5.4 **Decision Making Optimization in Beef Supply**	• Context broker • Cloud data storage • IoT Agent UL • IoT smart collars • IoT ear tags • Blockchain • Weather station	• Data Marketplace • Service Monetization	• Decision support system • Dashboard • Multi-platform application • AI algorithms

Use Case	Re-usable Components	(Potential) Additional Developments	(Minimum Viable) Products
UC5.5 Feed Supply Chain Management	• Cygnus • Context broker • Sensors • Cloud data storage • CKAN	• Authorisation, Authentication • Configurable Dashboard • Data marketplace	• INSYLO Hub • Smart Feed Logistics Platform • Mobile and web applications
UC5.6 Interoperable Pig Tracking	• Context broker • Cygnus • Ear sensors	• Identity Management • Authorisation, Authentication • Configurable Dashboard • Service Monetization	• Dashboard • Multi-platform applications • IoT platform

4.2 Overview of potential developments related to WP3

Based on the results presented in Table 1, and trial-level discussions, related reusable components are presented in Figure 6. These components remain unchanged from those identified previously in *D3.7 Compilation of Use Case Requirements*. It is relevant to highlight the increased interest of initial, and particular the new open call use cases, in using some key FIWARE components as the context broker. This increased interest is well aligned with the system-of-systems approach proposed within WP3, which can be tested and applied to a wider range of use cases. In addition, there is a wide potential for the components of service monetization, data marketplace and configurable dashboards in several use cases.

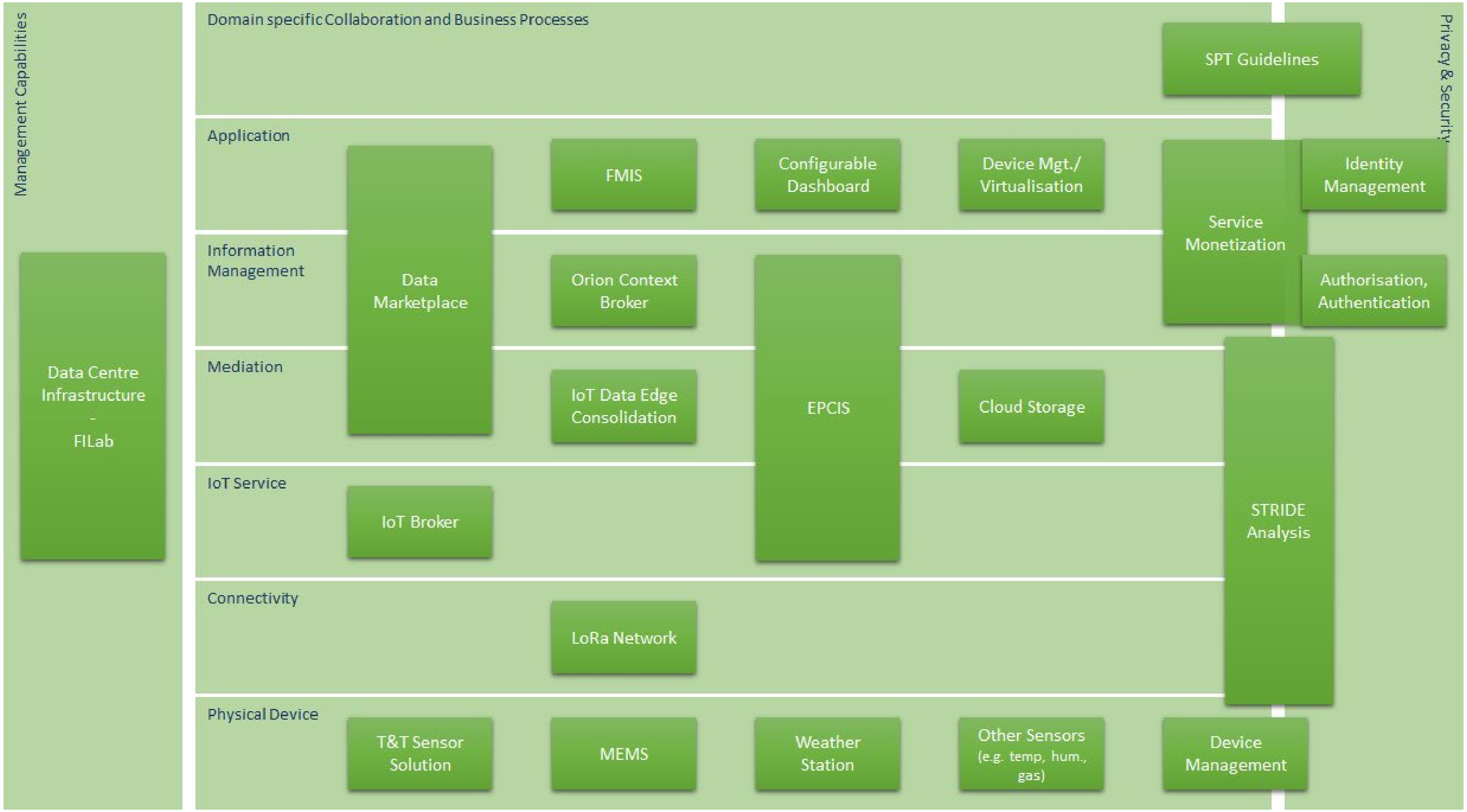


Figure 6: Potential reusable components to be further analysed, realised and potentially hosted with direct support of WP3

5 Overview of existing Data Models

5.1 Introduction

As stated before, the IoF2020 project comprises 33 different use cases that probe to the potential of IoT (Internet of Things) in the agri-food sector. The use cases were selected on their potential of scalability for large scale pilots with a specific prerequisite for technical readiness level.

From a technical perspective, a key aspect of the project is finding synergies between the use cases. This chapter presents mainly the work done as a follow up on a previous deliverable **D3.9** Synergy Analysis of the Use Cases. It is built on the results and recommendations of the mentioned deliverable. Summarized D3.9 proposed three synergy actions:

- Draft shared information model fragments for sensor data;
- Include the synergy analysis results in the IoT Catalogue;
- Organize short, lightweight task forces.

Objective of this investigation is to assess the proposed synergy actions as a follow up. Moreover, this chapter shares results of several actions designed to gain insights on information modelling (as a practice), current status of relevant information models within the Agri IoT domain, and identifying a consensus in an approach for information modelling in Agri IoT.

The IoF2020 project adopted a lean multi-actor approach having several focus areas like engagement of different stakeholders, acceptability and development of sustainable business models. For this study, qualitative research design was adopted to provide insights in possible information model fragments for sensor data within Agri IoT. The methodological approach taken in this study is a mixed methodology based on case study analysis, workshops and desk research.

When information modelling for an application of Internet of Things technology in the agri-food sector, several standards and models are available that can be fully or partly adopted. This chapter presents the results of exploratory desk research into literature on, standards for and case studies of the application of internet of things technology in agri-food context.

The results are presented in three sections, covering the different areas or perspectives found when studying relevant literature and standards:

- The agri-food perspective where domain concepts such as Crop, Field and Yield are central in modelling;
- The IoT perspective, where concepts such as Sensor and Actuators are central;
- The Predictive techniques perspective, where modelling Predictions and their associated Likelihoods and Confidence levels are central.

Additionally, from a cross-domain perspective, several standards are introduced. In particular, EPCIS (Electronic Product Code Information Services) and FIWARE are analysed and presented. Finally, a guideline is proposed as a summary for information modelling professionals.

5.2 Information Models for the Agri-Food Domain

The agri-food domain has a long tradition of data exchange and, consequently, quite a few generic agri-food information models are available. We have selected and analysed the models and standards in Table 2 below.

Table 2: Selected and analysed data models and standards.

Standard	Maintainer/Source
GSMA / FIWARE Agri Models	GSMA / FIWARE Foundation https://github.com/GSMADeveloper/NGSI-LD-Entities
ADAPT	AgGateway https://adaptframework.org/
rmAgro	Wageningen University & Research ftp://pragmaas.com/rmCrop/rmAgro_SNAPSHOT/
ISO 11783-11 / ISOBUS	ISO / VDMA https://www.isobus.net/isobus/
AgroVoc	FAO http://aims.fao.org/vest-registry/vocabularies/agrovoc
AgroXML	KTBL http:/www.agroxml.de
E-Crop	UNCEFACT https://www.unece.org/fileadmin/DAM/cefact/brs/BRS_eC-ROP_v1.pdf
SAREF-4-AGRI	ETSI / TNO https://mariapoveda.github.io/saref-ext/OnTool-ogy/SAREF4AGRI/ontology/saref4agri.ttl/documenta-tion/index-en.html
DAPLOS	UN/EDIFACT http://www.unece.org/trade/untdid/d08a/trmd/daplos_c.htm

5.2.1 Common Abstractions

Many of these models propose detailed representations of the concepts relevant within agri-food. Models may for instance contain many variants of a single intuitive concept to account for the different roles and contexts it is used in. To facilitate the adoption and reuse of the models, we have compared these models and outlined them using some common abstractions.

A common abstraction that many of the models implicitly or explicitly share is that of inputs, processes and outputs.

Inputs – The use of resources such as for instance products, land and labor used in an agri-food use case.

Processes – The processes that take place within the use case. These include the biological processes that take place, the agricultural operations performed and also tactical and strategical planning pro-cesses.

Outputs – The outcomes of the processes, both produce/yield, as secondary outputs such as pollution, emissions and waste.

These concepts can be recognized in many of the IoF2020 use cases. Most use cases in some way inter-vene in a process loop or control cycle. Some intervene at the level of the main production loop where resources and other inputs are turned into products, through biological processes, agri operations and planning. Other use cases intervene in smaller, operation specific loops such as smart weeding or smart irrigation. Also, many KPI's map to these concepts very well. They often can be abstracted to:

- Reduction of input consumption,
- By optimizing biological processes, agricultural operations and planning,
- Leading to output maximization (i.e. Yield) and/or minimization (i.e. Pollution, Waste).

A second common abstraction contains a high-level grouping of concepts recognized in the different existing models into a number of central concepts that operationalize the input, processes and output model.

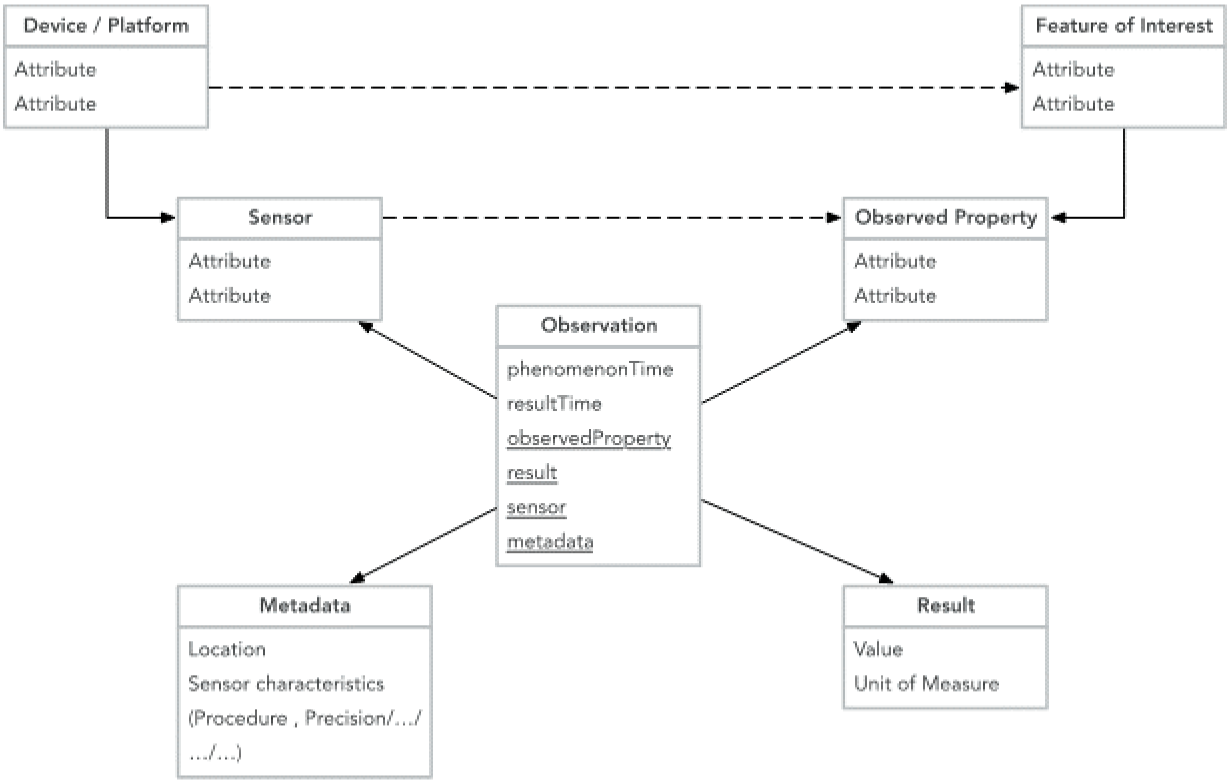


Figure 7: High level abstraction of Agri Information Models

Table 4 below provides explanation of concepts that map to the different high level concepts in the figure above. This model is by no means complete, but may help to find one's way in the (sometimes very large) existing models mentioned. In section 5.2.2, we make a more detailed analysis of how the existing models map onto this outline for the arable domain.

Table 3: Concepts that map to high level abstraction of agri information models.

Concept	Examples
Biophysical	Crop & Animal related concepts, e.g.: • Species and variety • Growing stage • Height Leafiness
Field	Geographical properties Soil type Soil moisture

Concept	Examples
Biological Processes	Crop Growth Evaporation Weed Pressure Diseases
Agricultural Operations	Pest Control • Weeding • Herbicide Fertilization Harvesting
Tactical Processes	Planning QA Job > Task > Operation
Yield Quality and Quantity	Yield per m2 Protein levels
Resource Usage and Waste	Water Energy Labour time Fertilizer Harvesting losses Decay in storage
Biophysical	Crop & Animal related concepts, e.g.: • Species and variety • Growing stage • Height • Leafiness

5.2.2 Detailed Analysis of the Arable Domain

There is a large number of interchange formats, ontologies and other modelling efforts to create canonical models of agri-food related information. The standards and models reviewed in this study are listed in Table 4. In order to have a desired quality of in-depth analysis, the scope in the detailed analysis has been the Arable domain.

Table 4: Standards and data models reviewed.

Standard	Field	Building	Weather	Diseases	Input product	Equipment	Labour	Biophysical -Animal	Biophysical-Crop	Operations	Planning	Produce	Used Resources and Waste	Emissions
FIWARE /GSMA Agri Models	AgriParcel	Building	Weather Forecast Weather Observed	Agri Pest	Product Agri Soil	Fleet Vehicle UAV Device Machine			Agri Crop	Agri Parcel Operation Device Operation Fleet Vehicle Operation Machine Operation		Agri Product	Product	Product
ADAPT	Field	Facility	Observations	Observations	CropProtection-Product CropNutrition-Product CropVarietyProduct	DeviceElement Implement Equipment Machine	Person		Crop Product Product - Status: ProductStatusEnum	WorkItemOperation	Plan (WorkItemIds)	Harvested-CommodityProduct	Product --> ActiveIngredient	Product ProductFormEnumeration: Gas
rmAgro	CropField	Building	Meteorological-Variable Predicted-Weather	Pathogen Weedspecies PestSpecies	PlantProtection-Product Fertilizer SeedAndPropagationMaterial	Equipment	Worker	Animal	CropClass	Operation	Scheduling	Produce	Product	Product
AgroXML	Field		ClimateType		OperatingSupplies		Employee		Crop Crop-GrowthStage	WorkProcess		Harvest		

5.3 Information Models for the Internet of Things Domain

The previous sections focused on existing information models about the primary domain of application, the agri food domain in our case. We have also found existing information models that have been developed with the internet of things as their domain.

These models choose as their domain of modelling the domain of connected things, their sensors and actuators, the measurements they produce and actions they perform. As such, these models are neutral with respect to the primary domain that IoT is used in, and are intended to be equally usable in other domains, such as for instance smart cities, mobility and health.

Based on our exploratory literature and web search, we have found several existing information models, standards and API's for IoT related information modelling. They are listed in Table 5 below.

Table 5: Highlighted information models, standards and API's for IoT related information modelling.

Standard	Maintainer/Source
W3C OGC Semantic Sensor Network Ontology • SSNX (2010) • W3C Rec (2017-10)	World Wide Web Consortium (W3C) Open Geospatial Consortium (OGC) https://www.w3.org/TR/vocab-ssn/
OGC Observations and Measurements *Widely used standard for capturing observations and measurements.*	Open Geospatial Consortium https://www.opengeospatial.org/standards/om
OGC SensorThings API • Sensing Profile • Tasking Profile	Open Geospatial Consortium https://www.opengeospatial.org/standards/sensorthings
Mozilla WebThings API	Mozilla Foundation https://iot.mozilla.org/wot/
W3C Web of Things[4]	https://www.w3.org/TR/wot-architecture/ https://www.w3.org/TR/wot-thing-description/
EC Project IoT-A Reference Architecture (D1.5)	IoT-A Project (EC FP7) https://cordis.europa.eu/project/rcn/95713/ Project website offline
A Linked-data Model for Semantic Sensor Streams *A research model into capturing sensor streams. Not a standard per se. Main differentiator seems to be mixed precision dealing with location metadata.*	Barnaghi, P., Wang, W., Dong, L., & Wang, C. (2013, August). A linked-data model for semantic sensor streams. In *2013 IEEE International Conference on Green Computing and Communications* (pp. 468-475).
Smart Appliances Reference Ontology (SAREF) *A generic ontology for smart appliances, with an emphasis on appliances found in smart buildings.*	TNO http://ontology.tno.nl/saref/

[4] Further described in paragraph 6.2.1

Standard	Maintainer/Source
Prov-o *A generic ontology to capture the provenance of linked data. Informally stated, to capture how data was produced and processed, and by who.*	World Wide Web Consortium https://www.w3.org/TR/prov-o/ Compton, M., Corsar, D., & Taylor, K. (2014, October). Sensor Data Provenance: SSNO and PROV-O Together At Last. In *TC/SSN@ ISWC* (pp. 67-82).
Quantity, Unit, Dimension and Type (QUDT)	QUDT.org, originally developed at NASA http://www.qudt.org/release2/qudt-catalog.html

5.3.1 A Common Abstraction: The IoT Control Loop

Almost all of these standards implicitly or explicitly use as a common abstraction the concepts from the typical IoT feedback or control loop. This loop was also used in D3.9 to categorize the use of IoT technology by the use cases. The loop in the figure below was slightly adapted from the model used in D3.9 based on the literature found.

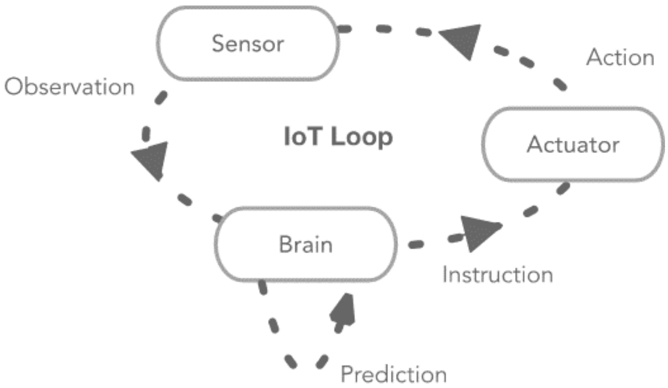


Figure 8: Internet of Things Control Loop (Adapted from D3.9)

The common abstraction for the different roles IoT plays in a typical control cycle consists of sensors, a brain and actuators.

- Sensors provide the observations from the outside world to the system;
- A brain applies logic to those facts to draw conclusions or make predictions to assert additional facts and determine next steps in the form of instructions for actuators;
- Based on these instructions, actuators may interact with the outside world to influence it;
- These effects are then observed by the sensors, completing the control or feedback cycle.

In section 5.3, we discuss the modelling aspects of sensors and actuators. The modelling aspects of intelligent behaviour is deferred to the next section, where we discuss the information modelling aspects of applying predictive techniques and machine learning in general.

5.3.2 Modelling Sensor Information

Most of the IoT oriented information models we found support some aspects of sensing. We have identified three aspects of sensing that every use case of IoT has to deal with and compared how they are supported across the standards and models.

Decoupling of devices and feature of interest - In many IoT cases, and especially in agri-food, the sensors deployed as part of a connected device do not measure properties of that device but of some other physical object. This is called the feature of interest in many standards in this area. For example: a cow collar may be registered in the IoT infrastructure as device with properties, but most of these properties actually apply to the cow wearing it. A choice has to be made whether (for instance) the FMIS records the device ID with each cow, or that the IoT infrastructure stores the cow ID with each device. This choice has consequences for data interoperability.

Sensor History – How to deal with historical measurements requires choices as well. History can be organized around the sensors and devices, and a number of IoT standards provide support for capturing observations in the past. In practice, devices and sensors are replaced or moved between animals/fields. Depending on a sensor ID alone to look up history therefore is not enough. The sensing data is typically used in some primary process application, and this application may have its own form of history.

Sensor Metadata – Almost all standards relating to sensing also specify ways of capturing sensor metadata. These are characteristics of the observation in addition the actual sensed value. Examples are time, location and properties of the sensing technology such as precision, resolution etc. These attributes can be important for a consumer to establish whether the observed value is precise or trustworthy enough for its particular use.

The next table captures on a high level how the different information models we found deal with these three modelling challenges.

Table 6: Ability of identified information models to handle modelling challenges.

Standard	Device vs FeatureOfInterest	Sensor History	Sensor Metadata
W3C en OGC Semantic Sensor Network Ontology SSNX (2010) W3C Rec (2017-10)	SensingDevice vs FeatureOfInterest	Multiple Observation instances	Procedure resultTime phenomenonTime Precision, Resolution, Selectivity. Sensitivity etc.
OGC Observations and Measurements (O&M)	featureOfInterest, observedProperty	Time Series Profile	procedure, resultTime phenomenonTime Sampling
OGC SensorThings API Sensing Profile	FeatureOfInterest	DataStream	Sensor (=Type)
Mozilla WebThing API W3C Web of Things	No support, Device only	No support	type, unit, min, max, multipleOf
EC Project IoT-A Reference Architecture (D1.5)			
Smart Appliances Reference Ontology (SAREF)	Property > Measurement < Sensor	-	Timestamp, unit of measure
Prov-o	-	The research on combining SSN and Prov-o creates extended opportunities to capture sensor data provenance: How was the data produced and processed, and by who? Entity > Activity < Agent	
Quantity, Unit, Dimension and Type (QUDT)	-	-	Provides generic concepts for dealing with quantities and units of measure, alternative to many of the concepts of the O&M standard.

The different sensing standards mentioned above can be summarized as illustrated in Figure 9 below.

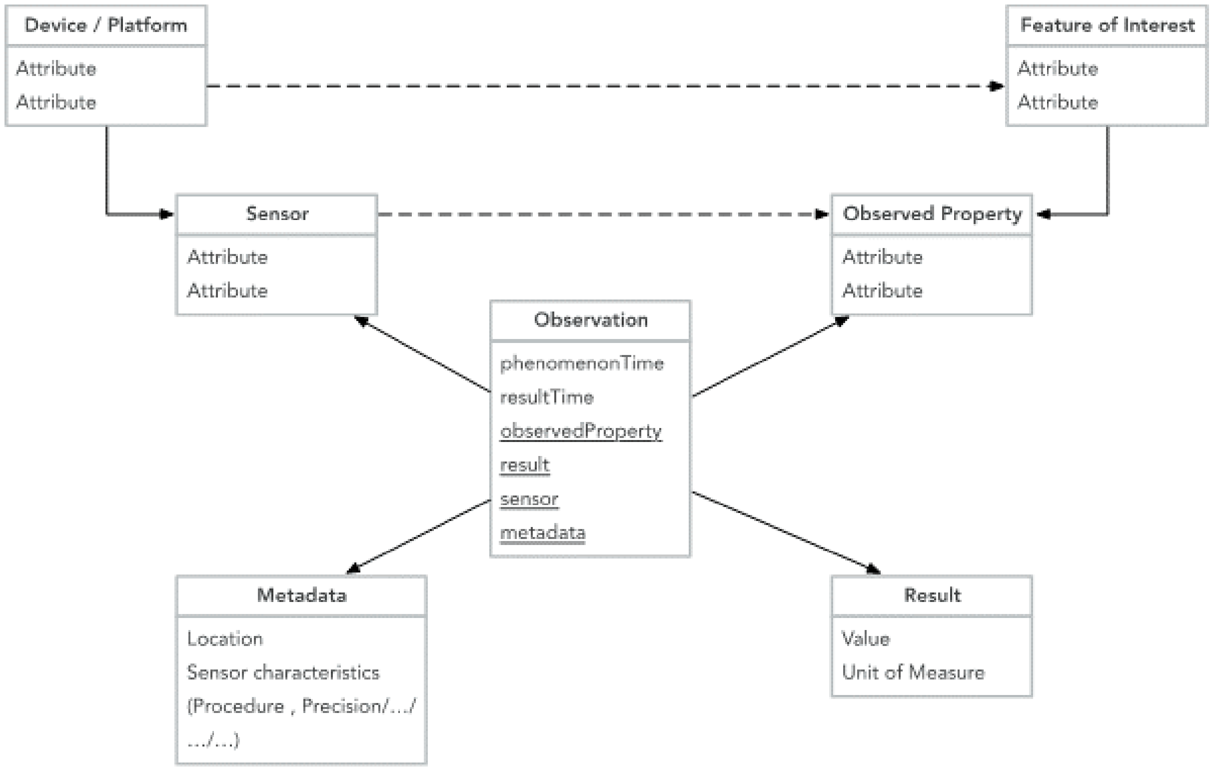


Figure 9: A high level summarization of existing IoT information models

5.3.3 Modelling Actuator Information

In D3.9, we shared the observation that there are two scenarios in terms of actuation in IoT.

Actuation as a means of Remote Control – Often, connected devices and equipment expose their actuators for remote control. In this scenario, agri-food professionals decide how and when to operate the equipment. The agri-food professionals may or may not be supported in making operational decisions based on sensor information from those same devices.

Actuation as part of Autonomous Behaviour – Autonomous devices and equipment have an internal, full control cycle. Based on sensor inputs and prediction models or algorithms, actuation is automatically initiated. Examples include automatic water management and fertilization in greenhouses or smart weeding in IoF2020 use cases.

In both scenarios, information models are needed that capture actuation instructions. In the RC scenario, these instructions are sent to the device by professionals through an FMIS for instance. In the autonomous scenario, devices need to expose the actuation actions they autonomously perform to external systems for auditing and evaluation purposes. In the following table, we capture how the different standards support these two scenarios.

As many of the information models focus on sensing only, those models are omitted here.

	Remote Actuation	Autonomous Actuation
OGC SensorThings API **Tasking Profile**	Tasking Profile Task > Actuator	Actuation at a device can be monitored through MQTT subscription on Task topic.
Mozilla WebThing API	Send Actions	Actuation at a device can be monitored through Websocket

		connection listening to Action, Property and Event messages.
Smart Appliances Reference Ontology (SAREF)	Being a reference ontology, SAREF does not support an API for performing either scenario. The model does support capturing the concepts needed to store the information used in both. Concepts: Actuator, Task, State	

5.4 Information Models for Statistical Processing / Machine Learning

We have searched for models in a third domain. As observed in D3.9, many IoF2020 Use cases use some form of statistical processing or machine learning based predictions. This may be as underlying technology for a sensing solution (i.e. fruit assessment by camera) or to guide some form of autonomous actuation (i.e. smart weeder).

When these algorithms are included in a system of systems, some information modelling requirements arise. Systems may interchange information that is created by a predictive algorithm. As with sensor created data, predictions have their own kinds of metadata. This metadata is used/needed when deploying and using predictive algorithms in IoT supported use cases. Requirements may include for instance:

Acting on confidence levels - A smart device that receives predictions from an algorithm may decide whether to act on them or not based on up-front knowledge of the quality of the algorithm or the individual predictions;

Evaluation in the field - A smart device may store the predictions of a property and the actual measurements of the same property in parallel to facilitate validation and evaluation of the performance of models and algorithms.

Training and (local) retraining - A smart device may use a static, pre-trained model in daily operation, deployed as part of the solution. Alternatively, a solution may continue training on the job, in order to adapt to local circumstances or changes in the environment.

In our exploratory literature research, no information modelling standards on predictive algorithms similar to the domain and IoT standards described were found. A brief analysis on information modelling of algorithms and their outcomes was made based on the public API's of machine learning offerings frequently used by data scientists.

5.4.1 Modeling Algorithm Metadata

When developing prediction models, metadata, and thus information models, seem to be vital. This paragraph addresses typical algorithm metadata which might implicate common abstractions. One kind of metadata originates in the algorithm is used to make the prediction. On the other hand, metadata could be supportive when having key-figures or indicators as a result of a predictive algorithms.

There is a large number of predictive algorithms available. Some have their origin in statistics, while others are developed within the fields of artificial intelligence and more specifically machine learning. Although these fields have quite different approaches, many machine learning algorithms depend on statistics as the foundation too. And although ML algorithms are often seen as black boxes from a statistical perspective, their statistical underpinning is often used to study their qualities and boundaries.

The tasks of machine learning are often categorized as follows:

- Classification: assigning a class or label to a data point. For instance: What is the growing stage of the plant in this photo?
- Regression: assigning a value to a data point. For instance: What is the expected soil humidity of this field given its soil constitution and the weather the last few days?

– Clustering. For instance: assign to a cluster whether a plant has which kind of specific disease.

When solving classification problems, most accuracy scores are based on the number of false-positives (FP) and false-negatives (FN). These are also referred to as type I and type II errors. A number of accuracy concepts are used, that are derived from these:

- Precision and Recall
  - TP/(TP+FP)
  - TP/(TP+FN)
- Sensitivity and Specificity
  - TP/(TP+FN)
  - TN/(TN+FP)

Accuracy of classification is often visualized in a confusion matrix, that plots predicted against actual classes in a matrix. It instantly shows how many classifications were correct (on the diagonal) and which specific classes were confused for which other classes. Figure 10 below shows an example of such a confusion matrix for an algorithm trying to solve the (classical) problem of classifying three varieties of irises.

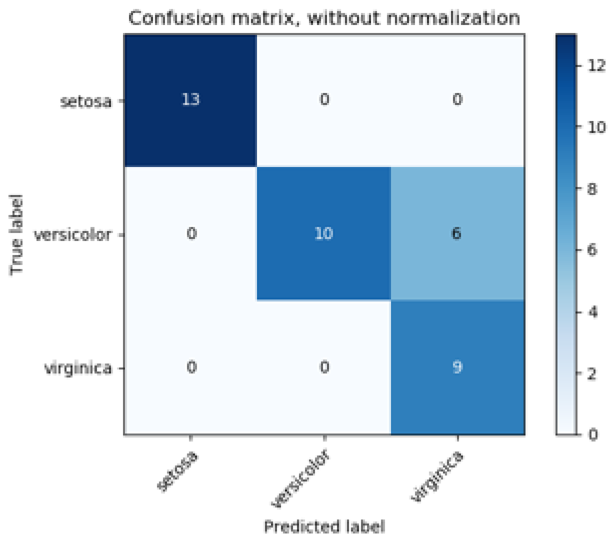


Figure 10: Example of a confusion matrix, source: SciPy documentation

When solving regression problems, the accuracy is often measured by error metrics, such as:

- Mean Absolute Error
- Mean Squared Error

In general, performance evaluations and metrics distinguish between performance on:

- Training Set: The historical data points used to train the algorithm or model;
- Evaluation Set: The part of the historical data used to evaluate the algorithms performance;
- Production: The performance in production setting.

5.4.2 Modelling Predictions

The performance measures in the previous paragraph were all at the level of algorithm or model. All predictions of the same algorithm share the same metadata. Some predictive techniques however also associate individual metadata to each prediction.

Probabilistic statement as outcome - Some algorithms are probabilistic in nature and their predictions are often too. That means they essentially predict the chance or likelihood of some fact being true. For example: The chance of this being an apple is 67%.

Intervals as outcome - Some algorithms predict values in terms of intervals, as a way to accommodate inherent uncertainty. Instead of predicting a single value, with a high likelihood of being false, they return a large interval in cases of uncertainty. Example: This apple is between 80 and 100 grams;

Confidence associated with outcome - Some algorithms have some internal notion of confidence that they share as metadata with their productions. In general, higher is better. But often these numbers have no known upper bound, making it hard to formulating a minimum confidence level or comparing confidence across different algorithms. Example: This apple is ripe with a confidence of 0.87.

The different metadata discussed, for both algorithms and predictions, are summarized by the figure below.

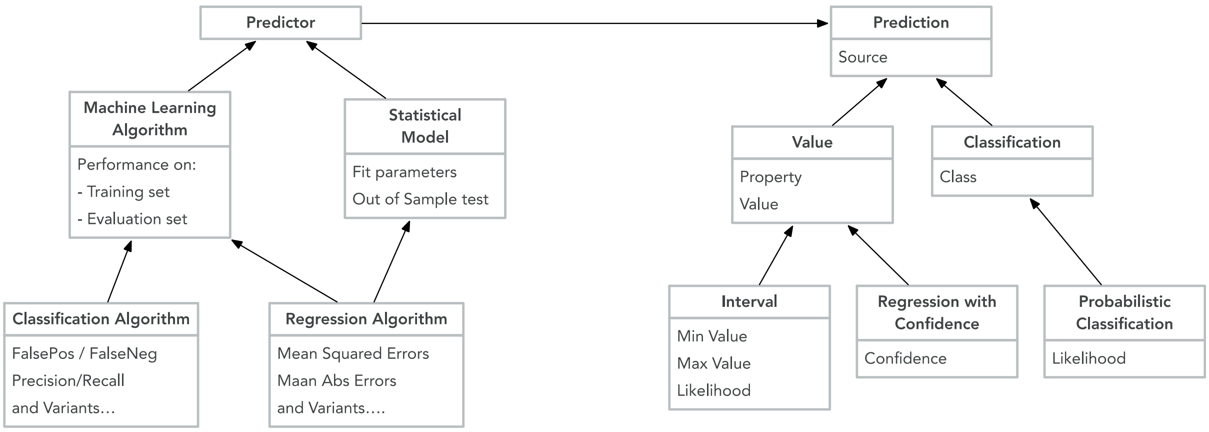


Figure 11: Predictive performance metadata at the level of algorithms and individual predictions

5.5 Cross-domain perspective

In paragraph 5.2.2, a detailed analysis is given for the arable domain. In this paragraph, a cross-domain perspective will be presented. Firstly, two relevant non-agri models are briefly mentioned. However, these are not elaborated since it's not or rarely used within the IoF2020 project. On the other hand, from a cross-domain perspective, the EPCIS is briefly analysed. More information is found in 5.5.1.

Two relevant non-agri models are:

- oneM2M (Machine 2 Machine). OneM2M is a global industry initiative which provides standards.[5,6] The base architecture is derived as a partnership from the Telco and Appliances Industry. In a Technical Specification (TS) document of ETSI, the architecture is used to map the SAREF model, considering oneM2M as a base ontology.[7]
- ISA-95 (International Society of Automation). This often referred as an international standard for developing interfaces between enterprises and control systems. The standard is developed for global manufacturers, and is intended for use in different kinds of processes (batch, continuous, repetitive).

5 http://onem2m.org/technical/published-drafts
6 https://en.wikipedia.org/wiki/ANSI/ISA-95
7 https://www.etsi.org/deliver/etsi_ts/103200_103299/103264/01.01.01_60/ts_103264v010101p.pdf

5.6 EPCIS

In this section, results of the analysis of the EPCIS data model are presented. EPCIS is used within IoF2020 *UC 5.3 Meat Transparency and Traceability* and *UC 3.4 Intelligent Fruit Logistics*. In section 5.6.3, UC 5.3 will be elaborated with the implementation of EPCIS.

EPCIS is a GS1 standard that enables trading partners to share information about the physical movement and status of products as they travel throughout the supply chain – from business to business and ultimately to consumers. It helps answering the "what, where, when and why" questions to meet consumer and regulatory demands for accurate and detailed product information. Examples are Tracking and Tracing, Inventory Management and IoT applications.[8]

EPCIS is intended to be used in conjunction with the GS1 Core Business Vocabulary (CBV) standard. The CBV provides definitions of data values that may be used to populate the data structures defined in the EPCIS standard. The use of the standardized vocabulary provided by the CBV standard is critical to interoperability and to query data by reducing the variation in how different businesses express common intent.

The EPCIS standard defines:

- A **data model** for visibility event data along with an accompanying concrete syntax for visibility data using the eXtensible Markup Language (XML);
- Open, standardised **interfaces** that allow for seamless integration of well-defined services in inter-company environments as well as within companies. There are two interfaces defined in the EPCIS standard:
  - A **capture interface** through which visibility event data conforming to the EPCIS data model may be delivered from capturing applications to a receiver, typically a persistent repository of EPCIS data; and
  - A **query interface** through which EPCIS event data may be requested by and delivered to a business application or a trading partner.[9]

Generically, EPCIS deals in two kinds of data: event and master. Event data arises in the course of carrying out business processes and is captured through the EPCIS Capture Interface and made available for query through the EPCIS Query Interfaces. Master data is additional data that provides the necessary context for interpreting the event data. It is available for query through the EPCIS Query Control Interface, but the means by which master data enters the system is not specified in the EPCIS standard.

The GS1 Core Business Vocabulary (CBV) standard provides standardised vocabulary elements for many of the vocabulary types used in EPCIS event types. In particular, the CBV defines vocabulary elements for the following EPCIS Standard Vocabulary types: Business Step, Disposition, Business Transaction Type, and Source/Destination Type. The CBV also defines templates for constructing vocabulary elements for the following EPCIS User Vocabulary types: Object (EPC), Object Class (EPC Class), Location (Read Point and Business Location), Business Transaction ID, Source/Destination ID, and Transformation ID.[10][11]

5.6.1 EPCIS event data model (Core Event Type Module)

The Core Event Types data definition module specifies the Event Types that represent EPCIS data capture events. These events are typically generated by an EPCIS Capturing Application and provided to EPCIS infrastructure using the data capture operations. These events are also returned in response to query operations that retrieve events according to query criteria.

This module defines six event types, one very generic event and five subclasses that can represent events arising from supply chain activity across a wide variety of industries:

---

8 www.gs1.org/epcis
9 EPC Information Services (EPCIS) Standard Release 1.2
10 Core Business Vocabulary Standard Release 1.2.2
11 EPCIS and CBV Implementation Guideline Release 1.2

- *EPCISEvent* is a generic base class for all event types in this module as well as others.
- *ObjectEvent* represents an event that happened to one or more physical or digital objects.
- *AggregationEvent* represents an event that happened to one or more objects that are physically aggregated together (physically constrained to be in the same place at the same time, as when cases are aggregated to a pallet).
- *TransactionEvent* represents an event in which one or more objects become associated or disassociated with one or more identified business transactions.
- *TransformationEvent* represents an event in which input objects are fully or partially consumed and output objects are produced, such that any of the input objects may have contributed to all of the output objects.

In the figure below is a visualisation of the data model presented.

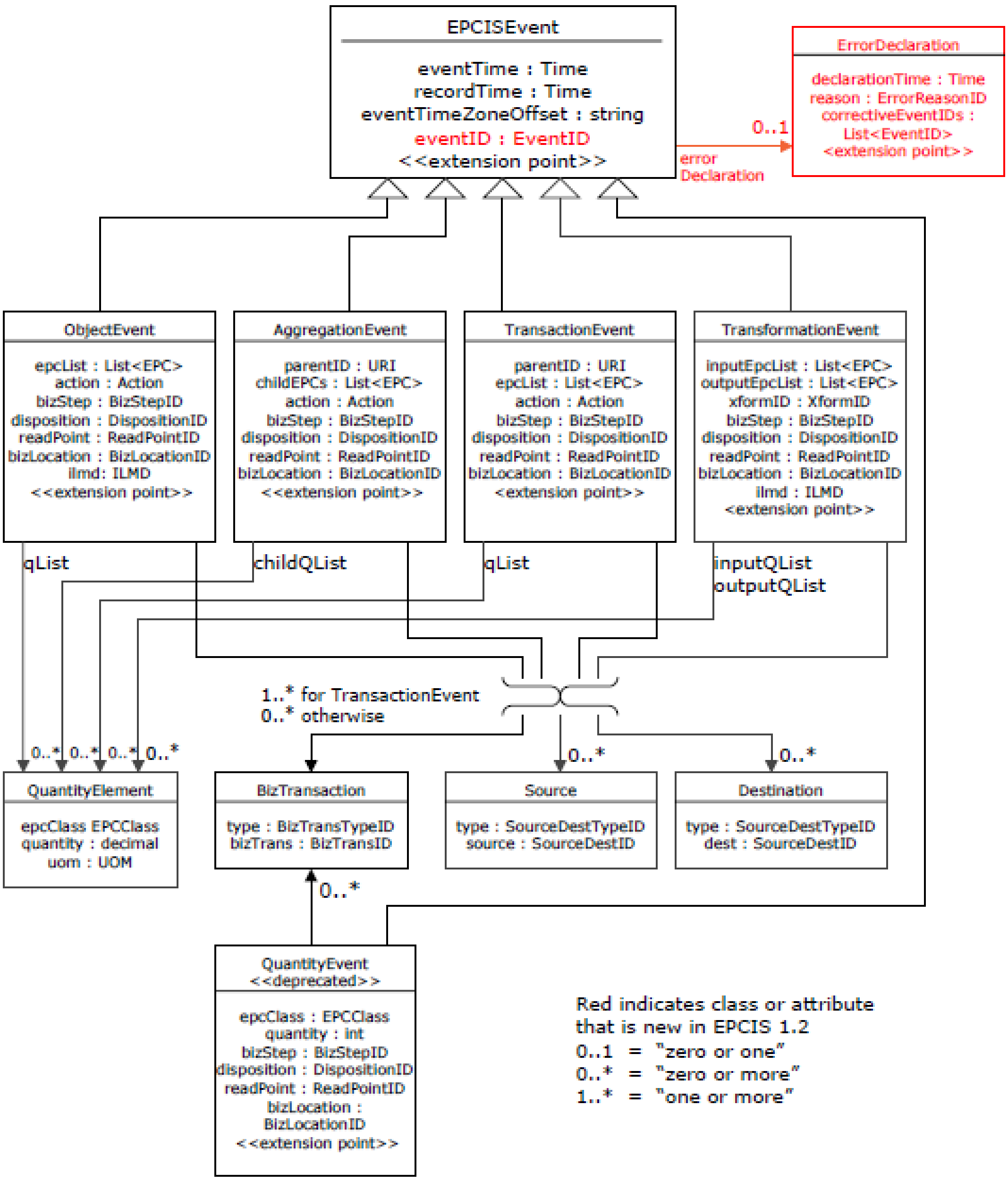


Figure 12: EPCIS Data Model.

5.6.2 Implementation in the Meat domain

Use Case 5.3 deals with Meat Transparency and Traceability by developing a cloud services for real-time transparency data exchange of all stakeholders in the meat supply chain. Herein, it focuses on supporting proactive auditing and enhancing transparency and traceability of meat along the value chain, based on monitored chain event data in an EPCIS-infrastructure. See below for an overview of the supply chain within UC 5.3.

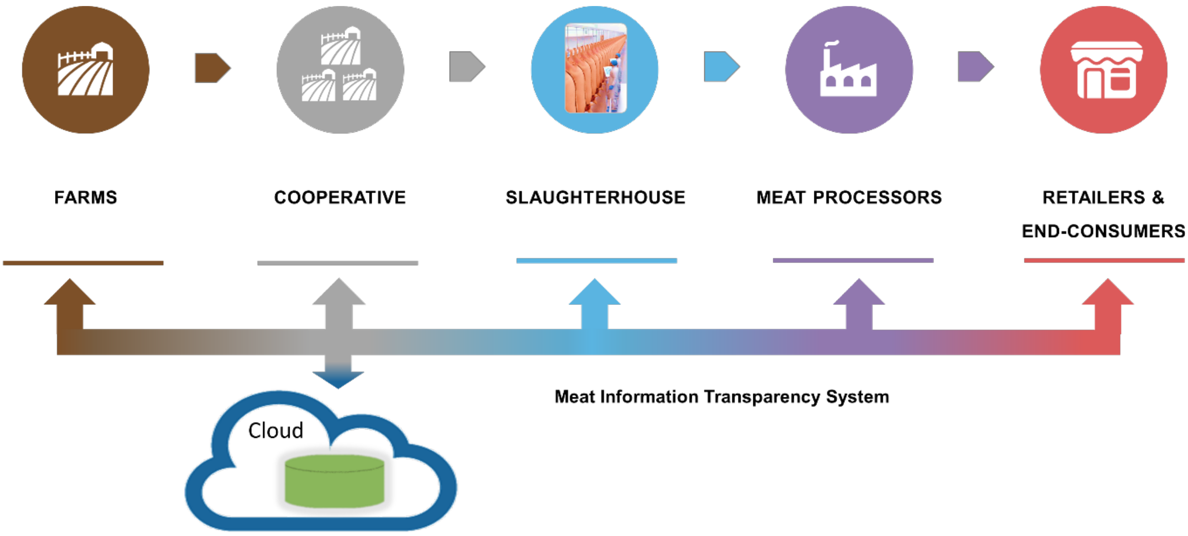


Figure 13: Overview Supply Chain UC 5.3

The overall idea is that information is delivered for proactive auditing based on events occurring during a pig's lifetime, like birth, growth, or veterinary inspections. The pen-up event for example is the basis for checking the minimum weaning age of a piglet and the vet event helps to check whether the minimum veterinary inspection cycle is met. The EPCIS-based solution in UC 5.3 allows to collect these data from different sources in a standardized way and display critical KPIs at the press of a button.

The EPCIS-based data model used in this UC provides three main functionalities:

(1) **Event Data Capture**

(2) **Event Data Repository**

(3) **Data Query & Accessing Applications.**

A **Data Capture** interface not only consumes data from multiple data sources (e.g. FMIS, FIWARE, etc.) but also translates data of diverse formats to standardized event information. The standardized **EPCIS Data Repository** stores all relevant life cycle information in a standardized format and enable at the same time multiple destination systems to consume the information. A standardized **Data Query** interface allows partners and systems to poll information on demand while also subscription-based information about specific events pushed to multiple destination systems is possible. **Accessing Applications** finally provide aggregated information to systems or partners.

5.6.3 Implementation in the Fruit domain

The general goal of the IoF2020 Use Case 3.4 on intelligent fruit logistics is to connect IoT-enabled Returnable Transport Items (RTIs) with smart applications to open a new dimension of added value services in multi-actor fruit and vegetable supply networks.

RTIs for packaging and transporting fresh produce play a crucial role in the agri-food value chain. Millions of RTIs are handled every day within and between thousands of companies. The main challenge for these companies is to manage their assets in an efficient way without losses and a possibility to track and trace them along the full supply chain. The generation of produce-and-process-related information and knowledge in the value chain are the prerequisites to create innovative services that will help to make dynamic multi-actor supply networks more efficient, avoid waste, inform consumers and even work towards autonomous procedures for product recalls.

At this moment of time, trading partners participating in Reusable Transport Items (RTI) pool systems have only limited visibility of related business processes (e.g. which parties get RTIs from whom). This results in a couple of significant issues:

- **Theft of RTIs** along with obtaining refunds for RTIs by fraud
- **Black use** (i.e. internal use or inner cycles from the farmer to DCs/marketplaces and back) e.g. leading to loss in revenue due to the increase of dwell times
- **Deception in shipping notices** (declaring a greater number of RTIs as actually dispatched)

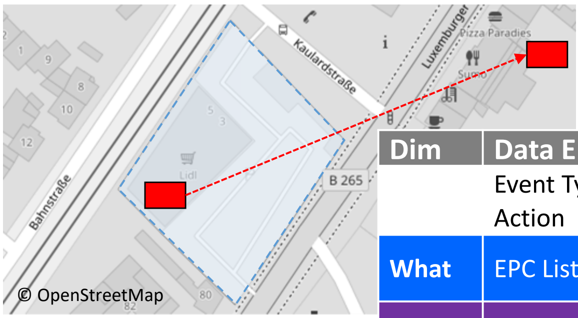
The strategic objective of the use case is to connect IoT-enabled RTIs with smart applications to open a new dimension of added value services, supply and demand simulation, forecasting, supply chain coordination and decentralised delivery models for produce in a circular manner.

The basis of the use case is an IoT-enabled RTI, which is able to determine its geographic position and send it autonomously within a defined frequency. At the same time, it is embedded into the tray without impairing the tray's applicability.

The following EPCIS messages enable all above-mentioned use cases. Thereby, based on either time- or condition-triggered business rules (e.g. 'trigger an EPCIS event once an asset arrives at a specific location/every 30 minutes when the RTI is in transit'/etc.), all data required to populate the fields of EPCIS events pertaining to assets in circulation can be either inferred from the data transmitted by the end nodes or provided by the application server:

- When: time stamp (when observation took place)
- What: GRAI (through mapping with the node's ID, if the GRAI is not transmitted directly)
- Where: geographic position or GLN
- Why:
  - Business step (e.g. 'arriving', 'departing', 'transporting', 'other')
  - Disposition (e.g. 'in transit', 'in progress')

For instance, an EPCIS event triggered once a geo-fence is left would look like this:



Dim	Data Element	Content
	Event Type	Object Event
	Action	OBSERVE
What	EPC List	Tray ID: GRAI (Global Individual Asset Identifier)
When	Event Time	Date/Time of event, e.g. 10th October 2018, 11:30 a.m.
	Event Time Zone Offset	Time Zone Offset to UTC, e.g. CET (+01:00)
Where	Read Point	Physical location ID of a business partner: GLN (Global Location Number)
Why	Business Step	Departing (CBV)
	Disposition	In transit (CBV)

Figure 14: Example of an EPCIS event.

Within UC 3.4, in order to achieve the above mentioned goals, the locations in the supply chain must be identified and classified. As they are partly not known, a Location Management Application was set up for managing unknown locations, identifying and analysing all incoming unknown locations.

Based on an EPCIS interface, the IoT data processing platform provides information from the IoT device of the RTIs to the Location Management application. Data on unknown locations is sent from the IoT platform to the Location Management Application, then contextualised, classified and finally sent back with enriched information.

E.g., for identifying the unknown locations, the following data is provided:

Table 7: Example of EPCIS data.

Seq	Field name	Description	Mandatory	Type
1.	eventTime	Date and time of event (location update) (ISO8601)	Yes	CHAR
2.	epcList:epc	GRAI in EPC URI format	Yes	CHAR(255)
3.	action	Always “OBSERVE”	Yes	CHAR(255)
4.	bizStep	Always “urn:epcglobal:cbv:bizstep:other”	Yes	CHAR(255)
5.	readpoint	Latitude, longitude and radius Format: “<id>geo:45.000001,3.900001;u=40</id>” with up to 20 decimals. Radius is rounded in meters.	Yes	CHAR(255)
6.	iof:dwellTime uomType="HUR"	The duration a GRAI is reported on this location (in hours) Format: HH (without decimals)	Yes	CHAR

5.7 FIWARE Data Models

The figure below describes the original OMA NGSI meta-model, the fundamental metamodel around the FIWARE Data Models. There are three main elements in this meta-model, as shown in the figure below: Entities, Attributes and Metadata.

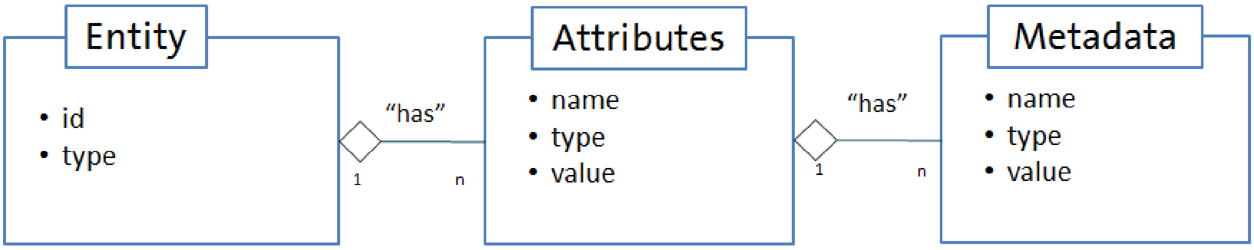


Figure 15: The OMA NGSI meta-model

Entities are the center of gravity in the NGSI information model. An entity represents a thing, i.e., any physical or logical object (e.g., a sensor, a person, a room, an issue in a ticketing system, etc.). Each entity has an entity id.

Furthermore, the type system of NGSI enables entities to have an entity type. Entity types are semantic types intended to describe the type of thing represented by the entity. For example, a context entity with id "sensor-365" could have the type "TemperatureSensor". Each entity is uniquely identified by the combination of its id and type. It is noteworthy that these elements are always mandatory when the meta-model is instantiated.

Attributes are properties of context entities. For example, the current speed of a tractor could be modelled as attribute "speed" of entity "tractor-104". In the NGSI information model, attributes have a name, type, value and metadata.

The attribute name describes the kind of property the attribute value represents of the entity, for example "speed". The attribute type represents the NGSI value type of the attribute value, which is usually similar or equivalent to the JSON data type. The attribute value, finally, contains the actual data and optional metadata describing properties of the attribute value like e.g. accuracy, provider, or a timestamp.

Metadata is used as an optional part of the attribute value, as described above. Similar to attributes, each piece of metadata has a name, describing the role of the metadata in the place where it occurs; for example, the metadata name "accuracy" indicates that the metadata value describes how accurate a given attribute value is; a type, describing the NGSI value type of the metadata value; a value containing the actual metadata.

5.7.1 JSON Representation

An NGSI Entity is represented by a JSON object with the following structure:

- The entity id is encoded by the object's id property, which value is a string containing the entity id.
- The entity type is encoded by the object's type property, which value is a string containing the entity's type name.
- Entity attributes are specified by additional properties, whose names are the name of the attribute and whose representation is described below. Obviously, "id" and "type" are not allowed to be used as attribute names.

An NGSI Attribute is represented by a JSON object with the following syntax:

- The attribute value is specified by the value property, which value might be any JSON value.
- The attribute NGSI type is specified by the type property, which value is a string containing the NGSI type, which usually is just the type of the JSON value and can be omitted.

The Attribute metadata is specified by the *metadata* property. Its value is another JSON object which contains a property per metadata element defined (the name of the property is the name of the metadata element). Each metadata element, in turn, is represented by a JSON object containing the following properties:

- *value*: Its value contains the metadata value, which may correspond to any JSON value.
- *type*: Its value contains a string representation of the metadata NGSI type, which usually is just the type of the JSON value and can be omitted.

The representation described above is usually referred as the NGSI normalized format. Its main drawback is that it is very verbose and more difficult to be parsed or validated. That is the main reason behind the introduction of simplified formats in NGSI, namely the key-value format (which can be activated by the *options=keyValues* query URL parameter in the REST API). The key-value simplified format allows representing entity attributes by their values only, leaving out the information about type and metadata. This is a very convenient format particularly for client applications, as it simplifies considerably the parsing and data extraction processes.

The **geospatial** properties of an entity can be represented by means of regular attributes. The provision of geospatial properties enables the resolution of geographical queries. The format used by the FIWARE Data Models is GeoJSON, which is a geospatial data interchange format based on JSON. GeoJSON provides greater flexibility allowing the representation of point altitudes or even more complex geospatial shapes, for instance multi geometries.

5.7.2 Using JSON Schema

JSON Schema is a JSON media type for defining the structure of JSON data. JSON Schema is intended to define validation, documentation, hyperlink navigation, and interaction control of JSON data.

The specification of FIWARE Data Models uses JSON Schema. Another advantage of adopting JSON-Schema is that other tools, such as Swagger (https://swagger.io/), will be enabled, allowing developers to create specific REST APIs on top of the existing NGSI APIs.

5.7.3 Example of Agri-food Data Models

Below it can be found an excerpt of an agri-food Data Model for Weather, specified using JSON-Schema. The vocabulary of attributes used is defined by the FIWARE Data Models for Weather, which ultimately are based on the work done by the GSMA under the IoT Big Data Project.

```
{
    "properties": {
      "type": {
        "type": "string",
        "enum": [
          "WeatherObserved"
        ],
        "description": "NGSI Entity type"
      },
      "dateObserved": {
        "type": "string",
        "format": "date-time"
      },
      "precipitation": {
        "type": "number",
        "minimum": 0
      },
      "barometricPressure": {
        "type": "number",
        "minimum": 0
      },
```

```
        "solarRadiation": {
          "type": "number",
          "minimum": 0
        },
        "pressureTendency": {
          "oneOf": [
            {
              "type": "string",
              "enum": [
                "raising",
                "falling",
                "steady"
              ]
            },
            {
              "type": "number"
            }
          ]
        },
        "dewPoint": {
          "type": "number"
        }
       }
    }
  ],
  "required": [
    "id",
    "type",
    "dateObserved",
    "location"
  ]
}
```

Figure 16: JSON Schema – WeatherObserved Data Model

The example above describes an excerpt of the schema for a weather observation entity. First of all, the different NGSI attributes are defined as properties of the JSON object. In addition, there are four elements required in this data model:

- *id*: corresponding to the entity id (automatically generated by data providers).
- *type*: corresponding to the entity type which shall be equal to “WeatherObserved”.
- *dateObserved*: which is the date and time at which the observation was made. Please note how this attribute was defined (*format=date-time*).
- *location*: which is a GeoJSON geometry which defines the location associated to the weather observation.

5.7.4 NGSI Instantiation Example

Below there is an example of instantiation of the common information meta-model using the key-values representation format.

```
{
  "id": "Spain-WeatherObserved-Valladolid-2016-11-30T07:00:00.00Z",
  "type": "WeatherObserved",
  "barometricPressure": 938.9,
  "dataProvider": "Fiware",
  "dateObserved": "2016-11-30T07:00:00.00Z",
```

```
  "location": {
    "type": "Point",
    "coordinates": [
      -4.754444444,
      41.640833333
    ]
  },
  "precipitation": 0,
  "pressureTendency": 0.5,
  "relativeHumidity": 1,
  "source": "http://www.aemet.es",
  "temperature": 3.3,
  "windDirection": -45,
  "windSpeed": 2
}
```


Figure 17: WeatherObserved Entity Instance (key-values format)

5.7.5 NGSI-LD

The OMA NGSI information model has been evolved to better support linked data (entity relationships), property graphs and semantics (exploiting the capabilities offered by JSON-LD). This work has been conducted under the ETSI ISG CIM initiative and has been branded as NGSI-LD. The main constructs of NGSI-LD are: *Entity, Property and Relationship*. NGSI-LD Entities (instances) can be the subject of Properties or Relationships. See figure below.

In terms of the traditional NGSI data model, Properties can be seen as the combination of an attribute and its value. Relationships allow establishing associations between instances using Linked Data. In practice, they are conveyed by means of an NGSI attribute, but with a special value (relationship's object) which happens to be a URI which points to another entity.

There is a special type of Property, named *GeoProperty*, and it is used to convey geospatial characteristics of Entities. In addition, NGSI-LD standardizes a set of predefined Properties, namely:

- **location**, main geospatial location of an Entity encoded as GeoJSON.
- **observedAt**, observation timestamp encoded using ISO8601.
- **unitCode**, unit of measurement encoded using the UN/CEFACT standard.

Properties and Relationships can be the subject of other Properties or Relationships. Thus, in the NGSI-LD information model there are no attribute's metadata, but just "properties of properties" or "properties of relationships". It is not expected to have infinite graphs, and in practice, only one or two levels of Property or Relationship "chaining" will happen. Typically, there will be one, equivalent to the NGSI metadata abstraction.

NGSI-LD Entities are represented using JSON-LD, a JSON-based serialization format for Linked Data. The main advantage of JSON-LD is the capability of expanding JSON terms to URIs, so that vocabularies can define terms unambiguously.

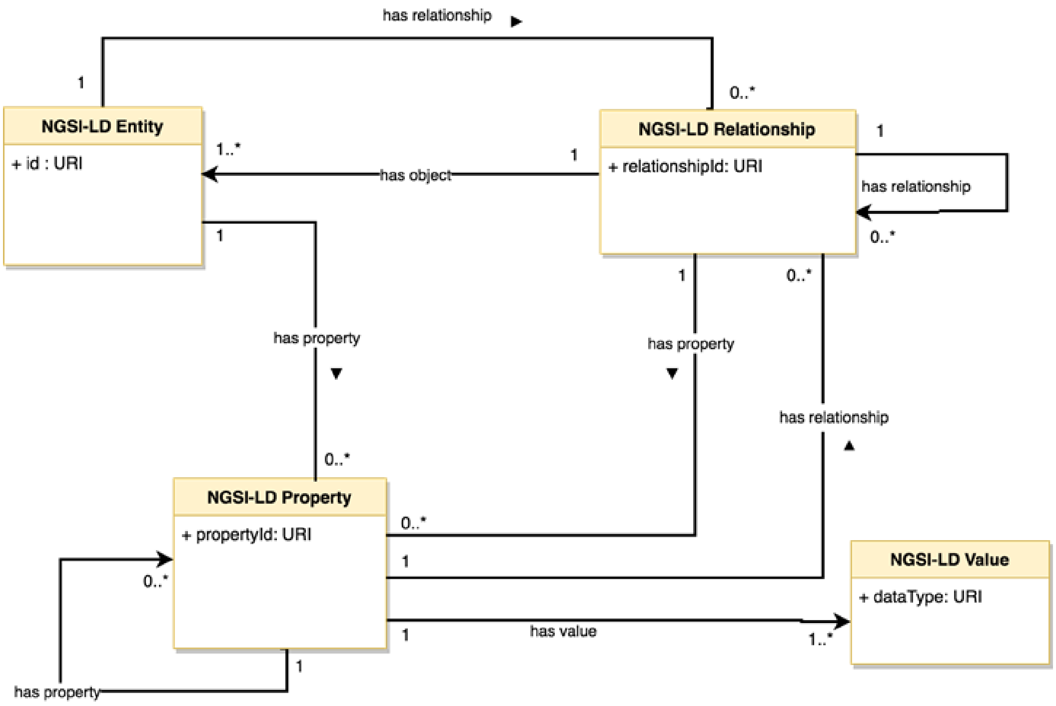


Figure 18: NGSI-LD information model expressed using the UML notation

5.7.6 Transitioning to NGSI-LD

In order to transition NGSI data model instantiations to NGSI-LD, the following recommendations can be followed:

- Each Data Model instance shall have a JSON-LD @context, providing an unambiguous definition by mapping terms to URIs. For practical reasons, it is recommended to have a unique @context resource, containing all terms, subject to be used in every IoF2020 Data Model, the same way as schema.org does.
- Entity ids have to be converted to **URIs**, preferably using the NGSI-LD URN namespace.
- Regular entity attributes have to be converted to JSON-LD nodes of type "*Property*".
- *ref* attributes (pointing to other entities) have to be converted to JSON-LD nodes of type "*Relationship*".
- The *timestamp* metadata item has to be mapped to the "*observedAt*" member of a Property node.
- The *unitCode* metadata item has to be mapped to the "*unitCode*" member of a Property node.
- The NGSI *DateTime* type has to be properly encoded as per the JSON-LD rules.
- The NGSI *geo:json* type has to be renamed to "*GeoProperty*".

The FIWARE Community has already provided a simple script to migrate FIWARE NGSI entity representations to NGSI-LD, see:
https://github.com/Fiware/dataModels/blob/master/tools/normalized2LD.py

5.7.7 NGSI-LD Instantiation Example

Below there is an instantiation example of an Entity encoded using NGSI-LD. It is noteworthy how the @context is used, combining the ETSI NGSI-LD Core @context together with the @context defined by the FIWARE Data Models.

```
{
    "id": "urn:ngsi-ld:AgriParcelRecord:8f5445e6-f49b-496e-833b-e65fc97fcab7",
```

```
    "type": "AgriParcelRecord",
    "hasAgriParcel": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:AgriParcel:d3676010-d815-468c-9e01-25739c5a25ed"
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Polygon",
            "coordinates": [ [ 100, 0 ], [ 101, 0 ], [ 101, 1 ],
                             [ 100, 1 ], [ 100, 0 ] ]
        }
    },
    "soilTemperature": {
        "type": "Property",
        "value": 27,
        "unitCode": "CEL",
        "observedAt": "2017-05-04T12:30:00Z"
    },
    "airTemperature": {
        "type": "Property",
        "value": 20,
        "unitCode": "CEL",
        "observedAt": "2017-05-04T12:30:00Z"
    },
    "solarRadiation": {
        "type": "Property",
        "value": 15,
        "unitCode": "N78",
        "observedAt": "2017-05-04T12:30:00Z"
    },
    "hasDevices": {
        "type": "Relationship",
        "object": [
            "urn:ngsi-ld:Device:4a40aeba-4474-11e8-86bf-03d82e958ce6",
            "urn:ngsi-ld:Device:63217d24-4474-11e8-9da2-c3dd3c36891b"
        ]
    },
    "observedAt": {
        "type": "Property",
        "value": {
            "@type": "DateTime",
            "@value": "2017-05-04T12:30:00Z"
         }
    },
     "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```

Figure 19: NGSI-LD instantiation example

5.7.8 Summary of FIWARE Data Models for Digital Farming

Table 8 below summarizes the available FIWARE Data Models for the smart agriculture domain. They are largely based on the former work being done by the GSMA IoT Big Data Project. IoF2020 is actively contributing to the extension and enhancement of these data models to cover gaps, so that the needs of different trials and use cases are properly covered. Such new contributions and extensions will be described in chapter 6.

Table 8: FIWARE Data Models for the smart agriculture domain.

Data Model	Specification
Building	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Building.md
Vehicle	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Vehicle.md https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Vehicle-Type.md
Weather	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Weather-Observed.md https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Weather-Forecast.md
AgriParcel	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Parcel.md
AgriParcelRecord	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Parcel-Record.md
AgriParcelOperation	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Parcel-Operation.md
AgriGreenhouse	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Greenhouse.md
AgriCrop	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Crop.md
AgriPest	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Pest.md
AgriSoil	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Soil.md
AgriProductType	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Agri-Product-Type.md
Device	https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Device-Model.md https://github.com/GSMADeveloper/NGSI-LD-Entities/blob/master/definitions/Device.md

5.8 Summary / Five Steps to Adopting Existing Models

As our preliminary literature search shows, there are a large number of models and standards available useful in information modelling. We have shown that relevant models can be discovered along three dimensions:

- Commonalities in a shared domain, for instance agri-food or a more specific subdomain (i.e. arable farming);
- Commonalities from the application of internet of things technology;
- Commonalities from applying predictive technologies, such as statistical modelling and machine learning.

In this section, we propose 5 steps to develop an agri-food use case based on internet of things technology, adopting existing models or model fragments from those three domains:

- **Establish the objectives with reuse of models and standards**. It is important to establish the objectives considering reuse of models, ranging from data exchange being part of the use case, to reaching high productivity in building agri-food IoT solutions.
- **Establish the role of information models in the architecture**. For many people, information modelling is a synonym of data or even database modelling. It is important to establish which components in the architecture require information modelling. Typical areas in the architecture that require (their own) information modelling are:
    - Central components such as the FMIS and IoT systems or infrastructures;
    - 3rd Party systems;
    - A broker or integration facility used for system integration.

  Information models are used to design and interact with different kind of interfaces:
    - Databases and their related extraction and import facilities;
    - Integration middleware such as message busses and the messaging formats needed to organize that integration;
    - (Open) API's.
- **Adopt existing models for the relevant domain entities.** Explore the existing domain models and adopt (parts of) models for the domain specific entities relevant to the use case. Where possible, take the summary domain model from 5.1 into account and make sure that the identified domain entities include the inputs, processing and outputs to be optimized within the use case. Also, consider any KPI's that the use case tries to achieve. These often map well to reducing the consumption of inputs, reducing effort and waste in the processing and optimizing the outputs in the former control loop model.
- **Choose metadata for observations and actions.** For each information entity identified, choose metadata according to the role they play in the IoT control loop described in section 5.2. The generic models mentioned in that section offer standardized metadata elements both for observations/measurements by sensors and for instructions/actions send to actuators.
- **Choose metadata for predictions and the algorithms/models that make them.** For those information elements that are predictions from statistical models or machine learning algorithms, choose metadata capturing prediction performance and confidence based on the models presented in 5.3. And consider whether to accommodate predictions and actuals in parallel, to support large-scale evaluation and additional training, or even situational training for specific environments.

6 Semantic Interoperability for Agri-Food

6.1 Introduction

This chapter reports (at the time of writing) on the main activities conducted under IoF2020 towards enhancing open platforms, with a view to enabling open interoperability for agri-food. The activities described have been organized in different work streams that tackle different MIMs:

- **MIM1** (*IoT Service Layer*). A proposal for interworking with *W3C Web of Things* is described, so that device manufacturers in the agri-food domain can semantically describe their devices, and to plug and play them in customer solutions or Farm Management Information Systems.
- **MIM2.1** (*Harmonized Data Models*). The set of GSMA Harmonized Data Models[12] for agri-food have been extended to better support the Arable and Meat Trials. Particularly, mappings with some relevant ADAPT entities (or fragments) have been defined.
- **MIM2** (*Mediation Layer*). Open source *Node-Red* Adaptors have been provided so that existing proprietary solutions in agri-food can better plugin into an open platform based on the System of Systems approach.

It is noteworthy that IoF2020 plans to continue working with other trials, particularly the Vegetables and Dairy Trial, so that further enhancements for MIMs are also achieved, particularly concerning *MIM2.1*, with the development of extra harmonized Data Models.

6.2 Enhancement of the IoT Service Layer

One of the main challenges currently faced by Smart Solutions (including agri-food solutions) is IoT Device interoperability, so that devices supplied by different manufacturers can be integrated.

W3C has been working on the definition of new Web standards that can enable simple IoT interoperability based on Web architectures, semantic description formats and scripting APIs. Such standard is known as the ***W3C Web of Things (WOT)***. The primary benefit of using WoT for describing devices and device types is the capability of deploying devices to different Open Platforms for Smart Solutions without the need of doing customization at device software / firmware level.

6.2.1 W3C Web of Things Overview

The *W3C Web of Things* (WoT) has been created to enable interoperability across IoT Platforms and application domains. WoT provides mechanisms to formally describe IoT interfaces, allowing IoT devices and services to communicate with each other, independent of the underlying implementation and across multiple networking protocols.

A **Web Thing** (or Thing) is the abstraction of a physical or virtual entity (e.g., a device or a room) and is described by standardized metadata. A Web Thing has four *architectural aspects* of interest:

- The **behavior** aspect of a Thing includes both the autonomous behavior and the handlers for the Interaction Affordances.
- The **Interaction Affordances** provide a model of how Consumers can interact with the Thing through abstract operations, but without reference to a specific network protocol or data encoding.
- The **protocol binding** adds the additional detail needed to map each Interaction Affordance to concrete messages of a certain protocol. In general, different concrete protocols may be used to support different subsets of Interaction Affordances, even within a single Thing.
- The **security configuration** aspect of a Thing represents the mechanisms used to control access to the Interaction Affordances and the management of related public and private metadata.

[12] https://github.com/GSMADeveloper/NGSI-LD-Entities

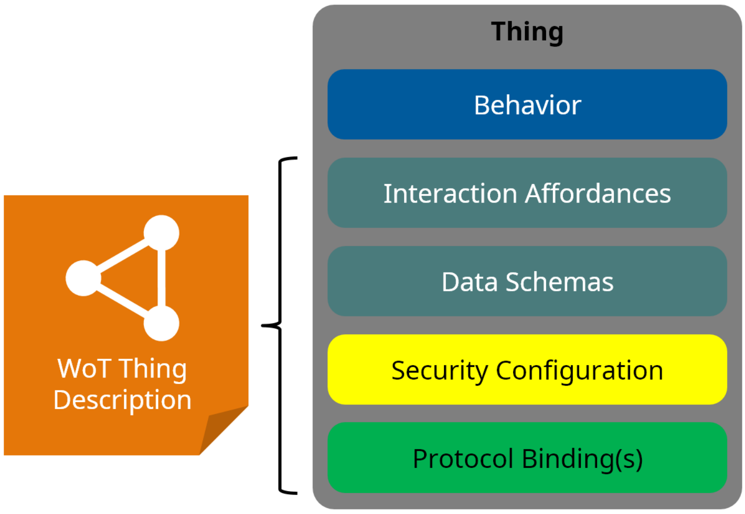


Figure 20: Architectural aspects of a **Web Thing** (source W3C)

In addition, a WoT Thing is composed by the following *building blocks*:

- **Thing Description (TD)** describes the network-facing interface of a Thing. Each Thing must have a corresponding TD. The TD Specification is normative and can be found at https://www.w3.org/TR/wot-thing-description/. The Thing Description Specification is expected to become a W3C Recommendation in Q2 2019.
- **Binding Templates** can be used to describe multiple protocol bindings, so that a Thing can communicate with different IoT Platforms. Currently this part of the standard is non-normative, but it will evolve during next standardization phases.
- **Scripting API** enables implementation of the application logic of a Thing using a standardized contract for JavaScript. Currently this part of the standard is non-normative, but it will evolve during next standardization phases.

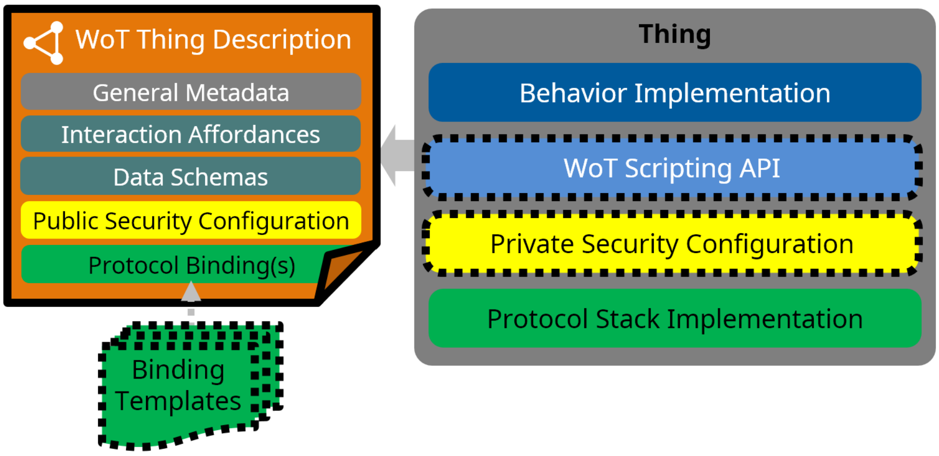


Figure 21: Relationship of WoT Building Blocks to the Architectural Aspects of a Thing. (source W3C)

The Interaction Model of W3C WoT introduces an intermediate abstraction that formalizes the mapping from application intent to concrete protocol operations and also narrows the possibilities how Interaction Affordances can be modelled. In addition to navigation affordances (i.e., Web links), Things may offer three other types of *Interaction Affordances*:

- **Properties**: expose internal state of the Thing that can be directly accessed (read) and optionally manipulated (write) or watched (observe), i.e. pull and push communication styles. Examples of

Properties are sensor values (read-only), stateful actuators (read-write), configuration parameters (read-write), Thing status (read-only or read-write), or computation results (read-only).

- **Actions:** allow to invoke a function or service exposed by the Thing or to trigger a process. Examples of Actions are changing multiple Properties simultaneously, changing Properties over time such as fading the brightness of a light (dimming) or invoking a long-lasting process such as printing a document.
- **Events** are Thing-initiated notifications, discrete events, or streams of values sent asynchronously to the receiver. Push communication style is used to deliver events. Examples of Events are discrete events such as an alarm or samples of a time series pushed regularly.

While this narrow waist allows to decouple Consumers and Things, these four types of Interaction Affordances are still able to model virtually all interaction possibilities found in IoT devices and services.

6.2.2 Thing Description (TD)

In W3C WoT, the description metadata must be a *WoT Thing Description* (TD). TD defines an information model based on a semantic vocabulary and a serialized representation based on JSON (but also allows JSON-LD processing). The main characteristics of a TD are:

- A TD is the default external, textual (Web) representation of a Thing.
- A TD is human-readable and machine-understandable.
- A TD is instance-specific (i.e., describes an individual Thing, not types of Things).
- A TD can be hosted locally by a Thing or externally (ex. by a Thing Catalogue).

A TD contains:

- Thing instances **general metadata** such as name, ID, descriptions.
- **Web Links** to related Things or other Web documents, making the Web of Things **navigable**, for both humans and machines.
- **Interaction Affordance** metadata (*Properties, Actions, Events*)
- Public **Security Configuration** metadata.
- **Schemata** of the information exchanged with the Thing (using *JSON Schema*).

For each of the aspects mentioned above there is a TD Vocabulary, which is essentially a set of Terms that can be used to build data structures, interpreted as objects in the traditional object-oriented sense. Objects are instances of classes and have properties. The TD Core Vocabulary is depicted in the figure below.

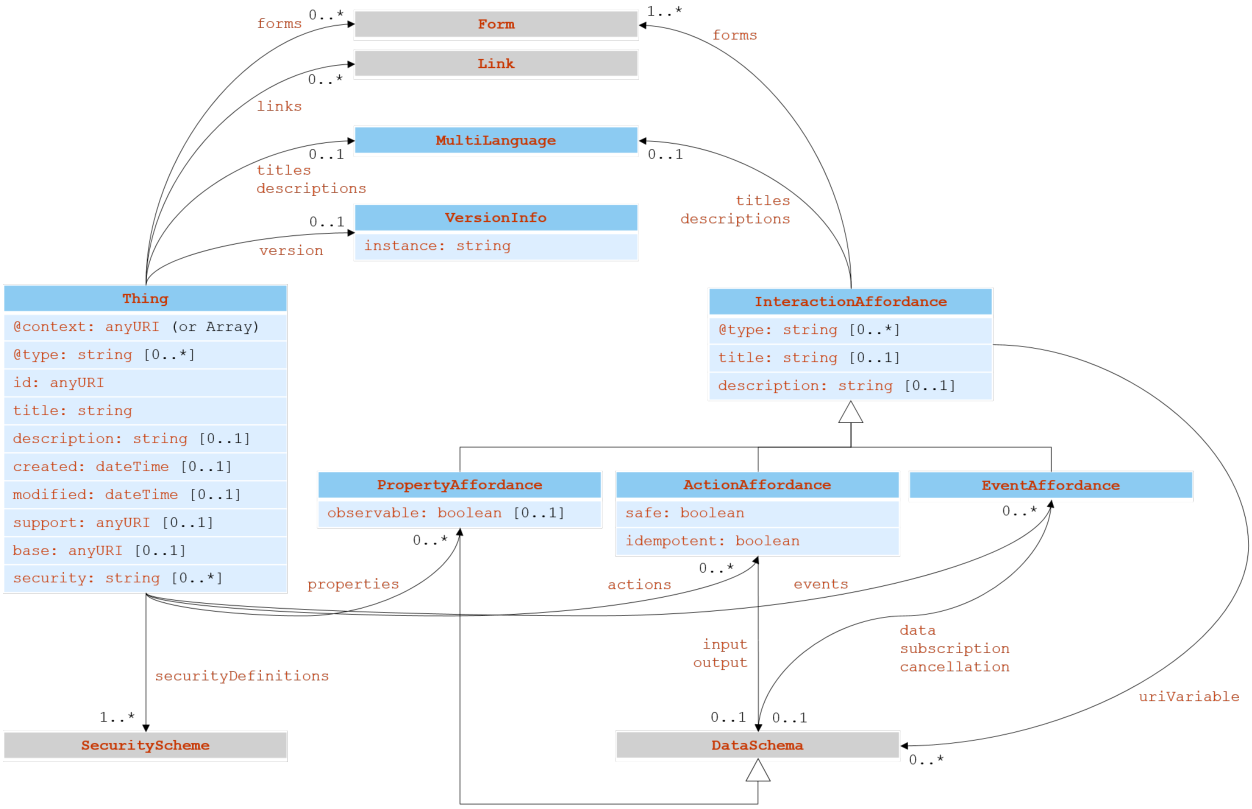


Figure 22: TD Core Vocabulary. (source W3C)

An example of a Thing Description can be found below. In this case it corresponds to the description of a Weather Station providing data through two different Interaction Affordances:

- A Property, using HTTP protocol bindings offers the current temperature value.
- An Event, using MQTT protocol bindings notifies high temperature conditions.

```
{
  "id": "urn:dev:ops:32473-WoT-WeatherStation-example1",
  "@type": "Thing",
  "title": "MyWeatherStation",
  "properties": {
    "temperature": {
      "type": "number",
      "forms": [
        {
          "href": "https://192.168.1.187:8080/temperature",
          "contentType": "text/plain"
        }
      ]
    }
  },
  "events": {
    "highTemperature": {
      "data": {
        "type": "number"
      },
      "forms": [
        {
          "href": "mqtt://192.168.1.187:1883/highTemperature",
          "contentType": "text/plain"
```

```
        }
      ]
    }
  },
  "@context": "https://www.w3.org/2019/wot/td/v1"
}
```

Figure 23: TD Example. Weather Station Sensor

One important aspect about TDs is that they can be semantically annotated, thanks to the extensibility offered by JSON-LD. The figure below shows how our initial TD can be enriched using semantic annotations with concepts provided by the SAREF http://ontology.tno.nl/saref/ and SAREF4Agri https://mari-apoveda.github.io/saref-ext/OnToology/SAREF4AGRI/ontology/saref4agri.ttl/documentation/index-en.html ontologies.

```
{
  "id": "urn:dev:ops:32473-WoT-WeatherStation-example1",
  "@type": "s4agri:WeatherStation",
  "title": "MyWeatherStation",
  "properties": {
    "temperature": {
      "@type": "s4agri:AirTemperature",
      "type": "number",
      "forms": [
        {
          "href": "https://192.168.1.187:8080/temperature",
          "contentType": "text/plain"
        }
      ]
    }
  },
  "events": {
    "highTemperature": {
      "@type": "saref:NotifyCommand",
      "data": {
        "type": "number"
      },
      "forms": [
        {
          "href": "mqtt://192.168.1.187:1883/highTemperature",
          "contentType": "text/plain"
        }
      ]
    }
  },
 "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    {
      "s4agri": "https://w3id.org/def/saref4agri#",
      "saref": "https://w3id.org/saref#"
    }
 ]
}
```

Figure 24: TD with semantic annotations using SAREF and SAREF4Agri

6.2.3 WoT Servients

Things must be hosted on networked system components with a software stack to realize interaction through a network-facing interface, the WoT Interface of a Thing. One example of this is an HTTP server running on an embedded device with sensors and actuators interfacing the physical entity behind the Thing abstraction. However, W3C WoT does not mandate where Things are hosted; it can be on the IoT device directly, an Edge device such as a gateway, or the cloud.

A **Servient** is a software stack that implements the WoT building blocks. *Servients* can host and expose Things and/or consume Things (i.e., host Consumers). Depending on the Protocol Binding, *Servients* can perform in both *server and client* role. See figure below.

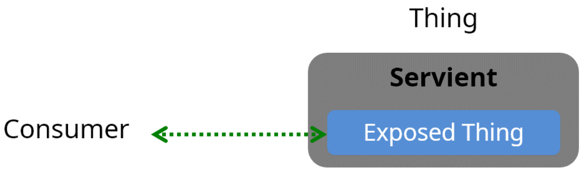


Figure 25: Servient as a Thing. (source W3C)

Likewise, an *Intermediary* is another WoT architecture component that can be implemented by a Servient. An Intermediary is located between a Thing and its Consumers, performing the roles of both a Consumer (to the Thing) and a Thing (to the Consumers). Consumer and Intermediary can communicate in a different protocol than Intermediary and Thing. For example, an Intermediary can provide a bridge between a Thing that uses the Constrained Application Protocol (CoAP) and the application of a Consumer that uses HTTP.

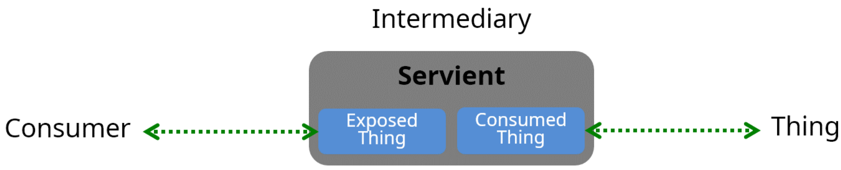


Figure 26: Servient as an intermediary. (source W3C)

6.2.4 Integration of WoT into Open Platforms for Smart Solutions

Once a semantic description of devices (TD) is available at the IoT Service Layer (offered by *MIM1*), it is necessary to solve the issue of integration of the data provided by these devices into the information management layer (offered by *MIM3*), so it can be mashed-up with other data coming from additional devices or external systems.

The final aim is to offer higher level APIs and information models to consumers (applications, higher-level services) through the NGSI-LD API, as key enabler of the *MIM3*. This implies that the interoperability between Web of Things and NGSI-LD shall be defined, which is equivalent to the definition of the “glue” between *MIM1* and *MIM2*. The figure below shows the overall architecture to enable interworking between NGSI-LD and Web of Things.

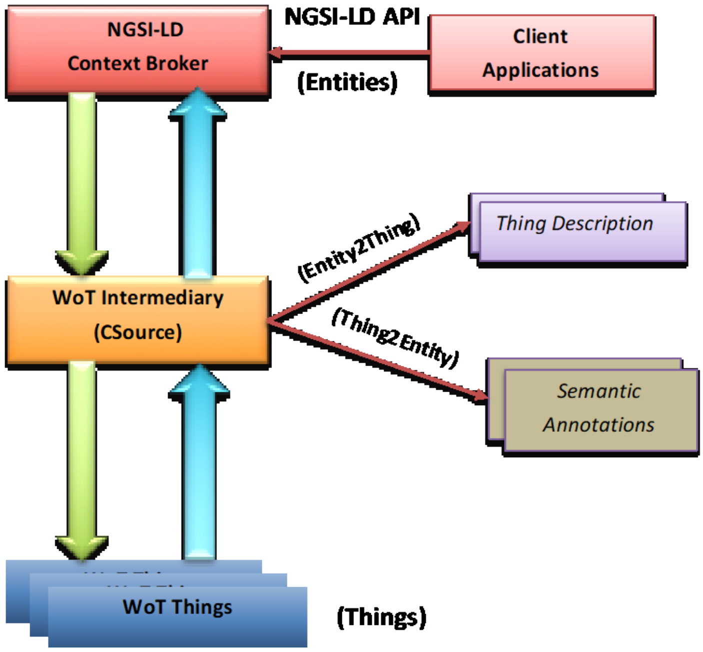


Figure 27: Interworking architecture between NGSI-LD and Web of Things

The proposed interworking architecture revolves around an "NGSI-LD-WoT Intermediary" which plays a dual role:

- NGSI-LD Context Source, i.e. it is capable to offer NGSI-LD represented data through the NGSI-LD REST APIs
- WoT Intermediary Servient

Therefore, the responsibilities of the NGSI-LD-WoT Intermediary are:

- Act as an *Intermediary Servient* to WoT Things.
- Perform NGSI-LD *mappings* for WoT Events, Properties and Actions **using extra semantic annotations added to the TD**.
- Expose WoT Properties as NGSI-LD Attributes (*Context Source*).
- Propagate WoT Event data to an NGSI-LD Context Broker, as the WoT Intermediary does not have data storage capabilities itself.
- Export WoT Actions as NGSI-LD Actuations

The mappings between WoT Interaction Affordances and NGSI-LD Elements are expressed in the table below:

Wot Interaction Affordance	NGSI-LD Element
Property	Attribute (Property or Relationship)
Event	Attribute (Property or Relationship)
Action	*For further study*. Preliminary draft study can be found in the proceedings of the FIWARE Summit Genoa 2019.

Semantic annotations can guide the mapping process between WoT Interaction Affordances and NGSI-LD Elements. Those semantic annotations can be conveyed by using concepts already defined by standard ontologies, particularly ETSI SAREF (and its extensions) and W3C Semantic Sensor Networks (SSN). Below, there is an example that shows how the TD corresponding to a weather station (already presented) can be annotated, using W3C SSN, to enable the mapping to an NGSI-LD Entity of type *AgriParcelRecord*. Annotations based on SAREF concepts could also be used.

```
{
  "id": "urn:dev:ops:32473-WoT-WeatherStation-example1",
  "@type": "s4agri:WeatherStation",
  "title": "MyWeatherStation",
  "properties": {
    "temperature": {
      "@type": ["s4agri:AirTemperature", "sosa:Observation"],
      "sosa:hasFeatureOfInterest": "urn:ngsi-ld:AgriParcelRecord:P456",
      "sosa:observedProperty": "https://uri.fiware.org/ns/data-models#airTemperature"
      "type": "number",
      "forms": [
        {
          "href": "https://192.168.1.187:8080/temperature",
          "contentType": "text/plain"
        }
      ]
    },
 },
 "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    {
      "s4agri": https://w3id.org/def/saref4agri#,
      "sosa": "http://www.w3.org/ns/sosa/"
    }
 ]
}
```

Figure 28: TD Semantically annotated to enable interworking with NGSI-LD

The resulting NGSI-LD Entity, after performing the corresponding mappings, guided by annotations, is shown below. An NGSI-LD Property (*airTemperature*) for an Entity of type *AgriParcelRecord* has been generated, as prescribed by the semantic annotation on the TD. It is noteworthy that the *source* of such an NGSI-LD Property has been set to the URI that identifies the corresponding Thing. Thanks to this mechanism, a navigation path and relationship between Things (*MIM1*) and Entities (*MIM3*) is being established.

```
{
  "id": "urn:ngsi-ld:AgriParcelRecord:P456",
  "type": "AgriParcelRecord",
  "airTemperature": {
     "type": "Property",
     "value": 22.5,
     "observedAt": "2019-05-22T12:34:55Z",
     "source": "urn:dev:ops:32473-WoT-WeatherStation-example1"
  },
  "hasAgriParcel": {
     "type": "Relationship",
     "object": "urn:ngsi-ld:AgriParcel:d3676010"
  },
  "location": {
     "type": "GeoProperty",
     "value": {
       "type": "Polygon",
       "coordinates": [ [ 100, 0 ], [ 101, 0 ], [ 101, 1 ],
          [ 100, 1 ], [ 100, 0 ] ]
     }
```

```
  },
  "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
  ]
}
```

Figure 29: NGSI-LD Entity resulting from the mapping process

6.2.5 Next Steps

This chapter has presented the preliminary ideas around the adoption of Web of Things as an enhancement of open platforms, with a view to further device semantic interoperability in the agri-food domain. However, additional work has to be done during the next phases of the IoF2020 project, in order to consolidate these ideas. First of all, an initial proof of concept of the WoT Intermediary has be to developed, so that the presented approach can be validated and refined. In addition, it is needed to extend the semantic annotation mechanism to properly support mapping of WoT Actions. Last but not least, it is of great importance the validation with ontology experts of the proposed semantic annotations, and to study different annotation alternatives based on SAREF, W3C SSN or custom ontologies.

6.3 Enhancement of Harmonized Data Models

This activity is intended to the creation and refinement of Harmonized Data Models for the Smart agri-food Domain, so that they can offer an enhanced *MIM2.1* to interested trials and use cases. The work has initially been focused on the **Arable** and **Meat** Trials, as a collaboration with UC1.4 (*Farm Machine Interoperability*), UC1.6 (*Internet of Potato*) and *ShareBeef* has been put in place. It is noteworthy that our intention is to replicate this approach with other trials and use cases, provided there is an interest and a value proposition for the corresponding use cases. Currently, there is a great potential with Vegetables Trial, concerning greenhouse modelling.

6.3.1 Enhancing developer experience with JSON Schema

The starting point of our work has been the agri-food Data Models defined under IoT Big Data Project of GSMA, adopted by FIWARE and already described on chapter 5 of this deliverable. However, initially, those Data Models had only been specified textually and no JSON Schema was available. Therefore, one of the first activities conducted has been the creation of JSON Schemas for all the agri-food Data Models defined by GSMA / FIWARE. Those schemas can now be found at https://github.com/FIWARE/dataModels/tree/master/specs/AgriFood and have been integrated in the common framework of the FIWARE Data Models project. As a result, now the Data Models can be used more easily by application developers together with API client or validation tools.

6.3.2 ADAPT Fragments

One of the main areas of enhancement developed, for the time being, has been around the Arable Domain.

One of the main data interoperability initiatives in the Arable domain is ADAPT[13] (Agricultural Data Application Programming Toolkit). The main target of ADAPT is to make it easy for the various systems that a grower wants to use in their business “talk to each other” when it comes to a shared data format.

In the Arable Domain, and particularly in UC 1.4, there is an interest in offering a mapping between certain ADAPT concepts (or fragments of them) and NGSI-LD Entities. As a result, the communication of task execution from different machine vendors’ cloud and FMIS systems can be simplified and harmonized on a portable and multi-vendor configuration.

The figure below provides an overview of how the NGSI API, together with the agri-food Data Models, can enable a seamless interoperability between a cloud system (proprietary) of a machine vendor and an open Farm Machine Information System. At first stage, a tractor or other agricultural vehicle provides task data

[13] https://adaptframework.org/

to the cloud environment, possibly on a real time basis. After tasks are completed, it is necessary to transfer results to the FMIS system, so that proper audits and analytics can take place. Besides, the previous workflow could also happen the other way around, i.e. in the FMIS some tasks are planned, and through NGSI they can be transferred to the machine vendor cloud (and finally to the machine). This approach enables an FMIS to work with different machines and vendors, thus increasing portability of applications. In addition, this also results in a real benefit to the farmer, who is no longer lock-in a particular solution, as she can work seamlessly with multiple machine vendors, and, potentially, integrated with different FMIS.

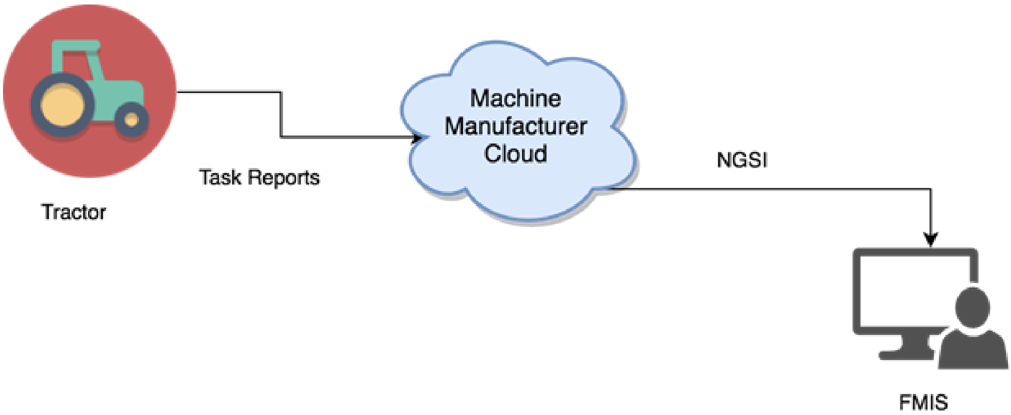


Figure 30: Communication between Machine Manufacturer's cloud and FMIS through NGSI(-LD)

The approach followed has consisted on the reuse and extension of the existing FIWARE/GSMA agri-food Data Models, if possible, and the creation of new Data Models when necessary. The result of our effort, after different coordination working sessions between UC1.4 and WP3, is summarized in the table below. Those readers interested in all the details should check the FIWARE Data Models Github repository, agri-food section at https://github.com/FIWARE/data-models/tree/master/specs/AgriFood

FIWARE Agri Data Model	ADAPT Entity	Enhancements
AgriParcel	Field CropZone	Added a new Attribute named *relatedSource* so that source application(s) can be tracked. Added a new Attribute named *seeAlso* so that extra references associated to this Entity can be found (inside or outside the system). Added a new Attribute named *belongsTo* to track the Relationship with **AgriFarm**. Added a new Attribute named *ownedBy* to track the ownership Relationship.
AgriParcelOperation	WorkOrder WorkRecord IrrRecord OperationData	Added a new Attribute named *relatedSource* so that source application(s) can be tracked. Added a new Attribute named *seeAlso* so that extra references associated to this Entity can be found (inside or outside the system). Added a new Attribute named *workOrder* to track the Relationship with ADAPT WorkOrder. Added a new Attribute named *workRecord* to track the Relationship with ADAPT WorkRecord. Added a new Attribute named *irrigationRecord* to track the Relationship with ADAPT IrrRecord.

FIWARE Agri Data Model	ADAPT Entity	Enhancements
AgriCrop	Crop	Added a new Attribute named *relatedSource* so that source application(s) can be tracked. Added a new Attribute named *seeAlso* so that extra references associated to this Entity can be found (inside or outside the system). Added a new Attribute named *agroVocConcept* to point to the **AgroVoc** concept corresponding to this Entity.
AgriPest		Added a new Attribute named *relatedSource* so that source application(s) can be tracked. Added a new Attribute named *seeAlso* so that extra references associated to this Entity can be found (inside or outside the system). Added a new Attribute named *agroVocConcept* to point to the **AgroVoc** concept corresponding to this Entity.
AgriProductType	Product ProductTypeEnum	Added a new Attribute named *relatedSource* so that source application(s) can be tracked. Added a new Attribute named *seeAlso* so that extra references associated to this Entity can be found (inside or outside the system). Added a new Attribute named *agroVocConcept* to point to the **AgroVoc** concept corresponding to this Entity.
DeviceModel	DeviceModel DeviceElement Implement	Add new Device Model categories corresponding to implements and other machinery used in Agriculture
AgriFarm	Farm	New Entity Type to capture details about a Farm. See JSON Schema at https://github.com/FIWARE/dataModels/blob/master/specs/AgriFood/AgriFarm/schema.json
AgriApp		New Entity Type to capture details about a mashed-up, integrated Agricultural App which acts mainly as a data provider. See JSON Schema at https://github.com/FIWARE/dataModels/blob/master/specs/AgriFood/AgriApp/schema.json

With all the proposed enhancements, UC1.4 now has all the tools to proceed with a proper integration between Farm Machine Manufacturer's Cloud systems and Farm Management Information Systems.

6.3.3 Animal Data

The Animal Data initiative developed by WP3 is intended to facilitate the participation of different stakeholders of the meat value chain within an IoF2020 enabled technology ecosystem. This initiative has benefited a lot from the collaboration of the Sharebeef use case that has contributed with a Harmonized Data Model for representing the static and dynamic conditions of animals or cattle within an agri-food environment. It is noteworthy that this Data Model can also be reused by other trials and actually an interest has already been raised by the Dairy trials.

The next diagram describes the beef chain considered under the ShareBeef use case. In this diagram, different stakeholders of the meat chain are described together with some of their interactions.

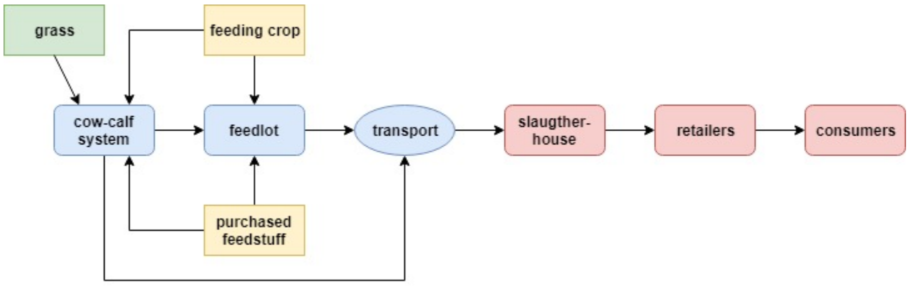


Figure 31: Stakeholders of the meat chain

During the execution of the *Sharebeef* use case, it is necessary to define several entities to handle the information generated in the proposed solution. Within all these entities, the animal entity, which is the gravitational centre of the solution stands out in the first place. The preliminary analysis of the Animal Data Model is depicted below and has resulted in a formal specification of a Data Model that can be found at https://github.com/FIWARE/dataModels/blob/master/specs/AgriFood/Animal/doc/spec.md

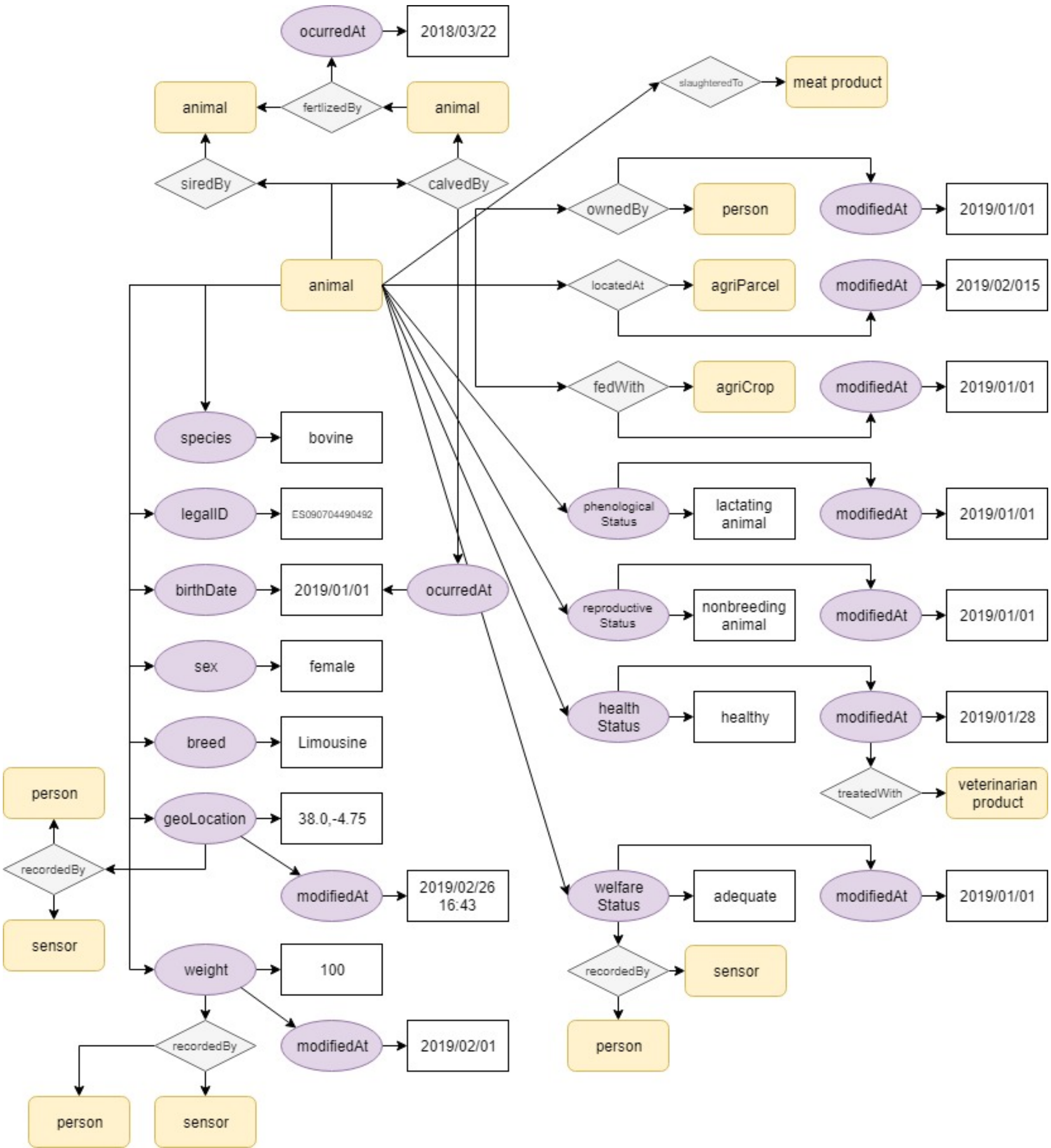


Figure 32: Animal Data Model preliminary analysis (source *Sharebeef* use case)

The proposed animal data model has been made from a more general point of view, trying to adjust it to the information coming from the devices and sensors used in the Sharebeef UC but with a view to make it generic enough for enabling reuse by other use cases and trials. Below a JSON example instantiation of the animal Data Model is shown (in the NGSI-LD simplified, key-value format):


```
{
    "id": "urn:ngsi-ld:Animal:ca3f1295-500c-4aa3-b745-d143097d5c01",
    "type": "Animal",
    "species": "sheep",
    "legalId": "ES142589652140",
    "birthdate": "2017-01-01T01:20:00Z",
    "sex": "female",
    "breed": "Merina",
    "calvedBy": "urn:ngsi-ld:Animal:aa9f1295-425c-8ba3-b745-b653097d5a87",
    "siredBy": "urn:ngsi-ld:Animal:aa9f1295-425c-8ba3-b745-b653097d5a87",
    "location":
    {
        "type": "Point",
        "coordinates":
        [
            -4.754444444,
            41.640833333
        ]
    },
    "weight": 65.3,
    "ownedBy": "http://person.org/leon",
    "locatedAt": "urn:ngsi-ld:AgriParcel:1ea0f120-4474-11e8-9919-672036642081",
    "phenologicalCondition": "maleAdult",
    "reproductiveCondition": "inCalf",
    "healthCondition": "healthy",
    "fedWith": "urn:ngsi-ld:FEED:1ea0f120-4474-11e8-9919-0000000081",
    "welfareCondition": "adequate"
    "@context": [
      "https://schema.lab.fiware.org/ld/context",
      "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 33: Animal Data Model instantiation example

6.4 Enhancement of the Mediation Layer

One of the main functionalities implemented by the Mediation Layer is the transformation of raw or non-harmonized data coming from devices or other external services, into curated, harmonized and possibly aggregated data that can be exposed to data processing algorithms or analytics.

In order to facilitate data integration in agri-food, open source Node-Red Adaptors have been developed so that existing proprietary solutions in agri-food can better plugin into an IoF2020-compliant open platform based on the System of Systems approach (described in detail by **D3.8**). The following Node-Red Nodes are already developed or under development:

- *Context Broker*. A Configuration Node that allows to set up the endpoint and security parameters of a FIWARE Context Broker (exporting NGSIv2 or NGSI-LD).
- *NGSI Entity*. Provides the JSON representation (both NGSIv2 and NGSI-LD) of an Entity queried by its identifier.
- *NGSI Dataset*. Allows to define an NGSI(LD,v2) dataset by providing the corresponding filtering and projection conditions.

- *NGSI Update*. Allows to update (in *upsert* or *update* mode) one or more NGSI(LD,v2) Entities.
- *NGSI Subscription*. Allows to subscribe to NGSI(LD,v2) Entities and Attributes enabling the propagation of NGSI(LD,v2) notification data to a Node-Red flow.
- *NGSIv2ToNGSI-LD*. A Node that allows to transform NGSIv2 representations to NGSI-LD.

A preliminary version of a Node-Red package, containing some of the Nodes described above, has already been published and can be found at https://flows.nodered.org/node/node-red-contrib-fiware_official. More details about this are provided in **D3.8**.

The figure below describes, succinctly, how the integration of an existing non-harmonized agri-food solution (based or not on FIWARE) can be performed by means of a Node-Red flow that adapts between the non-harmonized representation of Entities to the harmonized one (through a transformation node). More details about how this integration can be developed with a concrete example will be provided in **D3.8**

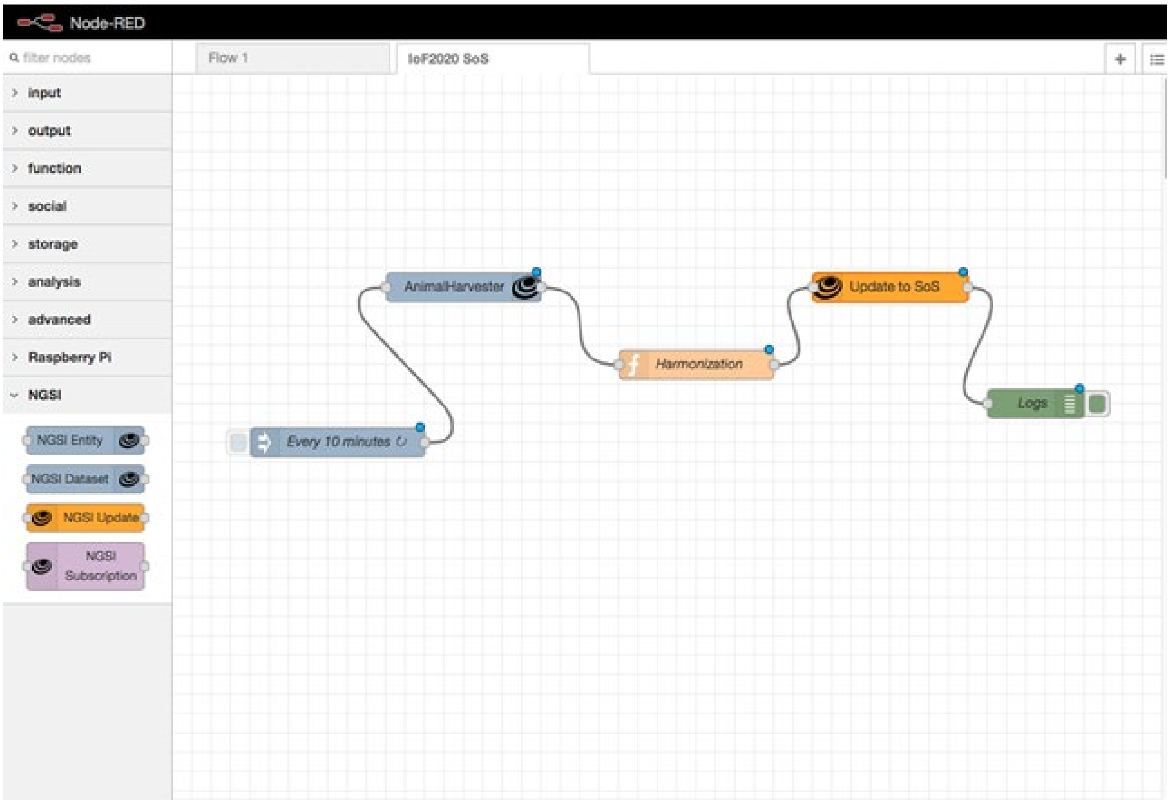


Figure 34: Node-Red flows to support harmonization of IoF2020 solutions.

6.5 Future enhancements and outlook

This chapter has reported on the enhancements developed at the time of writing. They demonstrate that existing open platforms, with some additional work, can perfectly fulfil the needs of further interoperability and portability of Smart agri-food solutions with a view to adopting a Systems of Systems approach. However, more work has to be done during the last phase of the IoF2020 project in order to consolidate this ongoing effort. The envisioned work can be summarized as follows:

Develop the first version of the Web of Things components, aligning the integration with existing Web of Things implementations, particularly Mozilla WebThings https://iot.mozilla.org/ and Eclipse IoT Node-WoT http://www.thingweb.io/.

- Extend the set of Harmonized Data Models by direct dialogue with all the Trials, especially the Meat, Dairy and Vegetables trials.
- Fully develop and consolidate the data integration offer and System of Systems approach through Node-Red, as a simplified tool that facilitates application development and integration.

7 FIWARE Open platform

7.1 Introduction

This chapter describes the FIWARE Open Platform and specifies how it can be used. It starts with an overview of the components, and then focuses on the Context Broker as the main component. The FIWARE Open Platform can be used to implement Smart Agri-food solutions compliant with the MIMs described on Chapter 3 and ready to be integrated in a System of Systems.

7.2 Overview of FIWARE platform and components

FIWARE is an open source initiative defining a universal set of standards for context data management which facilitate the development of Smart Solutions for different domains such as Smart Cities, Smart Industry, **Smart agri-food**, and Smart Energy.

In any smart solution there is a need to gather and manage context information, processing that information and informing external actors, enabling them to actuate and therefore alter or enrich the current context. The *FIWARE Context Broker* is the core component of any "Powered by FIWARE" platform. It enables the system to perform updates and access the current state of context.

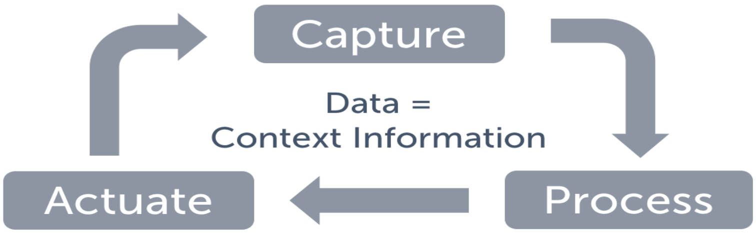


Figure 35: Smart solutions data flow

The *Context Broker* in turn is surrounded by a suite of additional platform components, which may be supplying context data (from diverse sources such as a CRM system, social networks, mobile apps or IoT sensors for example), supporting processing, analysis and visualization of data or bringing support to data access control, publication or monetization.

All interactions between applications or platform components and the *Context Broker* take place using the FIWARE NGSI RESTful API – a simple, yet powerful open standard, which currently offers two different flavours the NGSIv2 https://fiware.github.io/specifications/ngsiv2/stable/ and NGSI-LD https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf, the latter is the one recommended by ETSI as it fully supports Linked Data.

The open standard nature of the FIWARE APIs offers programmers the ability to port their applications across different "Powered by FIWARE" platforms and a **stable framework for future development**. Additional functionality can be easily added to a Smart Solution simply by using additional FIWARE or third-party components for which the integration with the FIWARE Context Broker component is solved. This integration is simplified since all components comply to the NGSI standard interface, which **eliminates vendor lock in**. The component-based nature of a FIWARE based solution allows for **re-architecting as the solution evolves** according to business needs.

Building around the FIWARE Context Broker, a rich suite of complementary FIWARE components are available (see figure below), dealing with:

- **Interfacing with the Internet of Things (IoT), Robots and third-party systems**, for capturing updates on context information and translating required actuations.
- **Context Data/API management, publication and monetization**, bringing support to usage control and the opportunity to publish and monetize part of managed context data.
- **Processing, analysis and visualization of context information**, implementing the expected smart behavior of applications and/or assisting end users in making smart decisions.

In addition, the FIWARE platform brings a number of deployment tools easing the deployment and configuration of FIWARE or third-party components and their integration with *FIWARE Context Broker* technology.

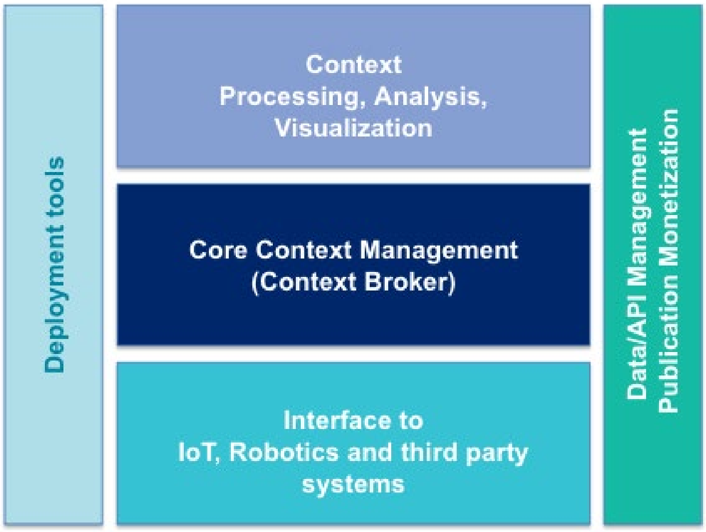


Figure 36: FIWARE Components grouped by Chapter.

However, FIWARE is not about take it all or nothing. It is not mandatory to use these complementary FIWARE components but other third platform components to design the desired hybrid platform. As long as it uses the FIWARE Context Broker technology to manage context information, the platform or solution can be labelled as "Powered by FIWARE" and solutions build on top as well.

FIWARE components go through an extensive set of Quality Assurance (QA) tests run by dedicated members of the FIWARE Community. This way, you may gain trust about the status and maturity of any FIWARE component. QA tests are not only functional but cover performance testing as well as quality of available documentation, training objects, etc.

7.2.1 Core Context Management

The **Orion Context Broker** Generic Enabler is the core and mandatory component of any "Powered by FIWARE" platform or solution. It enables to manage context information in a highly decentralized and large-scale manner. It provides the NGSI APIs which are simple yet powerful Restful APIs enabling to perform updates, queries or subscribe to changes on context information.

The Orion Context Broker Generic Enabler holds information about the current context. However, context information evolves over time, creating a context history. Accompanying the Orion Context Broker component as part of the Core Context Management is:

- The **Quantum Leap** Generic Enabler supports the storage of NGSI data into a time series database (*CrateDB*)
- The **Draco** Generic Enabler is an alternative data persistence mechanism for managing the history of context. It is based on *Apache NiFi* and is a dataflow system based on the concepts of flow-based programming. It supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic and also offers an intuitive graphical interface.

Other enablers in this area are **STH Comet** (short term historical data based on *mongoDB*) and **Cygnus** (data persistence based on *Apache Flume*).

7.2.2 Interface with IoT, Robots and Third-Party Systems

A number of Generic Enablers are available making it easier to interface with the Internet of Things, Robots and Third-party systems for the purpose of gathering valuable context information or trigger actuations in response to context updates:

- The **IDAS** Generic Enabler offers a wide range of IoT Agents making it easier to interface with devices using the most widely used IoT protocols (LWM2M over CoaP, JSON or UltraLight over HTTP/MQTT or OPC-UA)
- The **Fast RTPS** Incubated Generic Enabler has been adopted as default middleware in ROS2, the widely known Robot Operating System, therefore it helps to interface with robotics systems.
- The **Micro XRCE-DDS** Incubated Generic Enabler is an IoT-oriented middleware based on a publish-subscribe messaging pattern.
- The **OpenMTC** Incubated Generic Enabler brings an open source implementation of the oneM2M standard. A northbound interface with the Orion Context Broker is implemented as part of the product.
- The **Domibus** Incubated Generic Enabler helps users to exchange electronic data and documents with one another in a reliable and trusted way.

7.2.3 Context Processing, Analysis and Visualization

A number of Generic Enablers are available making it easier to process, analyse or visualize context information for the purpose of implementing the "smart behaviour" expected in any application:

- The **Wirecloud** Generic Enabler brings a powerful web mashup platform making it easier to develop operational dashboards which are highly customizable by end users.
- The **Knowage** Generic Enabler brings a powerful Business Intelligence platform enabling to perform business analytics over traditional sources and big data systems.
- The **Kurento** Generic Enabler enables real-time processing of media streams supporting the transformation of video cameras into sensors as well as the incorporation of advanced application functions (integrated audio and video communications, augmented reality, flexible media playing and recording, etc.)
- The **Cosmos** Generic Enabler enables to make easier data analysis over context integrated with some of the most popular Data Processing platforms.
- The **FogFlow** Generic Enabler is a distributed execution framework to support dynamic processing flows over cloud and edges.
- The **Perseo** Incubated Generic Enabler introduces Complex Event Processing (CEP) defined using a rules-based system, enabling you to fire events which send HTTP requests, emails, tweets, SMS messages etc.

7.2.4 Context Data / API Management, Publication and Monetization

You can implement secured access to components in the architecture of any "Powered by FIWARE" solution using Generic Enablers of the Security chapter:

- The **Keyrock** Identity Management Generic Enabler brings support to secure and private *OAuth2*-based authentication of users and devices, user profile management, privacy-preserving disposition of personal data, Single Sign-On (*SSO*) and Identity Federation across multiple administration domains.
- The **Wilma** proxy Generic Enabler brings support of proxy functions within *OAuth2*-based authentication schemas. It also implements PEP functions within an *XACML*-based access control schema.
- The **AuthZForce** PDP/PAP Generic Enabler brings support to PDP/PAP functions within an access control schema based on the *XACML* standard.

This division also brings Generic Enablers for the publication and monetization of context data resources, available through the core Orion Context Broker component of the platform:

- The **CKAN extensions** Generic Enabler brings a number of add-ons enabling to extend current capabilities of the world-leading *CKAN* Open Data publication platform to allow publication of datasets matching right-time context data, the assignment of access terms and policies to those datasets and the assignment of pricing and pay-per-use schemas to datasets.
- The **Biz Framework** Generic Enabler brings backend support to Context API/Data monetization based on open *TM Forum Business APIs*.

The following Generic Enablers are under incubation within this area:

- **Idra** is able to federate existing Open Data Management Systems based on heterogeneous technologies (e.g. *CKAN, SOCRATA, DKAN* etc.) providing a single API and a standard metadata format (DCAT-AP) to discover open datasets.
- **APInf** API Management Framework is a tool for API owners to manage their APIs. It provides all the necessary features to run business with APIs and makes it easy for API consumers to find and start using the standard APIs.

7.3 FIWARE Context Broker (Orion)

The most popular implementation of the NGSI interfaces (NGSIv2[14] and NGSI-LD[15]) is the *Orion Context Broker*[16] which uses MongoDB as its underlying data store. The figure below shows an overview of the architecture and functional interfaces supported by this component. It can be deployed using Docker in the most popular platforms.

The Orion Context Broker allows to publish, consume and subscribe to data coming from multiple devices and sources. In fact, it allows applications to get access to (harmonised) data entities, regardless data sources. The broker may store data in the short to medium term using a data store. The expected uses of Orion are:

- Retention of current instances of harmonized data entities processed from IoT devices and external sources (context data);
- Storage of a window of short term historical harmonized data entities that may be queried directly via a third party application.
- Storage of any results of Analytics and Intelligence results which become additional context data that can be queried or mashed up with other IoT data or external data sources.
- Register context provider (source) applications, e.g. a temperature sensor within a room
- Update context information, e.g. send updates of temperature.
- Being notified when changes on context information take place (e.g. the temperature has changed) or with a given frequency (e.g. get the temperature each minute).
- Query context information. The Orion Context Broker stores context information updated from applications, so queries are resolved based on that information.

[14] https://fiware.github.io/specifications/OpenAPI/ngsiv2
[15] https://forge.etsi.org/swagger/ui/?url=https://forge.etsi.org/gitlab/NGSI-LD/NGSI-LD/raw/master/spec/updated/full_api.json
[16] https://github.com/telefonicaid/fiware-orion

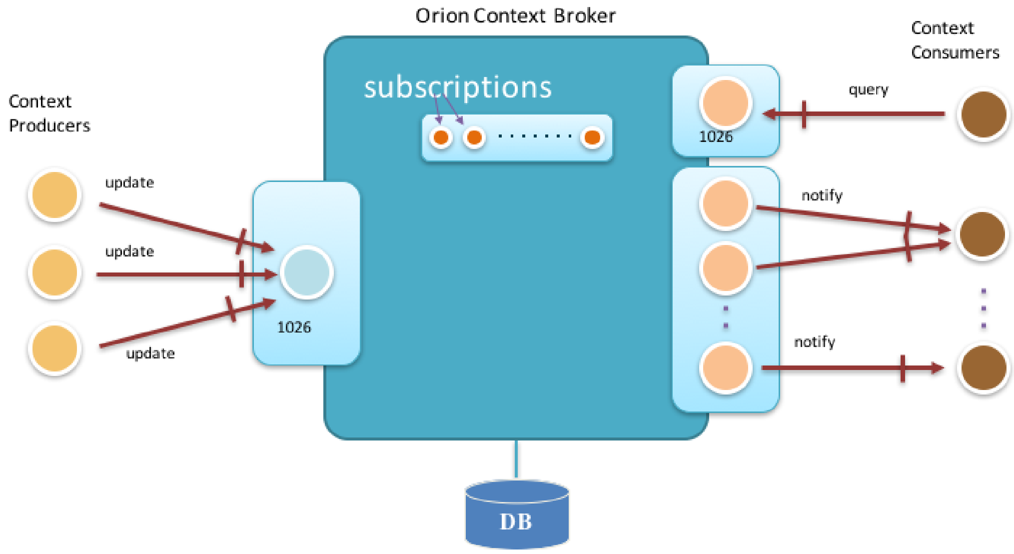


Figure 37: Orion Context Broker high level architecture

SW: Ok, that was a nice compact, up-to-date description of FIWARE, but what is the meaning of this Chapter in the context of this whole deliverable. I expect here a short paragraph that describes the context and meaning of this information for IoF2020.

8 365FarmNet Open Platform

8.1 Introduction

This chapter presents the new concept of 365FN open platform for the agri-food domain, resulting from the change of paradigm made, supported by the Open API development and marketplace. 365FarmNet is a platform that can be used to implement Smart Agri-food solutions compliant with the MIMs described by Chapter 3 and the System of Systems approach.

In the agricultural resource management, the information systems that integrate analysis and decision support tools have been under development for more than 30 years. Along with the intensive automation and specialization of the machinery, software systems have been incorporated as an essential tool for the producers of the new century.

With the rise of the digitalization, the use of sensors and mobile devices, and the growing network of data sources, producers have become more demanding: they are asking for faster and better services, more transparency and more user-specific solutions. For this reason, a FMIS should serve with more capabilities and functionalities to process, analyze and provide support for the farmers and growers, without compromise its performance. 365FarmNet has recognized that a better way to meet the current needs of the producers is to encourage cooperation and harmonization of the development to offer sustainable services, which adapt quickly and dynamically to the requirements of the users. Each stakeholder within an agri-food-ecosystem obtains direct benefits from the cooperative work such as the improvement of its own development process, but the more important thing is that the end user (a farmer, a grower, an advisor, a contractor) sees a robust and feasible ecosystem of solutions, which reacts in real time for his issues.

The following lines are intended to provide an overview of the mindset of open and cooperative development from the perspective of a FMIS as 365FarmNet.

8.2 Overview of platform

Traditionally, a FMIS gathers agronomic and administrative information from each production unit (field) to keep, among other things, a detailed historical record of the activities and tasks in the field during the crop seasons. The system also offers analysis tools, which allow the farmer to set calendars for sowing or harvesting, organize the mechanization of their fields, or management supplies and workforce.

365FarmNet has not been separated from this core use. 365FarmNet is a web-based farm management information platform (FMIP), which integrates the traditional functionalities of a FMIS with the new mindset of the cloud computing. The platform is developed under three main premises: prioritizing the user experience (easy/ready-to-use, customizable), offering high mobility and digitizing tools that are always at the farmers fingertips, and total independency providing high interoperability and compatibility.

The 365FarmNet user has as default a core features package developed by the 365FarmNet Team and to complement it, the user could find and choose extension modules developed by partners in the 365FarmNet module shop. The modules of the partners are classified into categories depending on the complementary covered feature: weather modules, seed-advice modules, plant protection catalogs, machine communication modules, livestock modules, etc. As an example: Farmers, who own a combined production system (livestock and crop), could find in the module shop Livestock-modules to register the activities of the herd, or to plan veterinary activities.

This customizable system, high modularity and free choice for the users of the platform is possible through the cooperative work between 365FarmNet and Partners, the 365Connect concept and the continuous building of a developer community focused on agribusiness.

8.3 Description of Open API and marketplace

Initially, 365FarmNet explored the establishment of partnerships in order to provide an “ad hoc”-development for the modules, which could be offered to the farmers and producers. However, with the growing demand for connections between services, the implementation of IoT solutions in the agriculture and the need to standardize and harmonize different data models and to optimize the integration of services, the platform decided to offer an Open API called *365FarmNet Connect*, following other application areas (smart cities or mobility networks).

The main goal is an easy and optimized integration process of new services within the 365FarmNet Platform and offer them directly in the 365FarmNet module shop to the end-users. The 365FarmNet Connect API has the following remarks:

- Integration is based on iFrame, that means, each partner can stay within their own technology stack
- The Open API provides information about grower/farmer, fields, crops and supplies.

The figure below shows a schema of the technical overview and the main components of the integration.

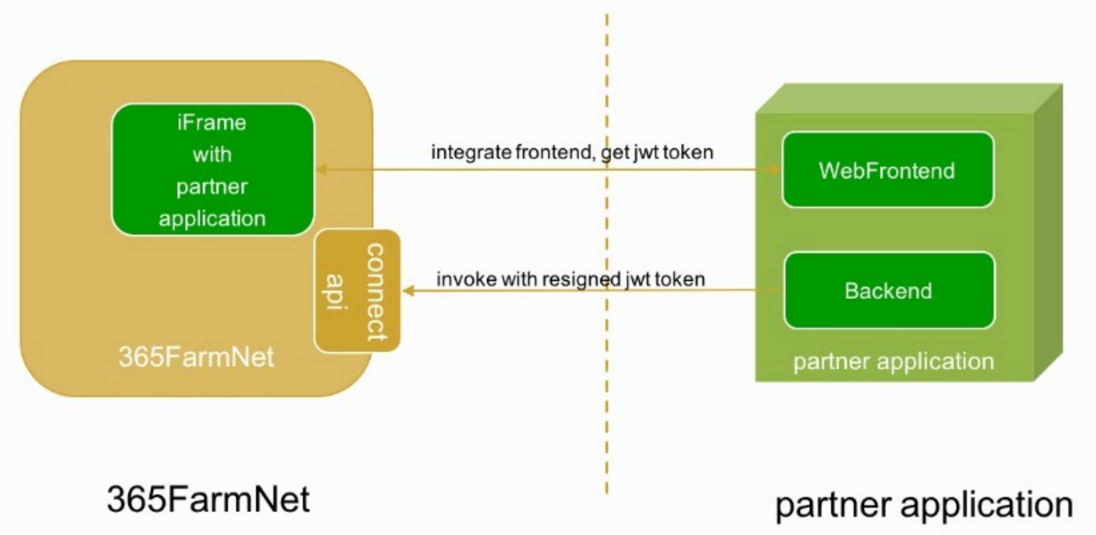


Figure 38: Technical overview 365FarmNet Connect (source: https://developer.365farmnet.com/site/#365connect-documentation)

8.4 Guidelines to develop APIs and connect to 365FN

A developer or potential partner of 365FarmNet needs to register and request permissions to the portal in order to access the test environment and explore the features of the platform. After the activation of the account, the account owner could initiate the integration of the partner's module web application at the technical level, before any contractual/legal activities. This first step allows access to the API gateway and test environment.

Partners or developers will get credentials (JWT-Token and a specific partner portal account) for the integration. Depending on the requirements of the development, 365FarmNet may configure two separate partner integrations with different URLs: one for local development and other for the live environment.

After the technical integration, the legal activities shall be compliant to: determine cost for customers, determine data permissions/data protection, exchange contracts between 365FarmNet and the Partners, establish new components and tiles in the portal. Finally, 365FarmNet activates the new component in an upcoming release and starts the marketing of the new module together with the partner.

8.4.1 iFrame Integration

The partner web application is embedded within the Portal through a Web *iFrame*. The partner application shall be available over HTTPs to grant transport security and avoid browser warnings for the end-users. The **iFrame** is accessible via a static application route:

https://<ENVIRONMENT>.365farmnet.com/365FarmNet/dist/index.html#/connect/<PARTNER_NAME>/<PARTNER_ROUTE>

Sometimes embedded applications could be refused by several browsers, because they are using cookies classified as third-party cookie. Encode values as URL session parameters instead.

8.4.2 Data exchange

365FarmNet Connect API gateway allows data exchange between the 365FarmNet Portal and the partner application, using JSON as exchange format. Each request to the Connect API needs to be authenticated with a JSON web token.

Currently there are two servers available for partners to request data. Each server provides a 365FarmNet portal instance and a Connect API gateway instance. The partner shall be sure to use the correct keys for each server.

8.4.3 Authentication

All requests to the Connect API gateway shall be authenticated with the JWT in the partner token. The token must be created and signed by the partner application; in addition this must contain another token provided by the 365FarmNet portal in order to identify the partner on every request. The mechanism is explained in the documentation of the Open API.

See: https://developer.365farmnet.com/site/overview/authentication/

8.4.4 About the REST API

Each environment within the 365FarmNet uses its own Connect API gateway. For this reason, the partner application should expose this gateway URL from the connect token instead of using a fixed URL. The API URL could be found in the attributes of the token payload.

Most REST resources provide only active data by default. However, the partner application can request inactive data, but it is not allowed to request only inactive data. For more details is recommended to consult the documentation.

The following list shows the current available REST endpoints (this is a functional description, for technical details it is recommended to consult the documentation in the developer portal):

Endpoint	Description
Company	Tenant for all other company related data. Each dataset belongs to one company and is not shared among other companies. Multiple user can exist. A company should to have at least one farm and one person
Crop Cultivation	Main Crop, Catch Crop, Permanent Crop A main crop is normally seeded and harvested within one year. Variety and expected product is provided
Fields	Fields are production units. Operations are related and referred to a field. Fields can be merged or split them. A field can only have a culture/crop. Boundaries are provided
Guidance Lines	A guidance line (aka reference line) is a geometric line inside a field. Every guidance line is related to a specific field. Used by terminals and machinery.
Cultures (= Crops)	A list of the available crops within the platform (a catalogue of crops)
Seeds	A catalogue of seeds used as start material of a crop

Endpoint	Description
Fertilizers	A list of active fertilizers
Farms	A list of the Farms registered for a company. By default, only active farms will be listed.
Images	Images of machines. This is for documentation and identification purposes.
Persons	A list of active persons. Persons can also be retrieved by a query using person-id parameter.
Machines	A list of machines Each machine has a unique number within the company account. Same applies for the serial number.
Operations	Send planned bookings or complete bookings to 365FarmNet. Minimal Requirements: • Type of activity • Reference to one or multiple fields Optional: • Machines • Persons • Supplies • Attachments like notes or maps Formats: • Exchange-Format (365FN-format) • ISOXML (in progress) • ADAPT (in the future)

Same here: what is the meaning of this Chapter in the context of this whole deliverable. I expect here a short paragraph that describes the context and meaning of this information for IoF2020.

9 Service and Data Monetization

IoF2020 use cases are realising functional components or using specific external components that are either hardware components (e.g. weather stations, actuators, cameras, or low bandwidth communication) or software services (e.g. geographic information systems, weather services, equipment manufacturer cloud services, or decision support systems). The provision or consumption of services and data requires transparent conditions for both sides that are usually agreed in specific licenses or individual agreements. After having agreed to provide or use the related offering, both sides need to support related interfaces. As highlighted in chapters 5, 6 and 7, there are diverse aspects from data models up to specific functional components that facilitate interoperability. However, when aiming at a straightforward provision and usage of services or data, one can aggregate numerous features for minimising the development effort. Therefore, WP3 partners ATB, CORIZON and ficodes have been working on the following open platforms:

- ***Service Monetization***, as outlined in the following section 9.1, enabling providers and users of services to offer and also use different services, while assuring that no details of a business interaction are stored in a central system, but the history of service usage as well as to assure a proper authentication and authorisation of service access.
- The ***Data Marketplace,*** as outlined in the following section 9.2, is representing a complete framework, which supports data management, including the means for visualizing, publishing, and monetizing data while enforcing access control and managing data usage terms and conditions.

Those key reusable components can be considered as enablers that are also generally compatible in their usage with the other open platforms as outlined before.

9.1 Service monetization: CoatRack

IoF2020, WP3 was analysing the different use cases and expects that some 11 use cases would potentially benefit by evolving their solution towards integrating service monetization, either to provide their own services or to consume services offered by third parties. Therefore, the IoF2020 partner organisations ATB and CORIZON joined forces to develop the so called “CoatRack” solution enabling service monetization.

In general, use cases and any other external party can use CoatRack to offer software service APIs to developers via a trusted framework for service monetization and access control. As presented in the following schematic Figure 39, CoatRack facilitates the service provision and access control as well as takes care for service monetization. The typical users of CoatRack are software developers offering and/or consuming software services as basis for additional applications for specific end-users.

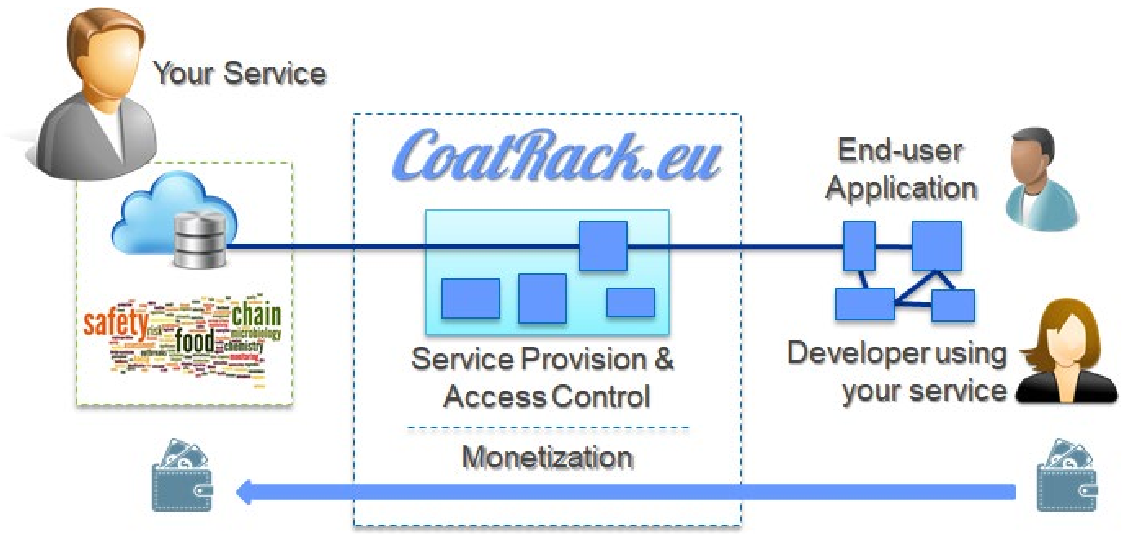


Figure 39: Schematic interaction of service providers and consumers, facilitated by CoatRack.

The term “service monetization” for characterising CoatRack generally summarises a key purpose. However, describing CoatRack in more detail, we can categorise the main groups of features around the following features:

- Offering software services.
- Visualising the access to services to enable an understanding of users’ interest.
- Enabling to share ICT (infrastructure/ overhead) costs inside an organisation, both for internal as well as external services.

The provision of access keys to customers and/or partner organisations is the basis for the monetization of available services as well as facilitating access control. According to the selected terms for payment, CoatRack will track service consumption and enables the monetization. CoatRack helps to define the payment model, to reflect the selected business model for monetizing a single software service or a solution package. Therefore, the current implementation enables flatrate and pay per access models. As presented in the following Figure 40 – in a scenario of a larger company – backend services using e.g. internal systems usually not exposed to external parties or business partner organisations can be monetised. However, by providing a managed access, the provision and revocation of access rights is highly facilitated.

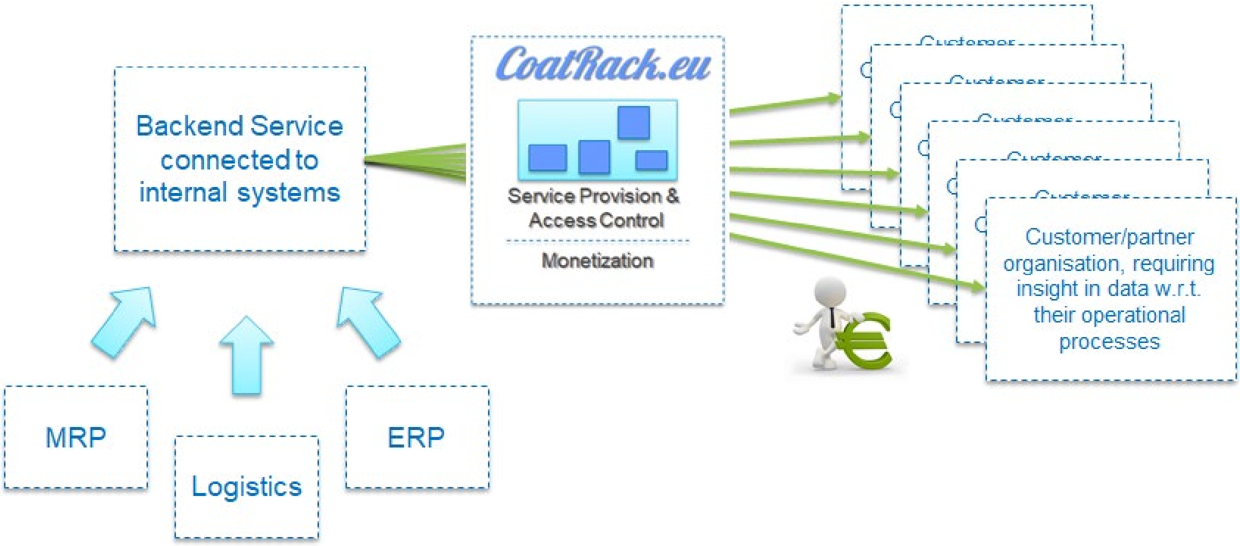


Figure 40: Offering services to customers or partners.

Besides the basic feature of monetizing services, CoatRack is also enabling to visualise the access by internal or external entities using the service(s). This can help to validate the acceptance of the offered services and evaluate the preference for specific service alternatives. At the same time, CoatRack can visualise the access over time, enabling to understand the service usage behaviour when considering certain time, user and location context. The following Figure 41 is outlining this basic feature.

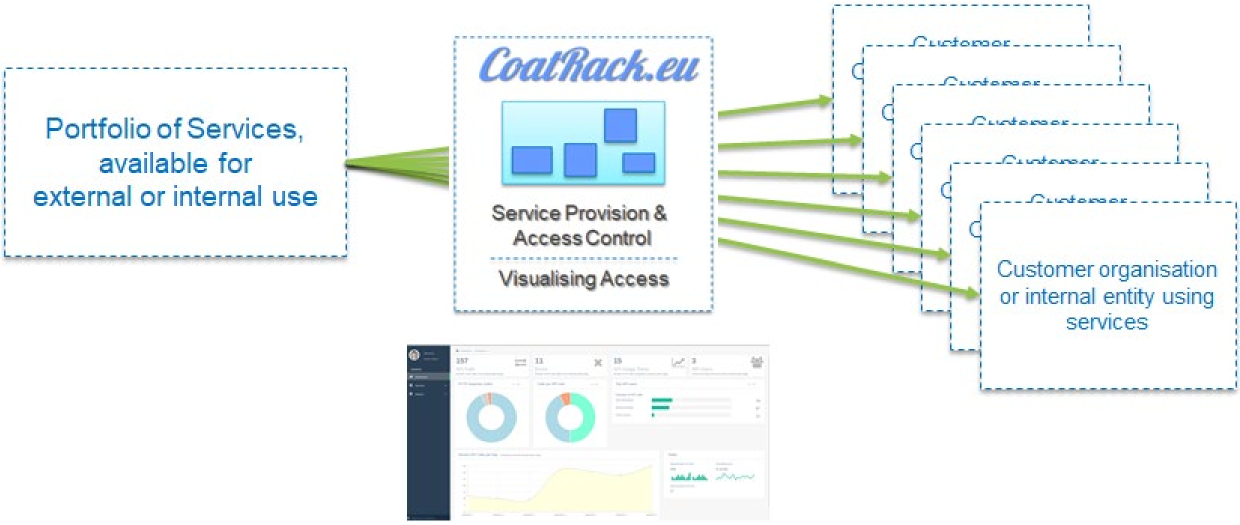


Figure 41: Understand needs and interest of customers or internal entities.

The development of overall software systems is often using external services that can provide specific features that would be rather too expensive when developing on its own. At the same time, the integration can be realised straightforward, when consumed from an external service provider. As presented in the following Figure 42, an example is the usage of Google Maps as an external geographic information system (GIS). It can be used for showing specific maps, information about places as well as visualise information items at specific locations in the map. The Google Maps API is offering diverse interactive features. As soon as such an external service is integrated in a backend service used by diverse company internal departments or profit centres, it is rather difficult to assign related costs to the usage of that external service. As for the example of Google Maps, as soon as the basic free offering is exceeded, significant costs can occur that might be even not transparent for the using entities. Therefore, CoatRack can be a straightforward feature to assign such costs to different service consuming applications.

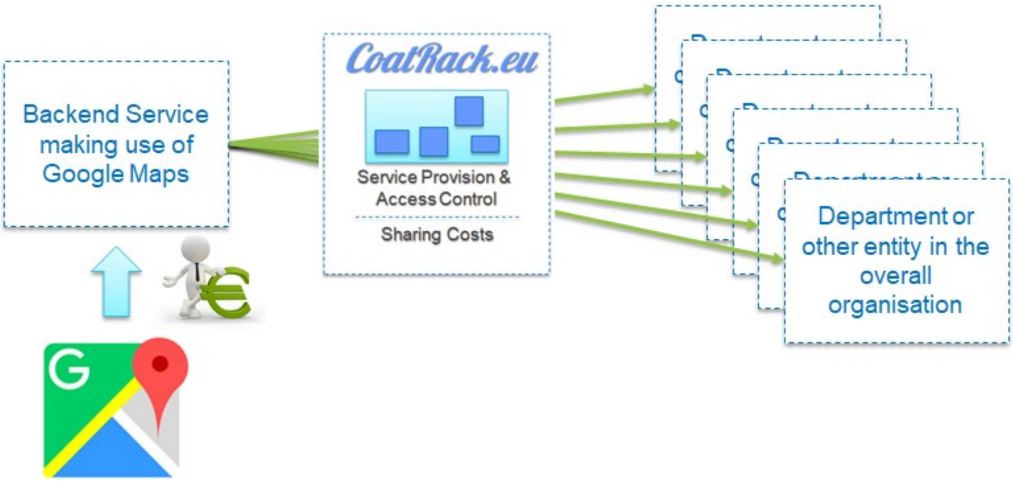


Figure 42: Sharing of ICT related costs inside an organisation.

As highlighted above, CoatRack is able to support monetisation and visualise usage of backend services as well as providing required information for being able to share the costs of service usage within an organisation. However, it needs to be highlighted that CoatRack is neither receiving nor forwarding the payload of the communication between services. A local service gateway that is usually installed in the backend service infrastructure, is only taking care for the authentication, authorisation, configuration and tracking of the usage statistics.

The following Figure 43 shows a typical architecture. The calls to the provided software service API are routed and logged by a custom CoatRack Gateway, which is installed in the local network. Configuration and statistics are accessible via the CoatRack website.

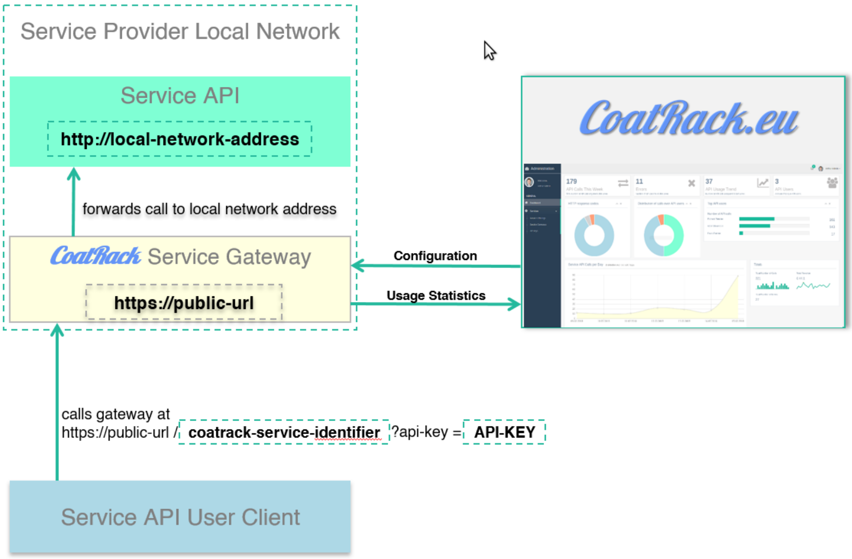


Figure 43: General architecture overview of CoatRack for Service Monetization.

A tangible example for using CoatRack is use case 3.4 Intelligent Fruit Logistics in IoF2020, where customer locations are centrally managed, while being accessed by different organisational entities. The application LocMan (i.e. which stands for location management) is offering diverse features for end-users, while it requires access to different backend services. By using CoatRack, it is possible to understand which back-end services are used and assign the usage to the specific organisational entities. Of course, the idea is not to classically monetise those services, but being able to plan ICT budgets accordingly, while also being able to understand and validate the end-user acceptance for specific services.

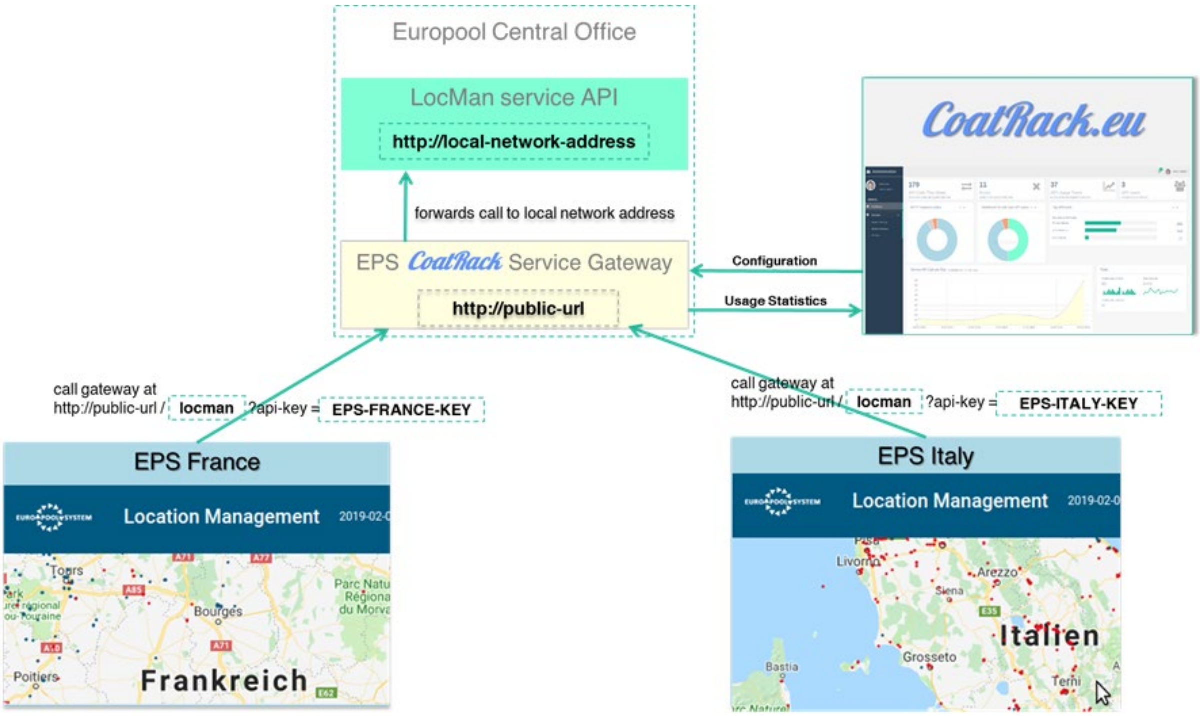


Figure 44: Example for sharing ICT costs and visualising access behaviour in the scope of the example in UC 3.4 which uses LocMan as Service for EPS internal users.

CoatRack is still under development and evolves in accordance to its usage and the related user feedback. The main features of CoatRack are described in the following:

9.1.1 Define Software Service Offering

An installed software service is registered in CoatRack. The service provider is entering the related URL and a meaningful identification of the service. This will help consumers of that service to understand the purpose of the backend service. Moreover, the provider is determining the potential payment options for being able to monetise its usage (i.e. currently a monthly fee, pay per call or free). At the same time, the provider can define, if the service will be available as a public offering, or being restricted only on invitation for potential users. The following Figure 45 is presenting a screenshot of the CoatRack user interface at the step for defining the payment options.

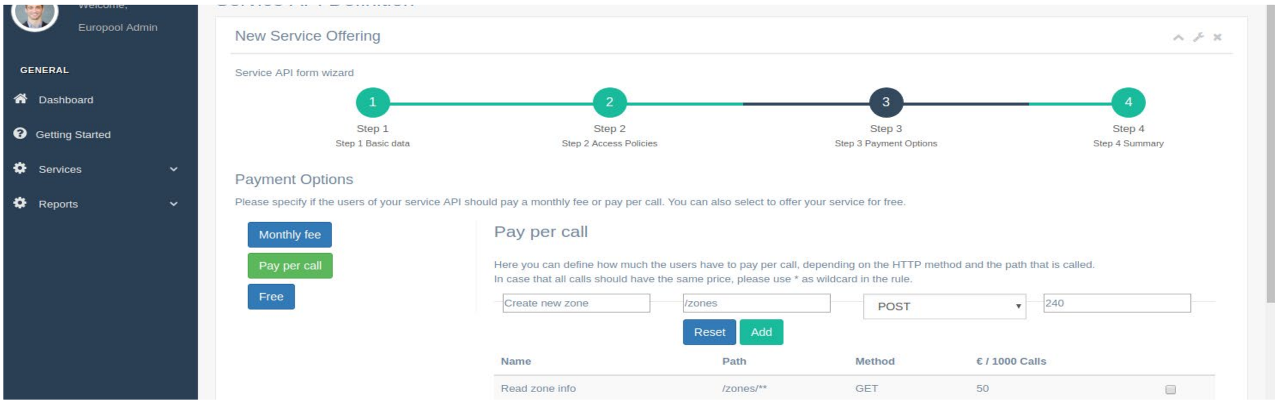


Figure 45: CoatRack Feature – Define Software Service Offering.

9.1.2 Generate Gateway

For being able to connect a local service to the central CoatRack instance, the service provider would need to generate and download a gateway that would be installed on its local server. Each time, CoatRack is generating an individual gateway that is matching the related service provider and backend service. The following Figure 46 is presenting the UI for generating and downloading a new service gateway.

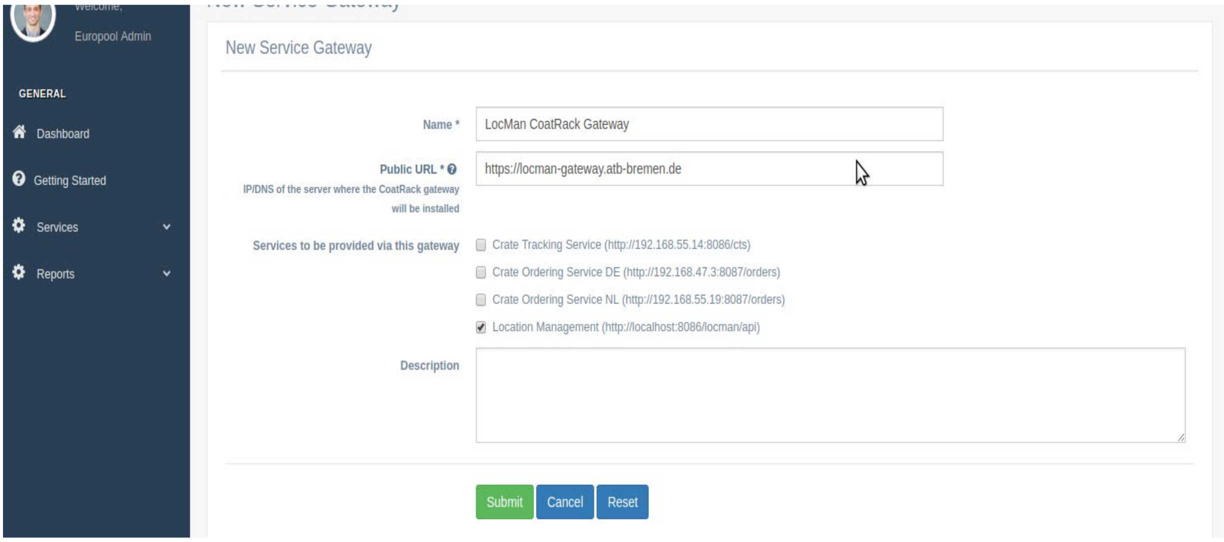


Figure 46: CoatRack Feature – Generating a new service gateway.

9.1.3 Software Services Catalogue – Provider

All services offered by a service provider can be displayed in the CoatRack overview for managing the service offerings. It offers to list all related details, edit or delete an offering as well as to start the procedure for creating an additional service offering. The following Figure 47 is presenting the overview of software services offered by the service provider.

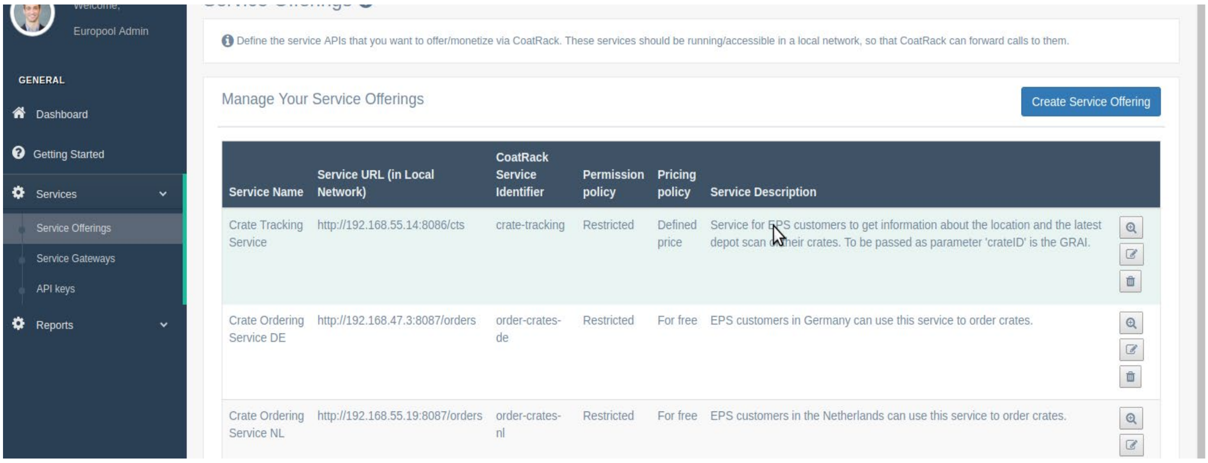


Figure 47: CoatRack Feature – Software services catalogue of a service provider.

As highlighted before, the offered services are usually running in the local network and are connected to the CoatRack gateway that enables their publishing in accordance to the access policy. For being able to download the individual service gateways, CoatRack is offering an overview of all gateways that were generated for the individual services. The following Figure 48 is presenting an overview of such service gateways, enabling to download, display all detailed information, edit and delete them.

Also, users of services can display a listing of available services. They can see all the services offered without any restriction, as well as those services where they received an API key from the service owner.

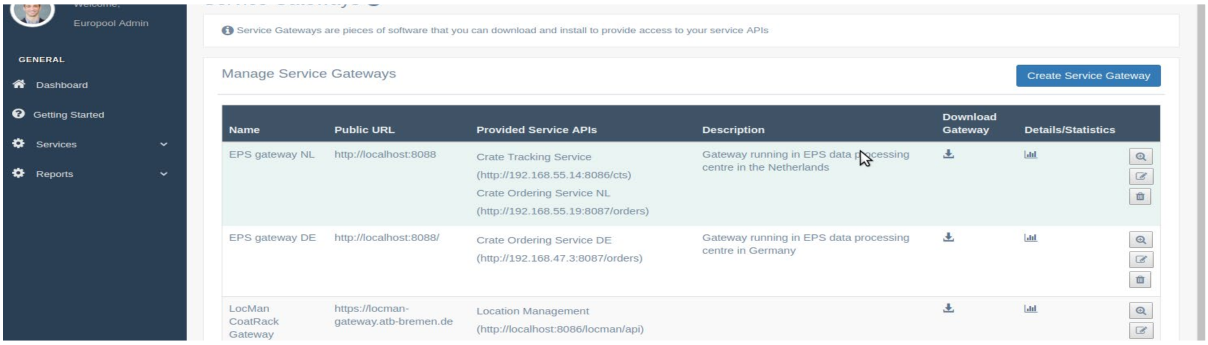


Figure 48: CoatRack Feature – Gateways of a service user

Based on this mapping, CoatRack is enabling an offering and usage of services in an N-to-M relationship. One provider can offer several services to many users, while one user can make use of one or several services from one or several providers.

9.1.4 Usage Logging

CoatRack is logging the usage of the offered services. Besides plain access numbers, CoatRack is also tracking the type of calls as well as identifying potential issues when a service is accessed. In its basic version, CoatRack is offering a standard overview as presented in Figure 49. If required, this can be further customised and extended on an individual basis. However, in the scope of IoF2020 WP3, ATB and CORIZON are discussing potentially interesting data with the related end-users. This will be the basis to further evolve this overview.

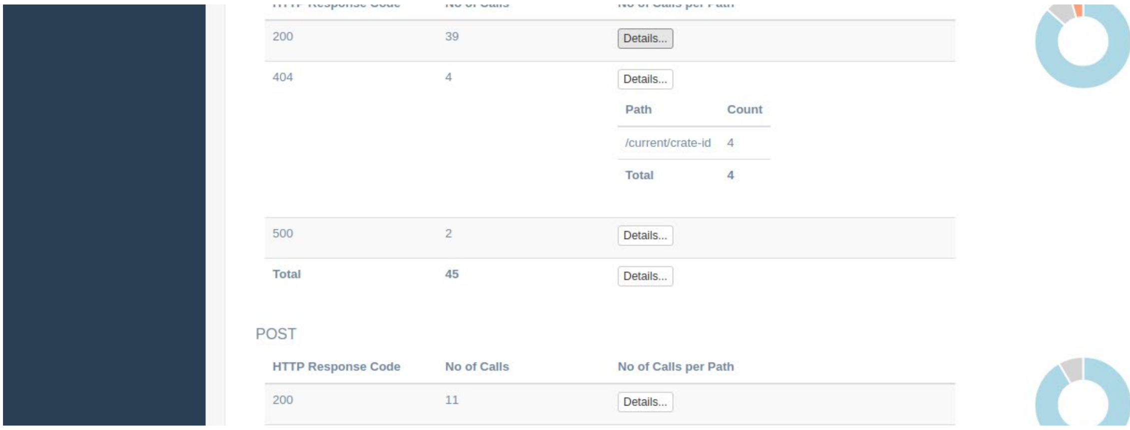


Figure 49: CoatRack Feature – Logging the usage of software services

9.1.5 Service Usage and Monetization Statistics

CoatRack is also offering a basic dashboard displaying key performance indicators about the offered or consumed services. The following Figure 50 is presenting this overview that is a mix of graphical and numerical information. Also this overview can be customised on specific user demand, while currently being validated in the scope of IoF2020 WP3. As the CoatRack offering is also available to the public, also project external stakeholders are invited to give feedback and to collaborate with ATB and CORIZON.

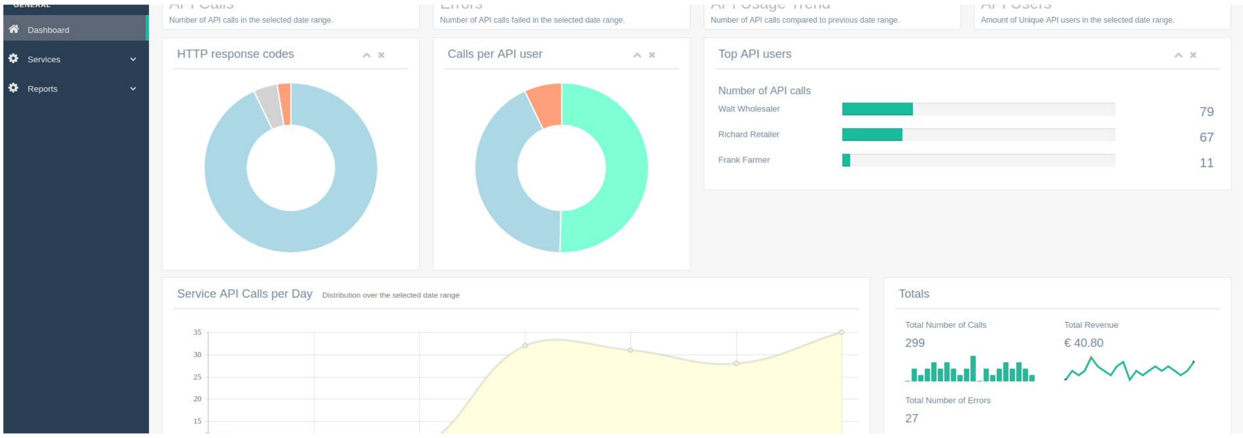


Figure 50: CoatRack Feature – Service usage and monetization statistics.

When summarising the main characteristics, one can highlight the following key features:

- Monetization of software services by APIs
- Offering pay per use and flatrates with public and restricted access
- Dashboard for service access statistics
- Multi-tenant access control
- Authentication & Authorisation
- Enabling payment processing

The hosting of CoatRack is described in Deliverable D3.8 that enables an easy set-up and immediate usage of CoatRack. Commercial usage of CoatRack outside the scope of the funded IoF2020 project can also be arranged on fair, reasonable, and non-discriminatory (FRAND) terms without up-front costs. On top of that, the pricing is limited to the usage of services. As long as no service access is generated, costs shall not occur. Therefore, a freemium offering outside the scope of IoF2020 is offered. CoatRack is currently available via the following URL: https://www.CoatRack.eu

9.2 Data Marketplace and Dashboards

In the different smart domains, and in particular in the context of the agri food domain handled in IoF2020, data is a cornerstone for the different business models. Use cases are interfacing with a large amount of data sources, not only IoT sensors but GPS, drones or robots operating in the farm as well as data coming from monitoring process within the supply chain. The result is a huge set of data that needs to be managed, maintained, exchanged, published and, ideally, monetized.

It is clear that providing the means for such data management as reusable components is of key importance within IoF2020 in order to unlock the potential of the generated information within the different use cases. In the context of WP3, ficodes is providing a complete framework to support such data management, including the means for visualizing, publishing, and monetizing data while enforcing access control and managing data usage terms and conditions. Figure below depicts such data management framework.

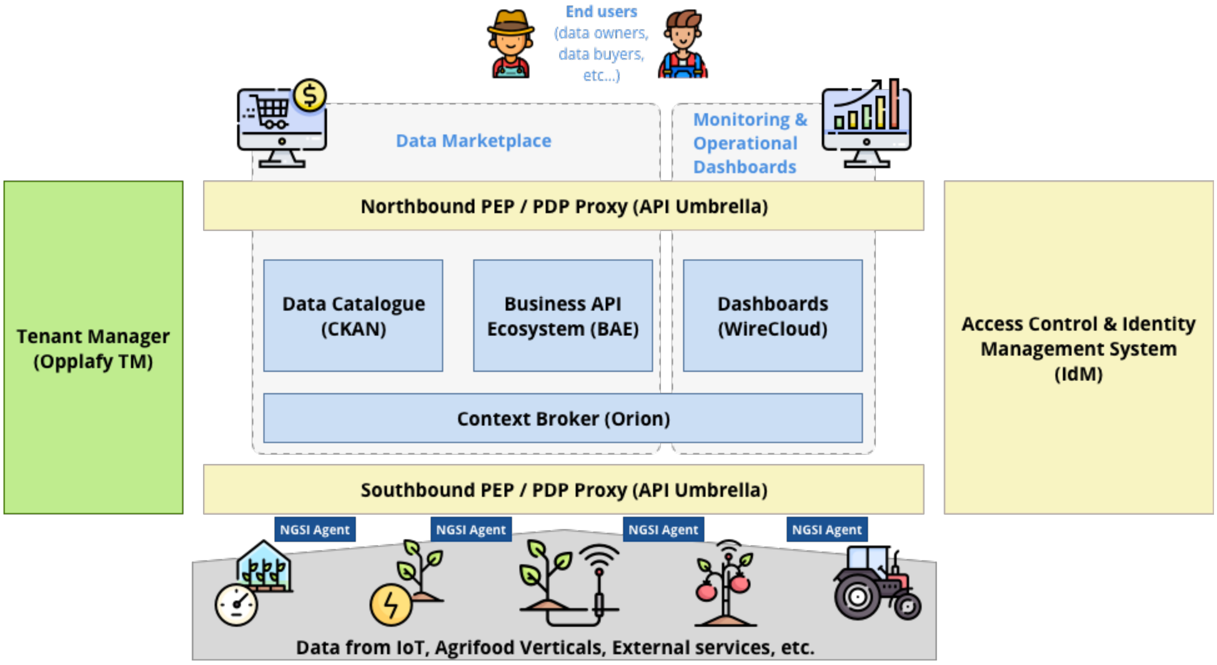


Figure 51: Architecture of the data management framework.

As can be seen in the picture, the different components are divided into two main reusable components highly integrated within a FIWARE-based architecture. On the one hand, the **configurable dashboards** component provides an adaptive tool supporting the creation of rich data visualizations and operational dashboards using a graphic approach. Such tool enables farmers and non-technical farm workers to monitor IoT sensors, actuate upon them, and monitor the progress of processes within the farm. This reusable component has been created using WireCloud, the *Application Mashup GE* of the FIWARE platform integrated with FIWARE NGSI data format by design.

On the other hand, the **data marketplace** reusable component provides the capacity for exposing and monetizing data, enabling data owners to establish the particular terms and conditions that need to be satisfied and without the need of uploading the data to a third party software; thus, enabling them to maintain control on the data. The data marketplace component is implemented by jointly using two different GEs of the FIWARE platform: *FIWARE Extended CKAN* and the *Business API Ecosystem (BAE).*

It can be seen that both reusable components rely on the FIWARE security framework for the management of users, roles and access policies needed to enforce access permissions and support the monetization of data. Such security framework is made of the FIWARE IdM managing users and roles, and the PEP/PDP proxy handling policies, securing access to data and monitoring user access.

Finally, in order to simplify IoF2020 use cases and organizations to be integrated into the described data management framework, FICODES is providing a Tenant Manager component which enables them to create isolated data tenants, hiding the complexity of interacting with the different framework components.

Using these tenants, data owners are able to establish data access permissions within its own managed data infrastructure, enforcing data privacy and giving organization owners control over its published data.

9.2.1 Data publication and monetization: Data Marketplace

Publishing and consuming data is a key aspect for the development of applications enforcing the innovation in agri-food systems, while having a common framework for such data publication and consumption materializes the System of Systems approach sought in IoF2020.

In this context, FICODES provides the FIWARE-compliant Data Marketplace reusable component made up of the BAE working jointly with the FIWARE extended CKAN and taking advantage of the core FIWARE components, such as the Context Broker and the Security Framework.

The BAE (see figure 61) is the needed component for handling monetization. This component is a complete market solution which supports the monetization of digital assets, providing the means for product and offering creation and acquisitions management. Within the scope of the data marketplace, the *BAE* is focused on data monetization, though it is able to monetize any kind of digital asset using a plugin-based approach. By using its portal, data owners can:

- register data offerings providing all the market-relevant information, including terms and conditions, SLAs, custom characteristics, or data relationships
- force customers acquiring access to the data to accept the terms and conditions, even in those scenarios where the data is provided for free but specific clauses need to be satisfied in its usage (e.g. not commercial use, etc.)
- charge users for accessing their data, depending on their business model. There are different payment models already available:
    - One-time payments: users pay only once.
    - Recurring payments: users pay periodically (monthly, yearly, etc) for getting access to some data. In addition, users will be able to cancel the subscription, but they won't be able to access data anymore.
    - Usage payment: users pay per use. Their payments are based on the amount of information consumed.
- provide end users with search, discovery, and revenue sharing features.

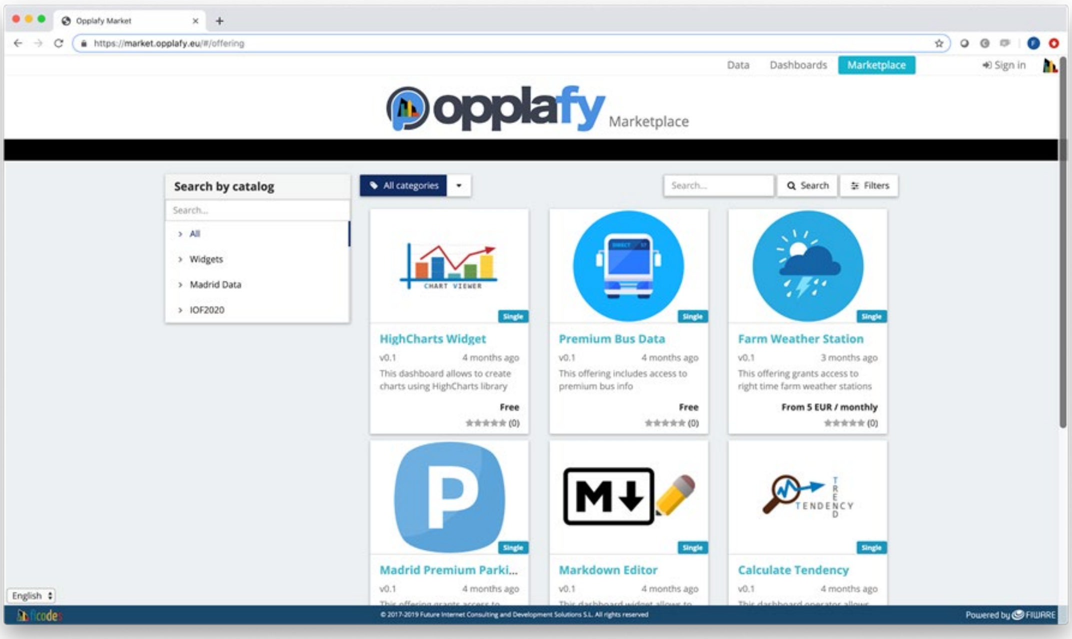


Figure 52: BAE main catalogue.

The Business API Ecosystem is fully integrated with the FIWARE Extended CKAN, which is a widely adopted data publication platform to catalogue, upload and manage open datasets and data sources. In this context, Ficodes provides a couple of extensions that support enhanced data management capabilities and integration with FIWARE technologies including NGSI. In particular, data publication and discovery features provided by CKAN have been enhanced with the following features:

- **Real (right) time data publication**. (see figure 62). CKAN does now speak the NGSI protocol, being able to publish queries to a Context Broker as data resources. That means published datasets follow harmonized/standardized data models and data is stored only in the Context Broker; thus, not needing to replicate the data or loosing control over it.
- **Identity Management, Access Control and Security**. CKAN has been extended to allow site visitors to login using an OAuth2 server. FIWARE's IdM component can be deployed to be used as the identity provider, leveraging the single-sign-on approach required for the monetization and access control management of CKAN datasets. CKAN users can also control the visibility of their datasets. They can create public, private and protected ones. A protected dataset features access lists to select users that are able to get access to it. Finally, the context broker is usually deployed in the backend, and for all those scenarios where datasets are not completely open, or maybe monetization is needed, the context broker instance is secured by using the security framework PEP/PDP proxy functionality.
- **Enhanced Data Visualization**. (see figure 63). CKAN leverages *WireCloud* (Configurable Dashboards component) to create adaptive and incremental data visualization. This way, farmers (sellers) can decide the way they want their data to be shown.
- **Data requests**. Farmers can also use CKAN to request data that is not currently published. In this way, this provides the CKAN based Marketplace with another possibility for trading with data.

Finally, both the BAE and FIWARE extended CKAN are fully integrated and able to support different use cases and needs. In this regard, BAE supports the automatic publication of private datasets in a FIWARE extended CKAN instance using the information included in data products and offerings, while FIWARE extended CKAN supports the creation of products and offerings in the BAE using the information included in a private dataset.

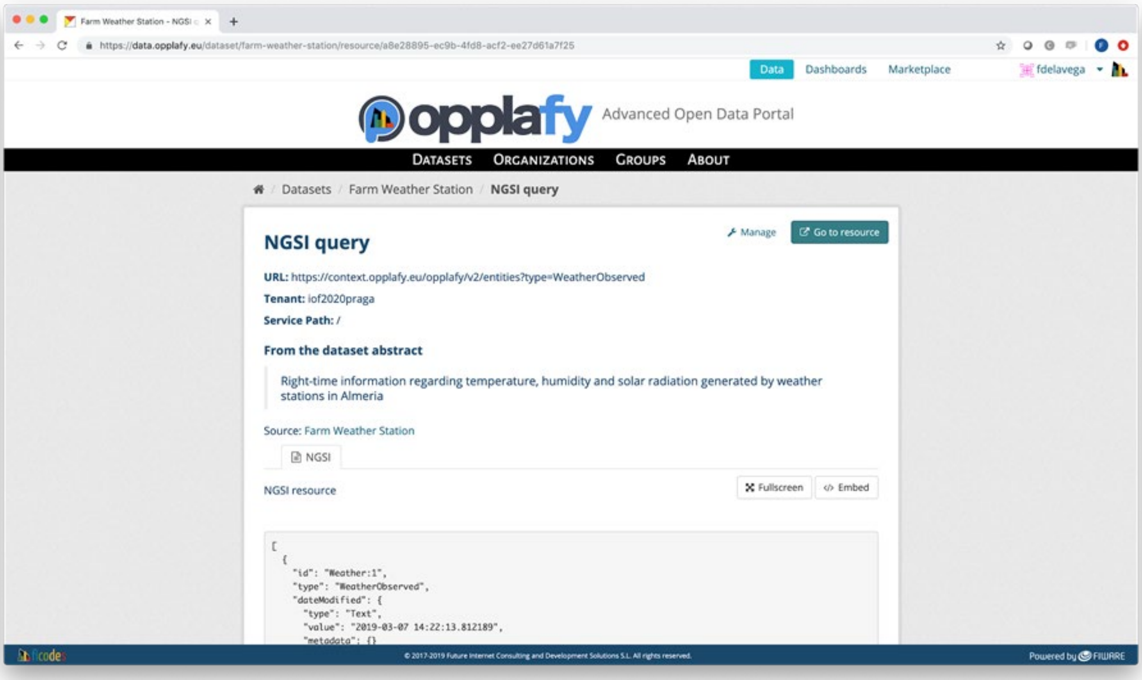


Figure 53: Real time data publication

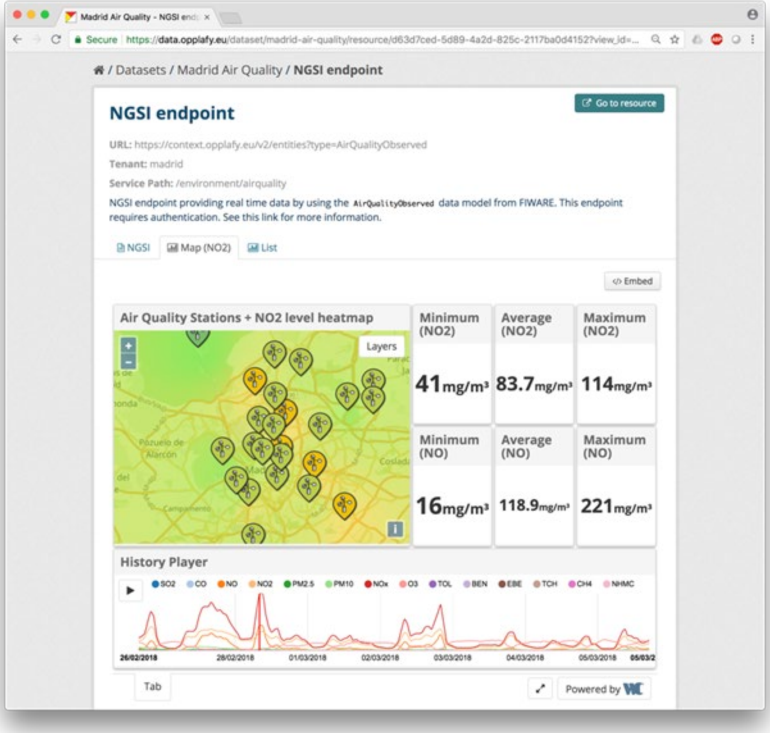


Figure 54: Example of CKAN Enhanced data visualization.

9.2.2 Operational Dashboards and data visualization: Configurable Dashboards

ficodes provides *WireCloud* as a reusable component for those use cases in need for adaptive and incremental visualization features, such as configurable monitoring and operational dashboards.

WireCloud is the reference implementation of the FIWARE Application Mashup GE. It is a web mashup platform aimed at creating web applications in an easy way by means of two kinds of web components (*widgets* and *operators*). It also offers an API to different services (e.g., REST APIs), besides several out-of-the-box features such as the wiring tool, in which widgets can be “wired” to share/flow data among them, a layout management to place the widgets just by dragging them, or a user’s management system leveraging different authentication protocols such as OpenID, OAuth2.0, etc.

WireCloud provides agri-food stakeholders with a framework for building web-based monitoring and operational dashboards, able to provide adaptive and incremental data visualizations to fit their needs:

- By adaptive visualization, we mean an attempt to improve visualization by incorporating adaptation, i.e., the ability to change visualization depending on various user features that can be explicitly provided or inferred from the trace of user actions. Through adaptation, users can modify the way in which the system visualizes a collection of elements: they can adapt not only the used visualization method by choosing between plot charts, tables, pie charts, sector graphs, bar charts, Euler diagrams, and so on, but also can make use of visual structure adaptation methods to adapt the structures either by varying the visualization layouts or by providing easy exploration methods. Dashboards follow this idea. *WireCloud* allows users to visualize data in different ways, allowing the modification of the visualizations on the fly.
- Regarding incremental visualization, it refers to an iteratively refining visualization process that adds or updates visualization details in a stepwise effort to complete a dataset’s visual representation. Completeness is understood with respect to the dataset and to its processing. Once completeness is achieved, the visualization process terminates. When completeness cannot be achieved (e.g., infinite streaming data or non-converging computations), the process can also terminate when it is guaranteed that none of the remaining unprocessed data and none of the remaining iterative processing steps would visibly change the outcome any further: the visualization is sufficiently stable. If

stability cannot be achieved, incremental visualization becomes an infinite loop that continuously modifies the visual representation.

WireCloud offers two different perspectives of use depending on whether the user is the final user (e.g. a farmer controlling their crops) or a web developer (e.g. a technical user with html/css/js skills):

- As a developer, *WireCloud* provides cutting-edge web technologies for easily developing of mashable application components, namely widgets and operators, and connecting them to backend services (e.g. support for pub/sub, “connectors” to a number of FIWARE GEs, etc.) and data sources (e.g. data APIs). Once created, *WireCloud* fosters their shareability and reuse by end users.
- As an end user, *WireCloud* makes easier the development of application dashboards without the need of programming skills. These dashboards are created visually by mashing up widgets and operators, i.e. integrating heterogeneous data, application logic, and UI web components to create value added “instant” applications.

Figure 55 below shows what a greenhouse management dashboard looks like.

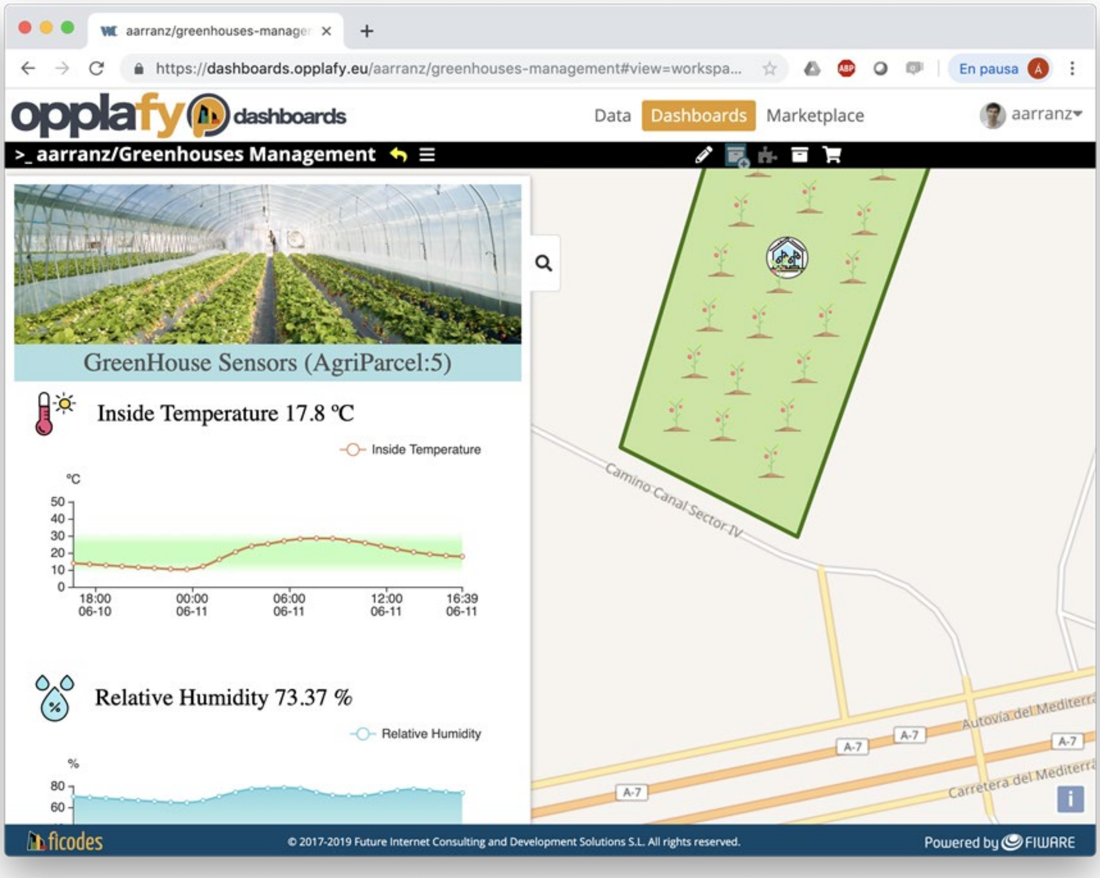


Figure 55: Example of *WireCloud* dashboard: Greenhouses management.

The main features of *WireCloud* are the following:

- **Easy creation of dashboards**. Usually, dashboards are built by domain experts by mashing up widgets and operators that were previously developed and shared by other developers (maps, data tables, charts, …), and then are used by end users, such as farmers or winery owners, to check their data in real-time (depending of course on the data source), control what is happening, and take instant actions in case of incoming events. In case a new widget or operator is needed, as they are based on web technologies it should be quite straightforward to develop.
- **Monitoring and operational dashboards**. Dashboards typically show different type of data of interest for their users. But *WireCloud’s* architecture also allows users to actuate upon their devices, e.g.

to switch on the watering system when the soil is getting dry. Moreover, *WireCloud* splits business logic implementation between widgets and operators allowing to change the behaviour by means of combining them in different ways. For example, it is possible to use the same map widget using data coming from different data source operators or sending commands to different services (e.g. context broker, REST API interfaces, ...) by means of selecting different target operators.

- **Fully customization.** *WireCloud* allows rapid configuration of almost any parameter needed to provide fully customized and functional dashboards:
    - It provides support for theming, as well as customizing logos, branding...
    - It features adaptive visualization by enabling users to configure their widget layout in a flexible way: widgets can be placed using a grid, in the background, leaving them floating, on sidebars, etc.
    - It supports internationalization and provides the tools to allow widgets and operators to leverage this feature.
- **Library of widgets, operators and mashups**. One of the advantages of using WireCloud is the availability of an extended library of widgets, operators and mashups to support all kind of interactive visual data exploration. Use cases stakeholders will find not only basic charts, but heat maps, graphs, geographic maps and other ad-hoc visualizations based on the requirements of the use case in particular. Each use case has unique requirements, and the dashboard will also provide visualizations (widgets) that will convey the specific data models of the datasets they represent.
- **Real (right) time capabilities**. Data is a moving target, and *WireCloud* is ready to deal with "live" data, leveraging an iteratively refining visualization process. Supporting real-time events (through its NGSIv2 API) and streaming data will expand use cases where organizations will leverage the streaming data generated by devices, sensors and people in a connected world in order to make faster decisions.
- **Easy integration with other reusable components**. *WireCloud* provides out-of-the-box integration with other reusable components. It fully supports the NGSI protocol, being able to request context data on demand and get subscribed to real-time changes on it. *WireCloud* dashboards/widgets can be used in both CKAN and the Data Marketplace to show ad-hoc data views. *WireCloud* users can be managed from an Identity Management and Access Control System making use of their OAuth2 API. Sensor data from the smart farming can be gathered from the context broker, or developing harvesting operators connected via gateway to the deployed IoT networks.
- **Sharing dashboards**. It is easy to configure who can or cannot access dashboards in a flexible way, allowing, among other things, to share dashboards with a specific list of users/groups or making it fully public. Moreover, *WireCloud* is able to offer embedding visualizations in other external web applications (i.e., those which farmers can be using right now). This will help to extend data content in the farmer context, adding extra value to the visualization itself.

In addition, ficodes provides a catalogue tool which manages the lifecycle of *WireCloud* inner components (widgets, operators and mashups), supporting its publication, maintenance, search and discovery. This catalogue supports federation, enabling organizations to have their own private instances or using global ones.

ficodes is working on providing an initial set of components for handling the visualization of the Data Models generated in the IoF2020 context. Those components will be integrated with the generic widgets (maps, charts, etc.). This will also requires evolving current NGSI support provided on WireCloud, to support the brand new NGSI-LD standard. This task is expected to be done by late 2019.

10 Conclusions

This report has facilitated an understanding by different stakeholders (particularly use case owners and developers) how Open Platforms and their reusable components can be extended, configured and leveraged to implement solutions in the domain of Agri-Food, with a view to creating interoperable and portable solutions. Such solutions are aimed at creating a rich marketplace and ecosystem for smart Food and Farming, representing an unprecedented opportunity to produce value and create business opportunities, by applying data-driven solutions. In addition, it can help stakeholders from other business domains to identify potential cross-sector implementation synergies. Ultimately this will allow building of an IoT ecosystem around the IoF2020 project and ultimately around Smart Farming in Europe.

The Open Platforms considered by this document address different functional and non-functional requirements and use cases (as described by **D3.7**) that have to be tackled when deploying a Smart Food and Farming solution, in accordance with the interoperability points (MIMs) described by **D3.3**. The Smart agri-food MIMs are similar to the ones present in other domains, even though, the digital farming domain has its own particularities, for instance the usage of field machinery, or the role of animals in production processes. Nonetheless, the opportunities for collaboration and synergies between domains are feasible and promising, particularly with *Smart City and Industrie 4.0*.

Concerning MIMs in the upper layers of the architecture, particularly the Context Information Management Layer, NGSIv2 and the emerging **NGSI-LD** specification (NGSI extended with linked data capabilities) are well positioned to be the technological standards needed for offering a consolidated view of information that is high quality and actionable, and which origin can be IoT devices or other sources of information, such as open data portals or external services. However, there is still a challenge in the massive adoption of these technologies and concept, as there are many smart farming solutions that still use their own cloud-based solutions relying on proprietary APIs and data models. To overcome this issue, it is of paramount importance, to promote the adoption of a common information management layer, especially for the sake of interoperability and integrability between smart farming solutions and, future-wise, for the development of multi-sided markets, where different smart domains cooperate by sharing and exchanging data towards a common goal.

To that aim, the availability of shared, well-adopted information models is a key interoperability mechanism for enabling a global market for IoT-enabled Digital Farming. Such models provide an essential element in the common technical ground needed for standards-based innovation, by making replicability and portability of Smart agri-food solutions practical. This allows for sector-specific focus in a procurement or development process, while maintaining cross-domain consistency. At this respect NGSI and NGSI-LD (already standardized by ETSI) are information meta-models through which concrete, domain-specific data models have been expressed in a coherent way for the agri-food domain. The **FIWARE / GSMA Data Models** enriched with ADAPT Fragments and other extensions (for instance, to capture *Animal data*) are a concrete, practical realization of this approach.

Besides, for realising the IoF2020 vision of replicable and interoperable Digital Farming solutions, it is necessary to count with integrated Open Platforms that can solve major development challenges, while offering an architecture compliant with the MIMs. **FIWARE** is an open source platform for implementing Smart Solutions, including but not limited Digital Farming. FIWARE includes different components gravitating around the *Context Broker*, a “data broker” which offers standardized interfaces to publish, store, and consume (through NGSIv2 and NGSI-LD) data tagged with associated temporal and spatial metadata (context data). **365FarmNet** is a popular Farm Management Information System (FMIS) software that offers open and pluggable interfaces, enabling the creation of a multiple stakeholder marketplace of Digital Farming solutions.

Once an Open Platform is consolidated as the substrate of a Smart Farming solution, new needs might arise. For instance, how to monetize data or services. Different IoF2020 partners are offering open software components that can be instantiated to experiment with these innovative services.

The final aim of our work is to help use case stakeholders, product owners and developers to identify, configure and extend a set of common technology enablers, software components, open platforms and related architectures that guarantee the **creation of a sustainable ecosystem** of portable solutions for the Farm and Food sector. In the end, that will foster the

- Flourishment of a marketplace composed by different vertical solutions capable of interoperating and integrating into a broader system of farm management.

- Identification and development of IoT reusable components and reference configurations and compositions in the framework of a common architecture.

Last but not least, this work has reported on the enhancements that have been developed at the time of writing. They demonstrate that existing open platforms, with some additional work, can perfectly fulfill the needs of further interoperability and portability of Smart agri-food solutions with a view to adopting a Systems of Systems approach. However, more work has to be done during the last phase of the IoF2020 project (and afterwards in successor projects) in order to consolidate this ongoing effort. The envisioned work can be summarized as follows:

- Develop the first version of the **Web of Things** components, aligning the integration with existing Web of Things implementations, particularly Mozilla WebThings https://iot.mozilla.org/ and Eclipse IoT Node-WoT http://www.thingweb.io/.
- Extend the set of **Harmonized Data Models** by direct dialogue with all the Trials, especially the Meat, Dairy and Vegetables trials.
- Fully develop and consolidate the **data integration** offer and System of Systems approach through Node-Red, as a simplified tool that facilitates application development and integration.

11 Appendix

11.1 Guidelines to enhance/configure Context Broker

11.1.1 Introduction

The following guidelines are intended to disseminate how to configure and enhance the Context Broker with extra components to meet extra requirements. The final aim is to enable the development of Smart agri-food solutions using open APIs and reusable components, aiming at the portability and interoperability. Different configurations are explained (they can be combined):

- Basic configuration with Orion Context Broker and mongoDB.
- Extended configuration to secure a Context Broker endpoint.
- Extended configuration to connect IoT devices to a Context Broker through the MQTT protocol.
- Extended configuration to enable time series data and dashboards based on the Context Broker.
- Extended configuration to integrate external data sources.
- Extended configuration for subscribing to context changes.

Other extended configurations are possible and we encourage readers of this deliverable to read more about them at the FIWARE Tutorials site http://fiware-tutorials.rtfd.io/.

11.1.2 Basic configuration

Our first configuration and deployment will only make use of one FIWARE component - the **Orion Context Broker**. Usage of the Orion Context Broker is sufficient for an application to qualify as "Powered by FIWARE". Currently, the Orion Context Broker relies on open source *mongoDB* technology to keep persistence of the context data it holds. Therefore, the described architecture consists of two elements:

- The Orion Context Broker receives requests using NGSI
- The underlying Mongo-DB database: used by the Orion Context Broker to hold context data information such as data entities, subscriptions and registrations

Since all interactions between the two elements are initiated by HTTP requests, the entities can be containerized and run from exposed ports (see figure below).

Figure 56: Basic configuration of Orion Context Broker

In order to deploy the configuration described above, Docker can be used as follows:

- First pull the necessary Docker images from Docker Hub and create a network for our containers to connect to:

```
docker pull mongo:3.6

docker pull fiware/orion
```

```
docker network create fiware_default
```

- A Docker container running a mongoDB database can be started and connected to the network with the following command:

```
docker run -d --name=mongo-db --network=fiware_default \
  --expose=27017 mongo:3.6 --bind_ip_all --smallfiles
```

- The Orion Context Broker can be started and connected to the network with the following command:

```
docker run -d --name fiware-orion -h orion --network=fiware_de-
fault \
  -p 1026:1026  fiware/orion -dbhost mongo-db
```

You can check if the Orion Context Broker is running by making an HTTP request to the exposed port:

```
curl -X GET \
  'http://localhost:1026/version'
```

The response will look similar to the following:


```
{
    "orion": {
        "version": "2.2.0-next",
        "uptime": "11 d, 17 h, 41 m, 26 s",
        "git_hash": "e3e0e61e2e1214a8dbc2a41f53ac9f8d1ea51d78",
        "compile_time": "Mon Apr 15 07:28:43 UTC 2019",
        "compiled_by": "root",
        "compiled_in": "37f16abb5bb2",
        "release_date": "Mon Apr 15 07:28:43 UTC 2019",
        "doc": "https://fiware-orion.rtfd.io/"
    }
}
```


Figure 57: Orion Context Broker query version payload

At its heart, FIWARE is a system for managing context information, so we need to add context data into the system by creating a new entity (for example *AgriParcel*). Any entity must have *id* and *type* attributes; additional attributes are optional and will depend on the system being described. The figure below describes the request and payload to be used to create a new *AgriParcel* Entity. Once executed a new Entity will be stored by Orion Context Broker (in mongoDB).

```
curl -iX POST \
  'http://localhost:1026/v2/entities' \
  -H 'Content-Type: application/json' \
  -d '
{
    "id": "urn:ngsi-ld:AgriParcel:72d9fb43-53f8-4ec8-a33c-fa931360259a",
    "type": "AgriParcel",
    "location": {
       "type": "geo:json",
        "value": {
             "type": "Polygon",
             "coordinates": [ [100, 0], [101, 0], [101, 1],
                                [100, 1], [100, 0]]
        }
    },
    "area": {
        "value": 200
    },
    "description": {
        "value": "Spring wheat"
    },
    "category": {
        "value": "arable"
    },
    "seeAlso": {
```

```
        "value" :[
            "https://example.org/concept/agriparcel"
        ]
    },
    "belongsTo": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:AgriFarm:f67adcbc-4479-22bc-9de1-cb228de7a765"
    },
    "ownedBy": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:Person:fce9dcbc-4479-11e8-9de1-cb228de7a15c"
    },
    "hasAgriParcelParent": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:AgriParcel:1ea0f120-4474-11e8-9919-672036642081"
    },
    "hasAgriCrop": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:AgriCrop:36021150-4474-11e8-a721-af07c5fae7c8"
    },
    "cropStatus": {
        "value": "seeded"
    },
    "lastPlantedAt": {
        "type": "DateTime",
        "value": "2016-08-23T10:18:16Z"
    },
    "hasAgriSoil": {
        "type": "Relationship",
        "value": "urn:ngsi-ld:AgriSoil:429d1338-4474-11e8-b90a-d3e34ceb73df"
    },
    "hasDevices": {
        "type": "Relationship",
        "value": [
            "urn:ngsi-ld:Device:4a40aeba-4474-11e8-86bf-03d82e958ce6"
        ]
    }
}
```

Figure 58: Request and payload for creating a new Entity in Orion Context Broker

Once new Entities are created in the system, updates, queries and other operations can be issued to the Orion Context Broker through the NGSI APIs. Additional details can be found at the corresponding FIWARE Tutorials http://fiware-tutorials.rtfd.io/.

11.1.3 Extended configuration for securing access

This extended configuration uses the FIWARE **Wilma** PEP Proxy combined with **Keyrock** to secure access to endpoints exposed by the FIWARE Context Broker. Users (or other actors) must log-in (through Keyrock) and use a token to gain access to services. The figure below shows the architecture of this configuration:

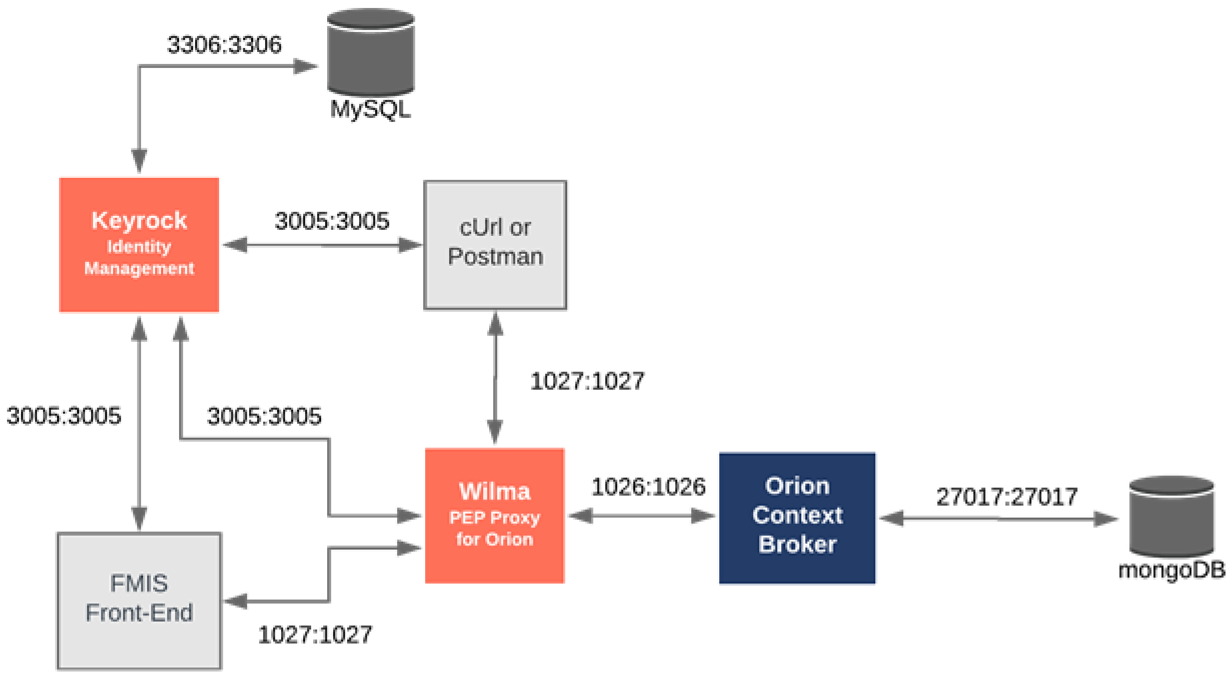


Figure 59: Extended Configuration with PEP Proxy to enable secure access to context data

This enhanced and extended configuration includes two new components: **Wilma** and **Keyrock**. FIWARE **Wilma** is a simple implementation of a PEP proxy designed to work with the FIWARE **Keyrock** Generic Enabler. FIWARE **Keyrock** offers a complement *Identity Management System* including:

– An OAuth2 authentication system for Applications and Users

– A site graphical frontend for Identity Management Administration

– An equivalent REST API for Identity Management via HTTP requests

Whenever a user or application (for instance an FMIS) tries to gain access to the resource behind the Wilma proxy (in this case the Orion Context Broker), Wilma will describe the user's attributes to **Keyrock** (through an OAuth2 token previously obtained), request a security decision, and enforce the decision. (Permit or Deny). There is minimal disruption of access for authorized users - the response received is the same as if they had accessed the secured service directly. Unauthorized users are simply returned a *401 - Unauthorized response*.

More details about how to deploy (using Docker) and use this configuration in practice and associated API operations can be found at https://fiware-tutorials.readthedocs.io/en/latest/pep-proxy/index.html.

11.1.4 Extended configuration for gathering data from IoT Devices (MQTT)

This configuration is intended to integrate data coming from IoT Devices into the Context Broker. The FIWARE IoT Agents act as a middleware between IoT devices and the Context Broker. They therefore need to be able to create or update context data entities with unique Ids resulting from a mapping between NGSI Entities and IoT devices. Once a service has been provisioned and an unknown device makes a measurement, the IoT Agent add this to the context using the supplied <device-id> (unless the device is recognized and can be mapped to a known Entity Id).

In this particular case, the IoT protocol used is **MQTT**, which is a publish-subscribe-based messaging protocol used in the internet of Things. It works on top of the TCP/IP protocol, and is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The goal is to provide a protocol, which is bandwidth-efficient and uses little battery power. MQTT is different in that publish-subscribe is event-driven and pushes messages to clients. It requires an additional central communication point (known as the *MQTT broker*), which it is in charge of dispatching all messages between the senders and the rightful receivers.

The main elements of our extended configuration (depicted by the figure below) are:

- The **FIWARE Orion Context Broker** receives requests using NGSI, coming from different clients, for instance, FMIS.
- The **FIWARE IoT Agent**:
  - receives southbound requests using NGSI and converts them to UltraLight 2.0 MQTT topics for the MQTT Broker
  - listens to the MQTT Broker on registered topics to send measurements northbound
- The **Mosquitto MQTT Broker** acts as a central communication point, passing MQTT topics between the IoT Agent and IoT devices as necessary.
- The underlying **mongoDB** database:
  - used by the Orion Context Broker to hold context data information such as data entities, subscriptions and registrations
  - used by the IoT Agent to hold device information such as device URLs and Keys

More details about how to deploy (using Docker) and use this configuration in practice and associated API operations can be found at https://fiware-tutorials.readthedocs.io/en/latest/iot-over-mqtt/index.html.

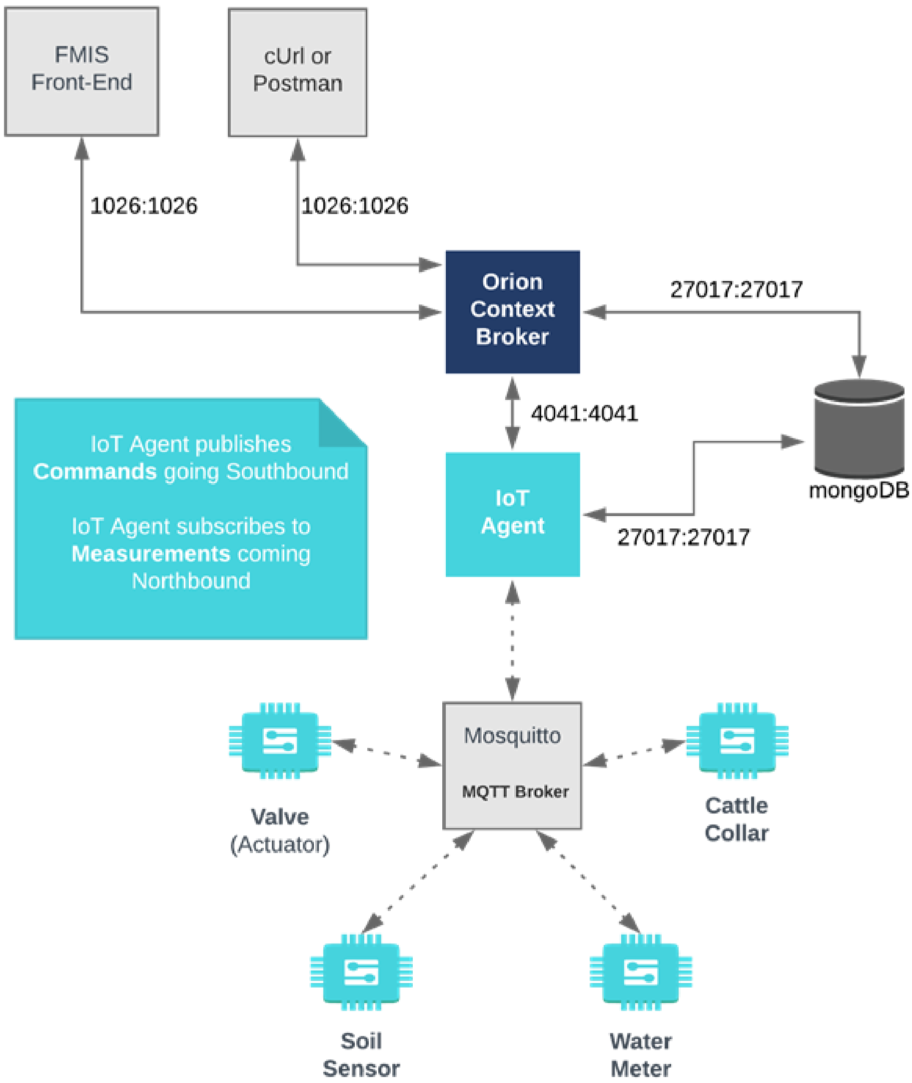


Figure 60: Extended Configuration to connect IoT devices (MQTT) to the Context Broker

11.1.5 Extended configuration to enable time series data

This configuration makes use of FIWARE **QuantumLeap** - a generic enabler used to persist context data into the *CrateDB* time-series database. CrateDB is a distributed NoSQL DB designed for use with IoT. It is capable of ingesting a large number of data points per second and can be queried in real-time. The database is designed for the execution of complex queries, such as geospatial and time series data. Retrieval of this historic context data allows for the creation of graphs and dashboards displaying trends over time.

This configuration (which architecture is depicted by the figure below) is based on the IoT sensors connected in the previous section and persists measurements from those sensors into the database. To retrieve time-based aggregations of such data, users can either use **QuantumLeap** query API or connect directly to the *CrateDB* HTTP endpoint.

Results are visualised on a graph or via the *Grafana* time series analytics tool. **Grafana** is an open source software for time series analytics tool. It integrates with a variety of time-series databases including CrateDB. It is available licensed under the *Apache License 2.0.*

The appropriate use of time series data analysis will depend on each use case and the reliability of the data measurements received. Time series data analysis can be used to answer questions such as:

- What was the maximum measurement of a device within a given time period?
- What was the average measurement of a device within a given time period?
- What was the sum of the measurements sent by a device within a given time period?

More details about how to deploy (using Docker) and use this configuration in practice and associated API operations can be found at https://fiware-tutorials.readthedocs.io/en/latest/time-series-data/index.html

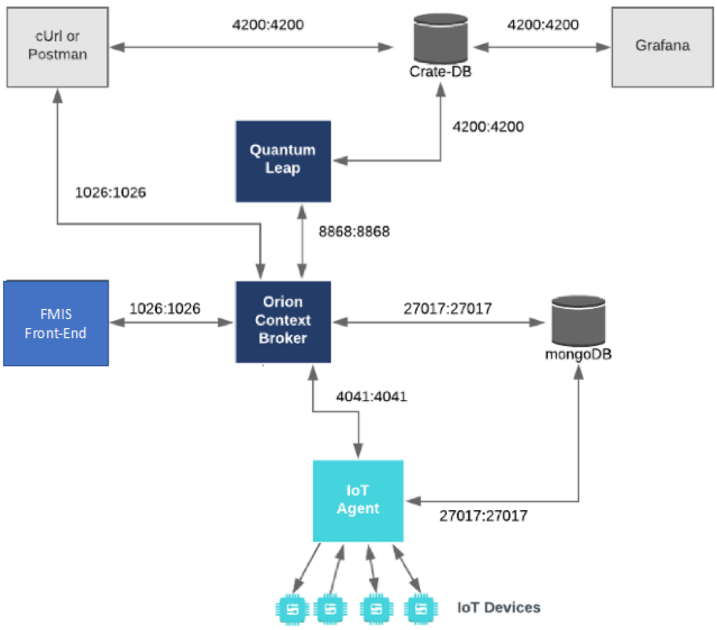


Figure 61: Extended Configuration to enable time series data with the Context Broker.

11.1.6 Extended configuration to integrate external data sources

Smart solutions are designed to react on the current state of the real-world. They are "aware" since they rely on dynamic data readings from external sources (such social media, IoT sensors, user inputs). The FIWARE platform makes the gathering and presentation of real-time context data transparent, since whenever an NGSI request is made to the Orion Context Broker it will always return the latest context by combining the data held within its database along with real-time data readings from any registered external data sources. In order to fulfil these requests, the Orion Context Broker must first be supplied with two types of information:

- The context data held within Orion itself (Entities that Orion "knows" about)
- Registered external context providers (external data sources) associated with existing entities (Entities that Orion can "find information" about)

The architecture of this extended configuration (depicted below) consists of three elements:

- The **Orion Context Broker** receives requests using NGSI
- The underlying **mongoDB** database:
  - used by the Orion Context Broker to hold context data information such as data entities, subscriptions and registrations
- The **Context Provider NGSI proxy**:
  - receives requests using NGSI
  - makes requests to publicly available external data sources using their own APIs in a proprietary format
  - returns context data back to the Orion Context Broker in NGSI format.

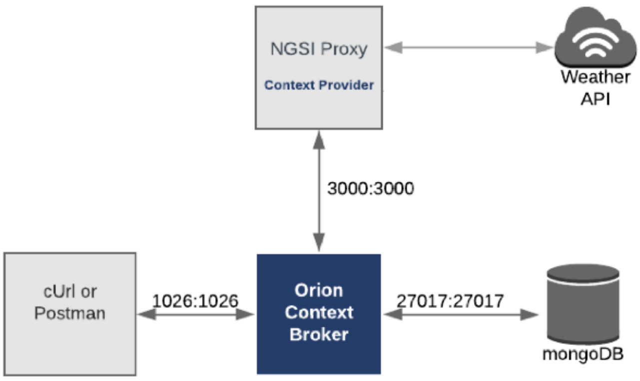


Figure 62: Extended Configuration to integrate external data sources with the Context Broker

The figure below shows how to register a Context Provider (NGSI Proxy) within the Orion Context Broker (using NGSIv2). In this case, the external data provider is supplying data corresponding to the temperature and relative humidity of NGSI Entities of type *AgriParcel*.

```
curl -iX POST \
  'http://localhost:1026/v2/registrations' \
  -H 'Content-Type: application/json' \
  -d '{
  "description": "Weather conditions provider",
  "dataProvided": {
    "entities": [
      {
        "type": "AgriParcel"
      }
    ],
    "attrs": [
      "temperature",
      "relativeHumidity"
    ]
  },
  "provider": {
```

```
    "http": {
      "url": "http://context-provider:3000/proxy/v1/random/weatherConditions"
    },
     "legacyForwarding": true
  }
}'
```

Figure 63: How to register an NGSI Context Provider that supplies data provided externally.

More details about how to deploy (using Docker) and use this configuration in practice and associated API operations can be found at https://fiware-tutorials.readthedocs.io/en/latest/context-providers/index.html

11.1.7 Extended configuration to enable context subscriptions

Within the FIWARE platform, an entity represents the state of a physical or conceptual object which exists in the real world. Every smart solution needs to know the current state of these objects at any given moment in time. The context of each of these entities is constantly changing. For a smart solution based on IoT sensor data, this issue is even more pressing as the system will constantly be reacting to changes in the real world. For example, within the agri-food domain, the context will change as sensors reports new values about soil or watering conditions.

The Orion Context Broker offers also an asynchronous notification mechanism - applications can subscribe to changes of context information so that they can be informed when something happens. This means the application does not need to continuously poll or repeat query requests. Use of the subscription mechanism will therefore reduce both the volume of requests and amount of data being passed between components within the system. This reduction in network traffic will improve the overall responsiveness.

The figure below shows how a subscription can be created (in NGSIv2), so that, when the soil temperature overpasses a certain threshold, a notification will be sent by the Context Broker. Such notification will be delivered through the HTTP POST protocol and using the NGSI JSON(-LD) representation format.

```
curl -iX POST \
  --url 'http://localhost:1026/v2/subscriptions' \
  --header 'Content-Type: application/json' \
  --data '{
  "description": "Notify high soil temperature",
  "subject": {
    "entities": [{"id": "urn:ngsi-ld:AgriParcelRecord:12345"}],
    "condition": {
      "attrs": ["soilTemperature"],
      "expression": {"q": "soilTemperature>20"}
    }
  },
  "notification": {
    "http": {
      "url": "http://fmis.example.org:3000/subscription/high-soil-temp"
    },
    "attrsFormat": "keyValues"
  }
}'
```

Figure 64: How to create a Context Subscription (NGSIv2).

aMore details about how to deploy (using Docker) and use this configuration in practice and associated API operations can be found at https://fiware-tutorials.readthedocs.io/en/latest/subscriptions/index.html

IOF 2020

INTERNET OF FOOD & FARM