D3.1 - GUIDELINES FOR USE CASE ANALYSIS & DESIGN

WP 3 - IOT

September 29th , 2017

Approach and reference architecture approach to be applied for the analysis, design and conception of IoT based solutions

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	2017-2021
Website	www.iof2020.eu

File Name	IoF2020-D3.1-UC-Analysis-Design-Guidelines-V11.docx
Date	September 29th , 2017
Version	11
Status	Final
Dissemination level	PU
Authors	R. Tomasi, F. Rizzo, D. Conzon (ISMB) H. Sundmaeker, G. Große Hovest, A. Vyas (ATB) C. Verdouw (DLO) J. Berg, F. Manoel (NXP)
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

In order to demonstrate the effectiveness of IoT solutions in a large spectrum of different agricultural domains and applications, IoF2020 has carefully selected 5 trials comprising 19 Use Cases (UCs), set in different regions of Europe. This is a key aspect to reflect the diversity of the agri-food domain, and perform evaluations in conditions, which are close to real scale and operational ones.

Building on the experience being generated on the field, the role of task T3.1 “Smart Agri-food Solution Reference Architecture and Interoperability end-point specification” is to establish a common architectural view for each of the UCs. This is an important preparatory step towards further activities in the project, which aim at the full realization of the IoT vision across the 19 IoF2020 UCs, ensuring that deployed components and solutions can prospectively inter-operate so to deliver added-value functionalities to various stakeholders – possibly maximizing re-use of common IoT enablers across different use cases and trials.

The main goal of this document is to provide an overview of the methodology employed by the IoF2020 consortium to elicit and specify the most important IoT architectural aspects of the UCs.

The methodology is centred on the definition of a minimal set of architectural views. Such views include: a domain model; a deployment view; an “IoT” Functional view; a “Business Process Hierarchy” valid for the Agri-food domain; a description of the Interoperability Endpoints; an Information model; a summary of gaps; a selection of assets identified for re-use; a Security, Privacy and Trust Analysis.

TABLE OF CONTENTS

1. INTRODUCTION

The IoF2020 “Internet of Food and Farm 2020” project aims at boosting the competitiveness of European agriculture on a global scale, by accelerating the adoption of the methodologies and products based on IoT technologies in the broad agriculture domain.

In order to demonstrate the effectiveness of IoT solutions in a large spectrum of different agricultural domains and applications, IoF2020 has carefully selected 5 trials comprising 19 Use Cases (UCs), set in different regions of Europe. This is a key aspect to reflect the diversity of the agri-food domain, and perform evaluations in conditions, which are close to real scale and operational ones.

From the organizational viewpoint, in order to ensure that all UCs can quickly achieve a working, operational status, since the early phases of the project, each UC is managed by a separate, dedicated team, working in proactive, highly autonomous fashion. This ensures the activation of UCs in relatively short time, allowing further iteration and enhancements over the duration of the project.

Building on the experience being generated on the field, the role of task T3.1 “Smart Agri-food Solution Reference Architecture and Interoperability end-point specification” is to establish a common architectural view, for each of the UCs. This is an important preparatory step towards further activities in the project, which aim at the full realization of the IoT vision across the 19 IoF2020 UCs, ensuring that deployed components and solutions can prospectively inter-operate so to deliver added-value functionalities to various stakeholders – possibly maximizing re-use of common IoT enablers across different UCs and trials. This can only be achieved by leveraging common interoperability end-points and data models and allowing secure and controlled exchange of information and capabilities across heterogeneous components.

The first step, to facilitate the identification and exploitation of common interoperability end-points, re-usable components and added-value functionalities, is to achieve a light-weight, yet harmonized technical description of the IoT architecture adopted in the 19 IoF2020 UCs. While building upon established best practices and formats for analysing and describing IoT architectures, the IoF2020 project has devised a dedicated methodology suitable to elicit the most important architectural aspects and facilitating further work, related to IoT Standardization, development, integration and deployment of Smart Agrifood open platforms, generation and coordination of synergies across UCs, development, integration and deployment of components usable across use-cases.

This document, entitled “D3.1 - Guidelines for Use Case Analysis & Design” has been developed by Task T3.1 “Smart Agri-food Solution Reference Architecture and Interoperability Endpoints Specification” – supported by the WP2 team.

Its main goal is to provide an overview of the methodology employed by the IoF2020 consortium to elicit and specify the most important IoT architectural aspects of the UCs. This document is associated to deliverable D3.2 “The IoF Use Case Architectures and overview of the related IoT systems” – which describes the concrete results, generated by the application of the methodologies described in D3.1. As the methodology has been progressively refined and improved during its application, no further revision of this document is foreseen.

Methodologies and guidelines described in this document are meant to structure the UCs’ IoT challenges in technological terms, to facilitate an assignment of the different structural elements to a high level, platform-based smart agri-food reference architecture, including the identification of necessary interoperability endpoints i.e. interfaces to IoT devices, IT systems, applications, data catalogues and any other relevant real time/historic data sources.

2. METHODOLOGY FOR IOF2020 USE CASE ARCHITECTURE ANALYSIS

Figure 1 depicts the overall process followed in IoF2020, and more specifically by participants of task T3.1 to elicit a common IoT architectural description within the IoF2020 project.

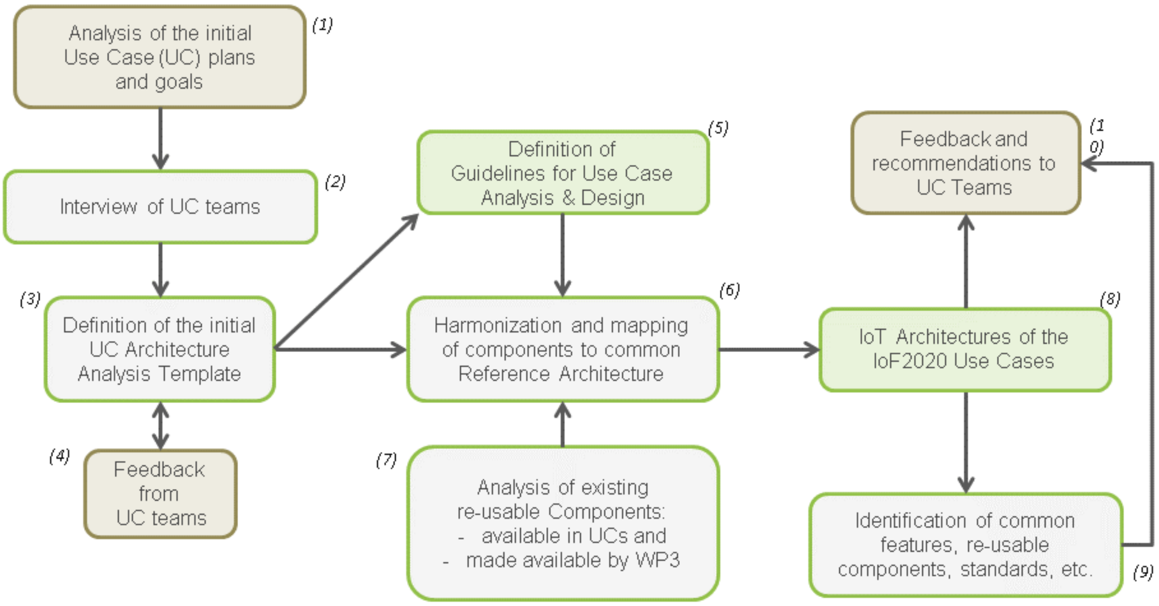
Analysis of the initial Use Case (UC) plans and goals
(1)
Interview of UC teams
(2)
(3)
Definition of the initial UC Architecture Analysis Template
(4)
Feedback from UC teams
Definition of Guidelines for Use Case Analysis & Design
(5)
(6)
Harmonization and mapping of components to common Reference Architecture
(7)
Analysis of existing re-usable Components:
- available in UCs and
- made available by WP3
Feedback and recommendations to UC Teams
(10)
(8)
IoT Architectures of the IoF2020 Use Cases
Identification of common features, re-usable components, standards, etc.
(9)

Figure 1 - Overall analysis and design methodology

Analysis work has started from the high-level description of UCs included in the initial plans of the IoF2020 project and further refined by UC teams in the first days of the project (the step 1 of Figure 1).

Based on the description of such initial plan, a team of system architects and analysts within the Task T3.1 team has elaborated a first set of open questions, as described in section 5.1. These questions have been followed in a set of interviews to key partners of each UC, which took place at the IoF2020 Kick-off Meeting (KoM) (2).

Based on the first assessment of UC architectures, collected through initial interviews, a more detailed, structured template, namely the “Use Case Analysis Template”, has been produced (3) and iterated between the IoT experts in task T3.1 and the UC teams (4), trying to find a balance between the expressiveness of the analysis and the ease of interpretation by experts in the Agri-food domain with no strong background in IoT, and, more in general in System Engineering. Since the main actors filling such template are not necessarily experts in architectural formalisms or IoT standardization, it has been structured to allow UC experts to formalize all the important technical aspects in simple format, ensuring that the T3.1 analysis team has all the necessary information to proceed with further harmonization / mapping of all components to a common IoT Reference Architecture, as well as with the identification of gaps (i.e. situations where the use case team requires assistance to find suitable IoT solutions), which cannot be easily procured from the IoT market and needs to be developed and / or improved to meet the needs of the UC. An excerpt from the template has been reported in section 5.2.

The proposed template has been then filled by the various UC teams, while refining the short-term work plan to be followed to reach the first deployment.

Thanks to the lessons learned during the interviews and the feedback by UC teams, the Task T3.1 team has been able to organize the subsequent steps (5,6) leading to the selection of the final methodology for IoT architecture analysis (described in Section 3) and its implementation (whose results are described in deliverable D3.2).

In this stage, the team has adopted a common template also for the style and the formats of the different views, collected in a “Companion Template for Use Case Analysis” described in section 5.2, which has supported the team in the finalization and iterative improvements of the results.

From the organizational viewpoint, the work has been collaboratively performed by a team of analysts spread across the organizations most active in task T3.1. More specifically, a dedicated team member, namely the “UC Analyst” has been selected in the T3.1 team and associated to each UC.

The responsibilities of the UC Analysts have been:

- To develop the different “views” on the IoT architecture of the UC
- To report to the T3.1 the status of the analysis and facilitate the identification of synergies and common entities (components, standards, data models, etc.) across the UCs
- To interact with the UC teams to obtain feedbacks and improve the analysis
- To review the security analysis performed for the UC

In order to facilitate the work, whereas possible, the UC analyst have been selected among the organizations directly involved in the realization of UCs. As not all UCs had a corresponding participant in task T3.1, the remaining UCs have been divided and allocated based on preferences.

While this “bottom-up” or “IoT pull” analysis process was being developed, a parallel thread of activities has been developed, to prepare the way for possible “IoT push” actions. Specifically, IoT solutions that can be potentially made available by partners to UCs have been collected and taxonimized. This activity has been performed while keeping in mind the set of “gaps” and “needs” identified for each UC.

2.1. ARCHITECTURE DESCRIPTION FORMALISM SELECTION

In the last decades, scientific and technical communities have spent major effort in trying to define unified guidelines and formalisms suitable to properly describe and document the key architectural aspects of software-intensive, complex system, which is sufficiently generic and descriptive to cover the needs of all application domains and all purposes. Such common architectural approaches are especially needed for technologies and projects, which work in cross-sectorial and “horizontal” fashion, as in the case of IoT applications – also to support technical communication and collaboration among teams with different goals and technical backgrounds.

While a single “one-fits-all” methodology suitable to cope with the description of complex, cross-domain systems has not yet emerged, a reasonably successful best practice exists, namely the multi-view approach described in the ISO/IEC/IEEE 42010 international standard[i] for “Systems and software engineering - Architecture description”.

Following such approach, IoF2020 has adopted the practices and suggestions indicated by the AIOTI WG03[ii], therefore taking the decision to map the architecture of each UC towards a common High-Level Architecture (HLA) model, as described in Section 3.

3. GUIDELINES FOR IOF2020 USE CASE ARCHITECTURE ANALYSIS

This section shortly describes the guidelines implemented in the IoF2020 project for UC Architecture analysis.

3.1. GENERAL GUIDELINES

In IoF2020, each UC must be specified by defining and analysing a minimal set of architectural views. Such views include: a domain model; a deployment view; an “IoT” Functional view; a “Business Process Hierarchy” valid for the agri-food domain; a description of the Interoperability Endpoints; an Information model; a summary of gaps; a selection of assets identified for re-use; a Security, Privacy and Trust Analysis.

As a general guideline, all provided views shall be built according to the following general principles:

- **Views shall be simple**: all descriptions shall be provided as much as possible in simple, pragmatic, unambiguous way.
- **Views shall be concise**: redundant or verbose information make that architectural views too long to read or too complex shall be avoided.
- **Views shall be consistent**: the same naming conventions shall be used for the same entities, especially across different views.
- **State-of-the-art shall be referenced, not replicated in descriptions**: in order to keep architectural views compact, all information from existing standards or products shall be clearly referenced (if possible with web links).

At the same time, IoF2020 embraces a demand-driven methodology in which end-users from agri-food are actively driving the entire development process, aiming at cross-fertilisation, co-creation and co-ownership of results. All innovative IoT technologies initially selected in the use cases have a value-proposition for end-users. Nevertheless, their development was possibly based on a large set of market and technical assumptions that have not been tested in a systematic way in their operational environment. Therefore, IoF2020 will apply an innovative approach, the lean multi-actor approach to test these assumptions in real bottlenecks of end-users in their operational environment and with value chain stakeholders, using so called MVPs (Minimum Viable Products). Feedback is translated into technical improvements that better meet end-user needs and better fit into the production environment and the value chain.

IoF2020 is planning several cycles, where each cycle is realising an MVP. By passing through this cycle, the technology is altered with a new set of features and a new minimum viable product (MVP) compared to the beginning of the process. Therefore, in the first iteration, the use case architecture analysis is considering the envisaged requirements and features of the first MVPs that shall be available in early 2018 and serve as baseline for subsequent refinements and changes. This was also one reason for having a rather structured and systematic architecture analysis, even offering the potential for harmonising and reusing architectural elements.

A detailed explanation of each view and its rationale in the IoF2020 project is provided in the following.

3.2. DOMAIN MODEL

The domain model provides a high-level view of the main concepts and relationship in the domain of interest for the UC being analysed, describing its «sphere of knowledge, influence or activity[iii]». Being IoF2020 a cross-domain project, it should capture entities that are relevant either for the IoT domain and the agri food domain.

Whereas possible, entities described in such domain should be easily linked to:

- Well-established IoT architectural entities and patterns, such as the ones described in the IoT ARMs[iv] adopted by AIOTI's WG03
- Key vocabularies and concepts established in well-known ontologies for agriculture and farming[v], as well as well-known data dictionaries such as e.g. the ISOBUS Data Dictionary[vi]

The purpose of this view within IoF2020 is to summarize in a single picture the key functional aspects of the UCs, answering questions such as:

- Who are the key actors of the UC?
- What are they main systems and physical entities (e.g. animals, goods, equipment) already existing on the field?
- What are the main IoT systems that we plan to deploy in this UC?
- How will these entities (actors, existing systems, physical entities, IoT systems, etc.) interact with each other in the UC? For what purpose?

For the domain view, the chosen formalism has been a standard UML class diagram format, whose basic elements (i.e. entities and relationships) are described in Figure 2.

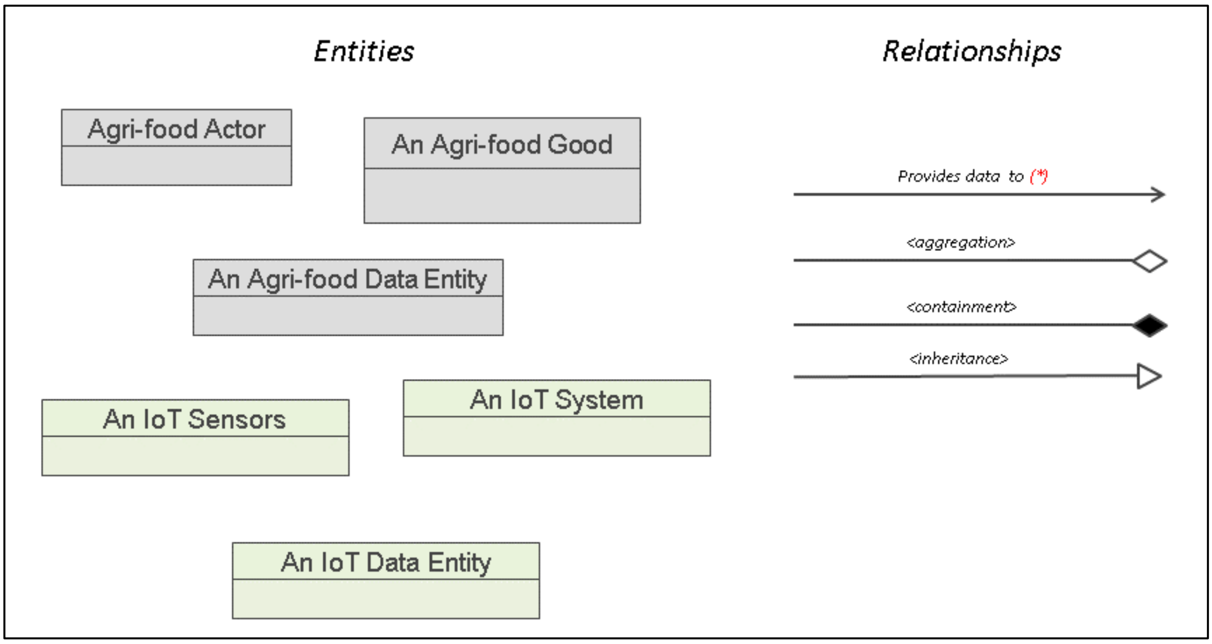

Figure 2 - Domain Model Elements

Within this diagram, all the main elements (actors, physical entities, virtual entities, etc.) are represented as classes or objects. In order to better focus the contributions of the project and the aspects where IoT elements are integrated with existing agri-food entities, pre-existing "field" elements are colorized in grey, while new IoT components and entities introduced by IoF2020 are colorized in green**.**

Following the UML class diagram conventions, arrows connecting classes are used to specify relationships among elements. In the project, mostly "specialized" relationship* are defined, but standard relationships (aggregation, containment, inheritance, etc.) are also possible.

Some UCs are involving features with a complex relation to the specific agri-food domain. A comprehensive description of all those aspects might require several inter-linked domain models. However, for the sake of readability and to facilitate cooperation among UCs, the targeted description for the domain model has been limited to a single class diagram accompanied by a concise description. This focus shall rather initiate the work about communalities than complicate the understanding, due to differences.

Domain model Example

An exemplary Domain Model description is reported in the following Figure 3.

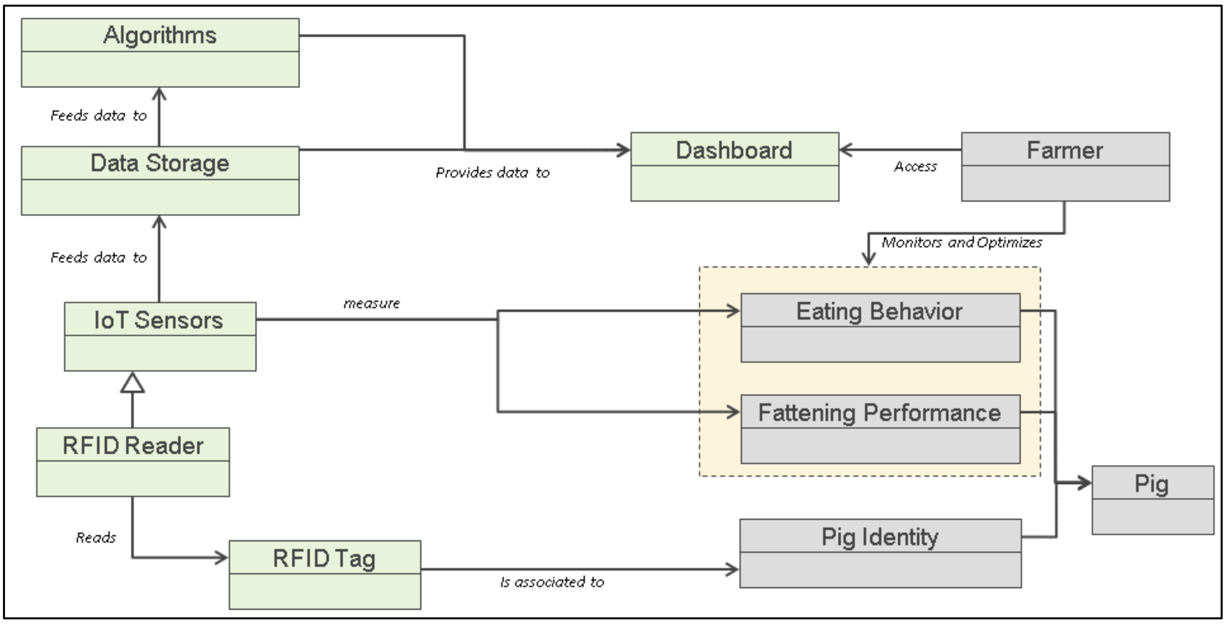

Figure 3 - Domain Model Example

In this UC, a Farmer is interested in monitoring and optimizing the Eating Behaviour and Fattening Performance of individual Pigs.

IoT sensors, deployed in the farm, are available to measure such parameters of interest. A special IoT sensor, namely the Radio Frequency Identification (RFID) Reader, is available to track individual pigs in specific areas where other sensors are active. This combination is exploited to associate measures with the Identity of each Pig. This is possible because each Pig is associated with a unique RFID Tag physically attached to its neck, which is in turn uniquely associated with the Pig Identity.

Data monitored by IoT sensors is locally stored to a Data Storage System, which feed dedicated algorithm suitable to extract eating behaviour and fattening performance figures – which are made available to the Farmer through a dedicated web-based Dashboard.

3.3. DEPLOYMENT VIEW

The deployment view models the physical deployment of the UC, i.e. where and how different systems (software, hardware) are deployed on the field. It is generally implemented as a standard UML deployment diagram[vii].

The purpose of this view within IoF2020 is to give a concrete overview of which systems are deployed, where they are placed and how they are integrated, answering to questions such as:

- What are the concrete components (software, hardware) deployed in this UC?
- Where are such components installed? On which hardware?
- What is the needed network infrastructure? How it is used by different components?

The main elements that can be used in the deployment view are described in Figure 4.

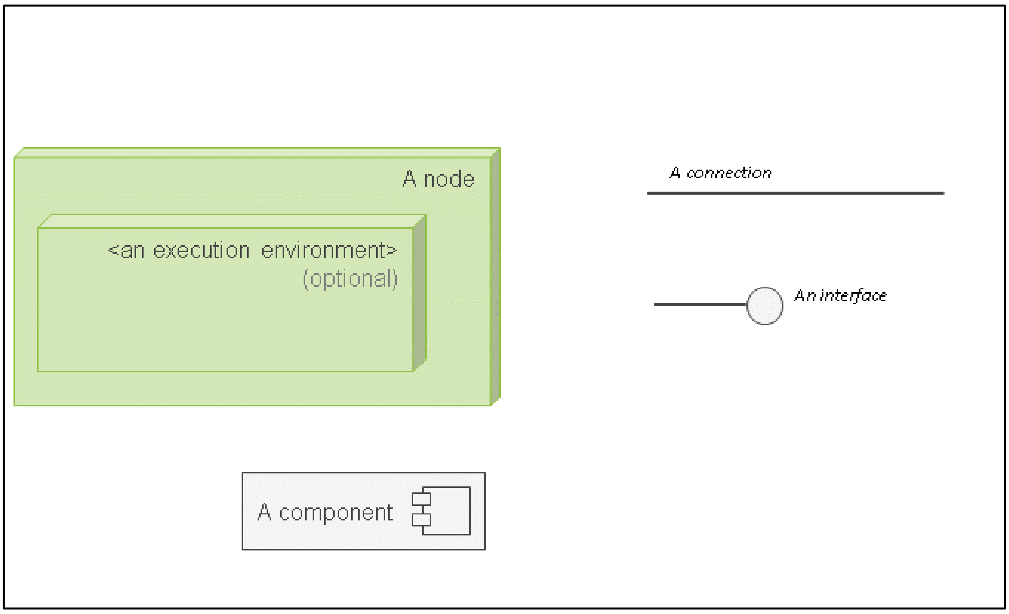

Figure 4 - Deployment View Elements

Within this diagram, the key element is the “node” i.e. a concrete computing environment (i.e. a PC, a gateway, an embedded system) hosting one or more components.

Execution environments, e.g. Docker[viii], Java Runtimes, application servers, etc. can be specified inside nodes, but in order to keep the picture simple, it is advised to specify such environments in the picture, only in cases where more than one execution environment is used within the same node, or where the fact that a specific execution environment is used has important consequences, e.g. for re-use of components.

Components i.e. any software or hardware building block are represented inside nodes and are inter-connected through lines representing specific interfaces, communication protocols and/or network technologies used for the connection.

In order to easily display “services” offered by components accessible by multiple parties, it is also possible to describe “open-handed” interfaces.

In IoF2020, the foreseen description to implement the deployment view is a single deployment diagram accompanied by a short, concise description. Since the re-use of components is a key aspect for the IoF2020 project, the deployment view can be accompanied by a table, summarizing some additional details for components included in the picture.

Deployment View Example

An exemplary Deployment View description is reported in the following Figure 5 and Table 1.

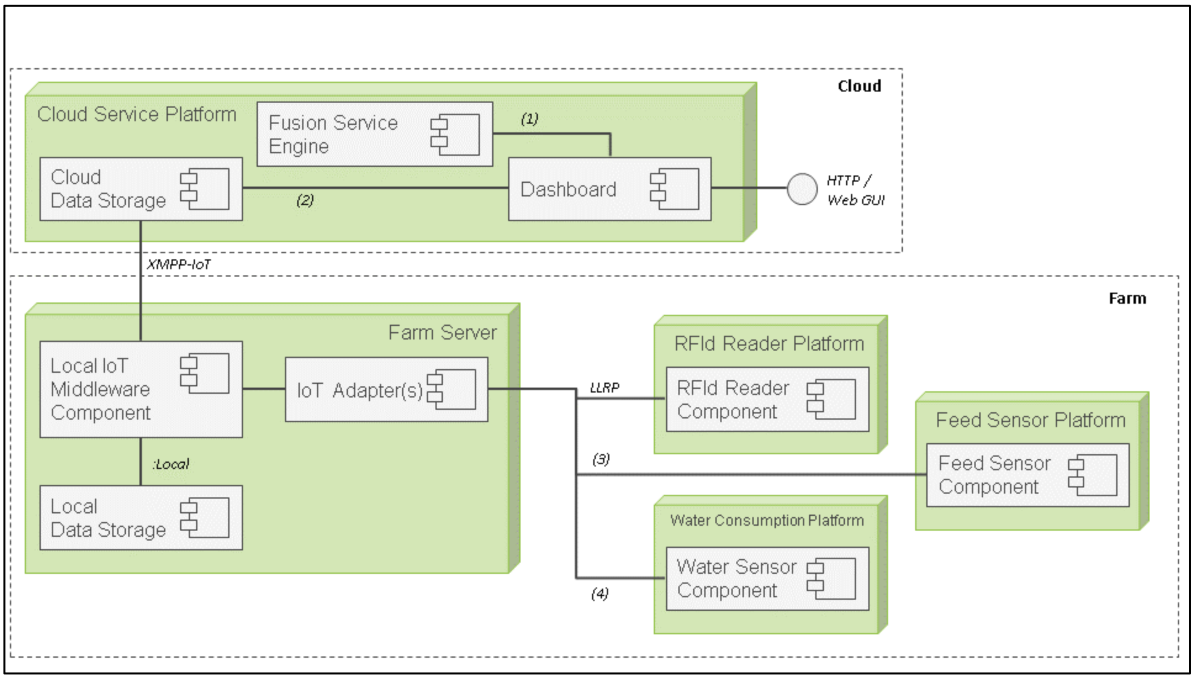

Figure 5 - Deployment View Example (Chart)

Table 1 - Deployment View Example (Table)

Name	***Description***	***Supplier (brand/model)***	***Number of units***
RFID Reader	RFID Reader monitoring presence of Pig in the trough	FEIG/DTE , model XYZ	8
…	…	…	…

Components in this UC are deployed either locally (i.e. in the Farm) or remotely (i.e. in the Cloud or in a self-hosted cloud server hosted by project partner XYZ).

In the Farm, three different physical, dedicated sensor platforms are deployed, namely the RFID Reader platform, the Water Consumption Platform and the Feed Sensor Platform. Platform correspond to a dedicated, stand-alone PC installed in a protected location in the farm. The nodes implementing these three platforms are dedicated PCs, and cannot host other components except for their original software. They are all connected to the local farm Local Area Network (LAN), which is a traditional ethernet-based local network, which is specifically used to inter-connect these nodes to the Farm Server. This is done by means of specific over-IP protocols including Low Level Reader Protocol (LLRP). Protocols (3) and (4) depicted in the diagram have not yet specified, but they will likely be implemented through protocol XYZ and YZW.

The Farm server is a general-purpose ruggedized x86-64 PC running Linux, which host three dedicated "IoT Adapter" components, one "Local IoT Middleware Component" and one "Local Data Storage" built upon a standard MongoDB installation.

The Farm Server is connected though the Internet to a global VPN, which allows secure communications towards a private Cloud Service platforms, hosted at XYZ premises. The cloud platform runs: a "Cloud Data Storage" service, receiving data via XMPP-IoT from Farm Servers; a Dashboard, accessible via HTTPS; a Fusion Engine Service running Algorithms. Interfaces (1) and (2) are built upon a local messaging protocol i.e. MQTT.

3.4. IOT FUNCTIONAL VIEW

The IoT functional view classifies the role of each component from the IoT point of view. It serves the purpose of placing each component of the UC within a categorization suitable to understand of what is the most suitable provider of infrastructure or technology suitable to offer such component.

Within the IoF2020 project, in order to better align with on-going IoT trends and standardization efforts, as well as following the recommendation by AIOTI WG03, this is done by depicting all the main functionalities within the ITU-T Y.2060 IoT Reference Model[ix].

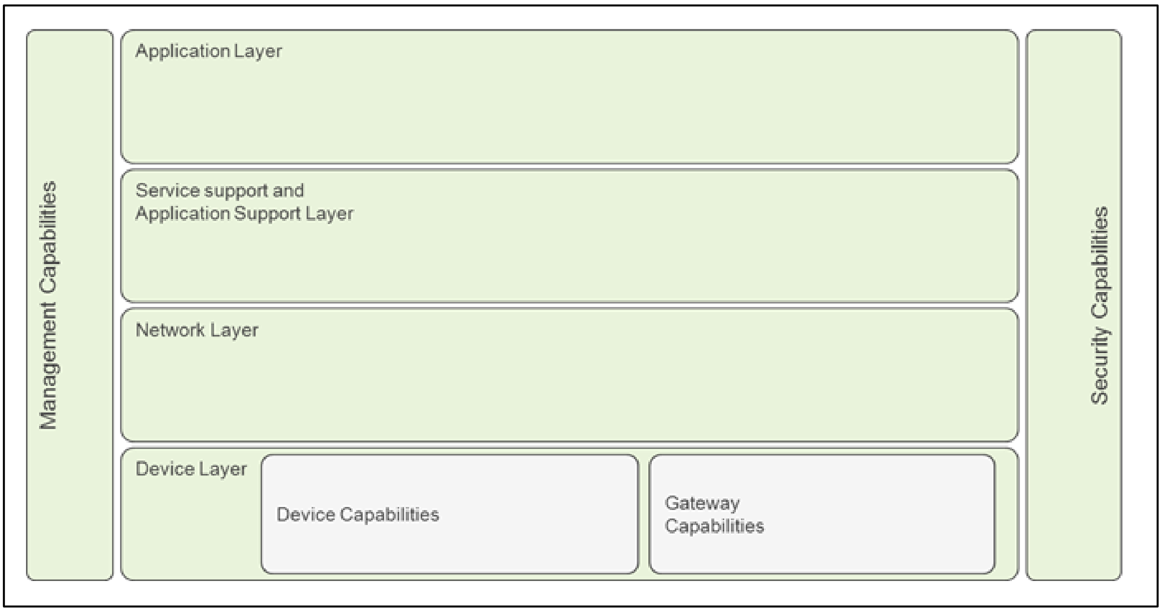

Figure 6 - The ITY-Y Y.2060 Reference Model

The purpose of this view within IoF2020 is to give a concrete overview of how adopted components map to common IoT functionalities, answering to questions such as:

- To which IoT layer belong the features of component XYZ?

The foreseen description for each UC is a picture, accompanied by a short description.

IoT Functional View Example

An example of mapping towards the IoT Functional View are described in Figure 7.

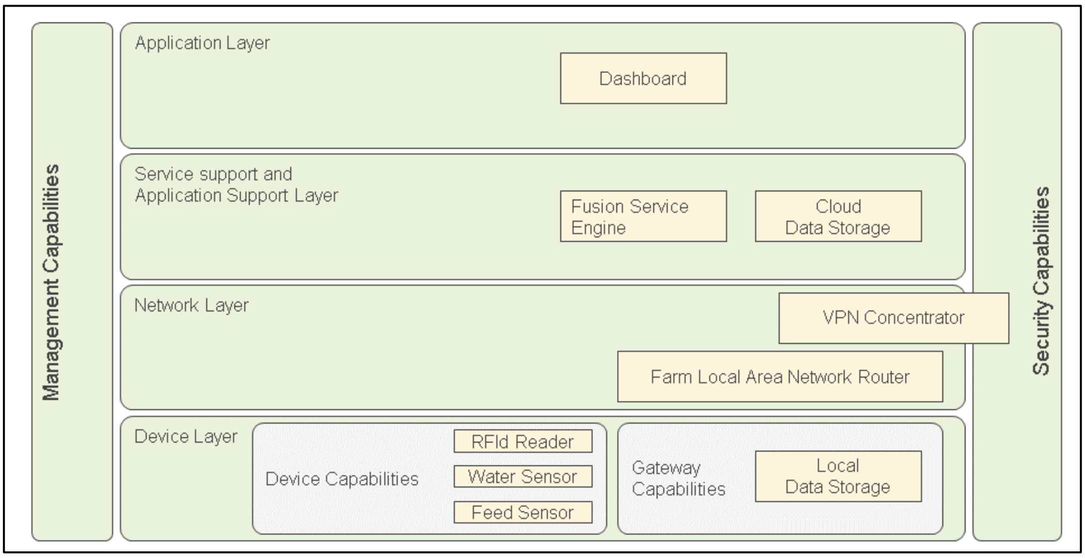

Figure 7 - IoT Functional View elements

3.5. BUSINESS PROCESS HIERARCHY VIEW

The IoT Functional View, mostly targets the needs of IoT-related stakeholders, including e.g. technology providers, infrastructure providers, integrators, etc. Being IoF2020 a cross-domain project, it has been decided to include an additional view helping players in the agri-food domain in correctly place the components and functionalities provided by UCs in a Business Process classification they are more familiar with. At this purpose, the Process Hierarchy View provides an overview of the business processes and their interrelations.

The business processes tackled by the UC are layered according to their position in the production control hierarchy, ranging from operational control of physical objects to enterprise management level. The levels are based on the ISA-95 reference model[x] for the industrial domain, which has demonstrated to be also very valuable for farming and food production systems.

Figure 8 shows that the Process Hierarchy View comprises four layers:

- The *Management Information Layer* identifies the processes that are related to the control of the entire enterprise (e.g. a farm). These processes have the longest time horizon (months, weeks, days) and focus on control on an aggregate level.
- The *Operations Execution Layer* identifies the processes that are related to the definition, control and performance of tasks. These processes have an intermediate time horizon (days, hours, minutes).
- The *Production Control Layer* groups the processes that are directly related to the execution of tasks by equipment and humans. These processes have the shortest time horizon (minutes, seconds, milliseconds).
- In the *Physical Object Layer* the relation to objects in the physical world is depicted. This is the actual real-time layer. Examples of these objects are fields, stables, animals, plants, farm equipment, processing facilities, containers, boxes, trucks, but also humans like employees or consumers. IoT devices including sensors and actuators can be attached to these physical objects.

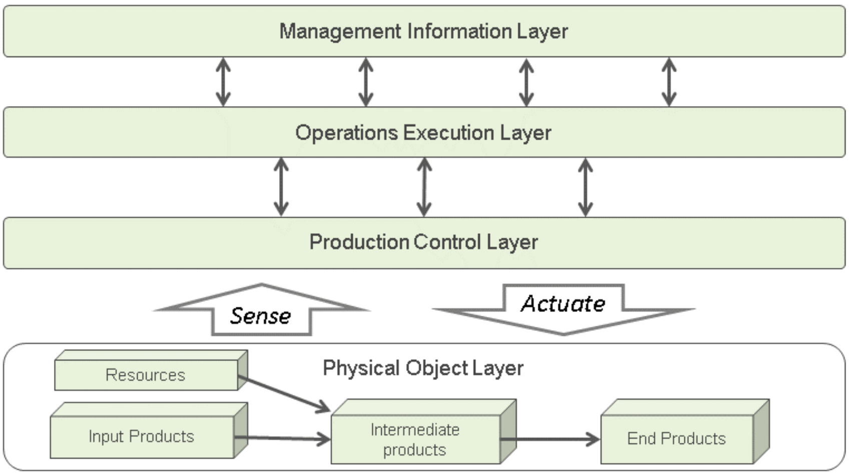

Figure 8 - Layers of the Process Hierarchy view

Business Process Hierarchy View Example

Figure 9 provides a generalized example of a Process Hierarchy Model.

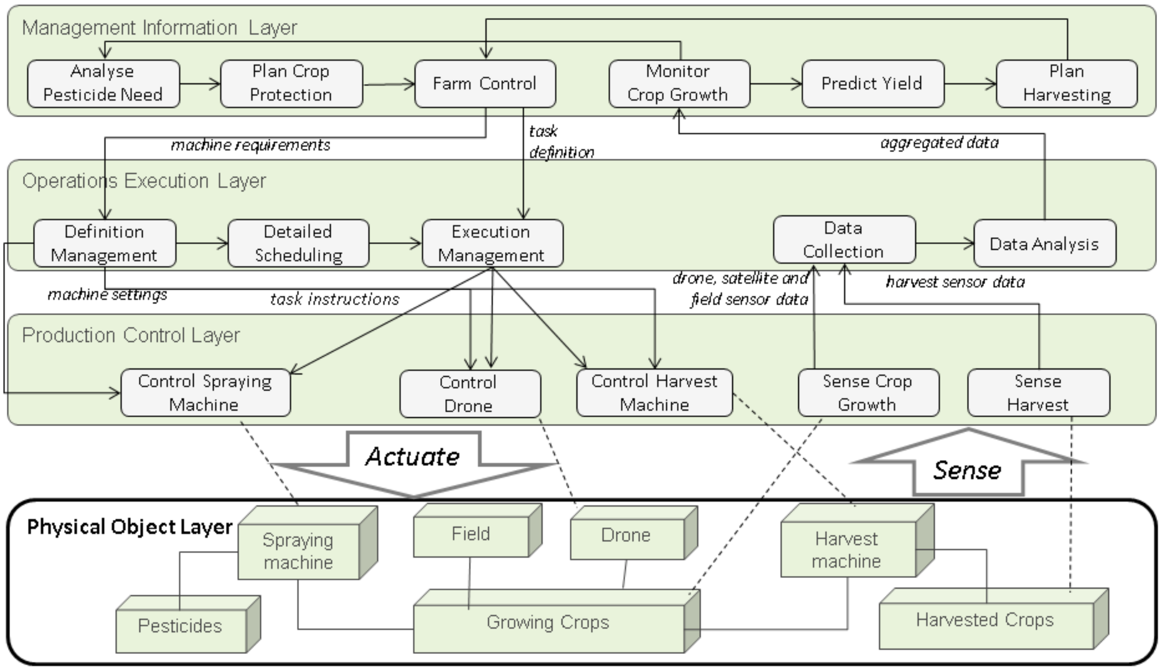

Figure 9 - Generalized example of a Process Hierarchy Model

3.6. INTEROPERABILITY ENDPOINTS

This Interoperability Endpoints View summarizes the main end-points, which can be exploited to integrate available systems to other systems. Its main purpose is to help identifying the most suitable entry points to access available legacy and IoT systems, deployed in each UC, referencing the standards and protocols, which must be implemented to perform such integration.

While this is not a "standard" view (such information is typically spread across the information, communication and deployment views), it has been adopted to facilitate the identification of technical synergies.

An example of the list of Interoperability End-points as exemplified in Table 2.

Table 2 - Interoperability Endpoints example

Interface name	Exposed by	Protocol	Notes
RFID Reader Interface	RFID Reader	LLRP (over IP, local)	Global EPC Standard
…	…	…	…

If the interface descriptions (or links) are not self-explanatory, text can be added below the table to clarity important aspects.

3.7. INFORMATION MODEL

The Information View[xi] models all the data entities across the use case architectures. This includes e.g. data models of databases used in the use cases, specifications of raw data collected from the fields, standard identification schemas, data entities in communication protocols, etc. It is generally implemented as a standard UML class diagram.

The purpose of this view within IoF2020 is to summarize in a single picture the key data-related aspects of the UCs, answering to questions such as:

- Which data entities appear in this UC? Which Systems handle such data entities
- What is their format?
- Can it be mapped to standard ontologies or taxonomies?
- How does data entities in this use case relate to each other?
- How is data transcoded, converted, mapped, elaborated within UC?

The main elements that can be used in the Information Model are described in Figure 10.

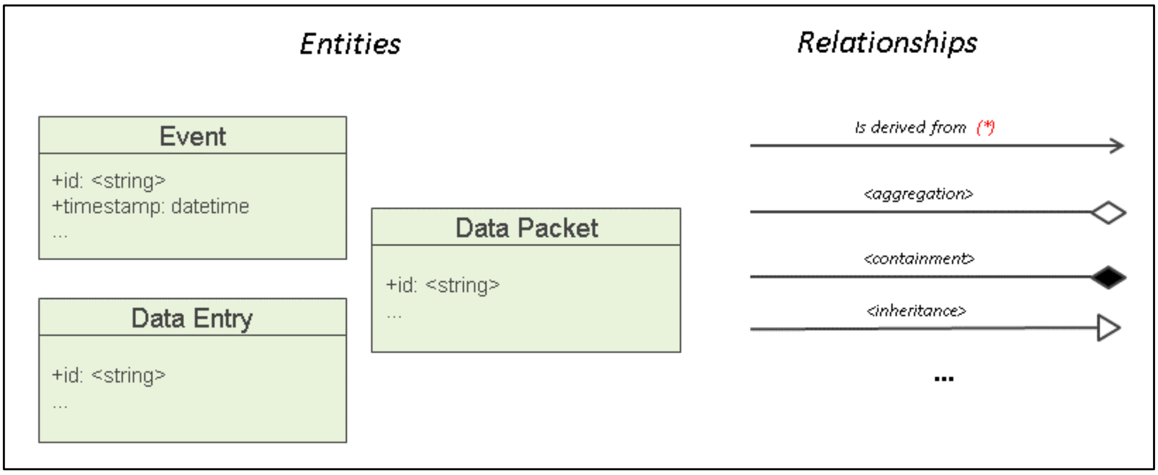

Figure 10 - Information View Elements

Within this diagram, all data elements (ids, raw data, time series, etc.) are just represented as objects/classes. UML arrows connecting classes are used to specify relationships among elements. We mostly used “specialized” relationship* as in the example on the left, but standard relationships are also possible.

In IoF2020, the Information Model is implemented for each UC with a single data model, represented in a single picture, optionally accompanied by a descriptive table providing details about the most relevant data entity.

Information Model example

An example of possible Information Model is described in the following Figure 11 and Table 3.

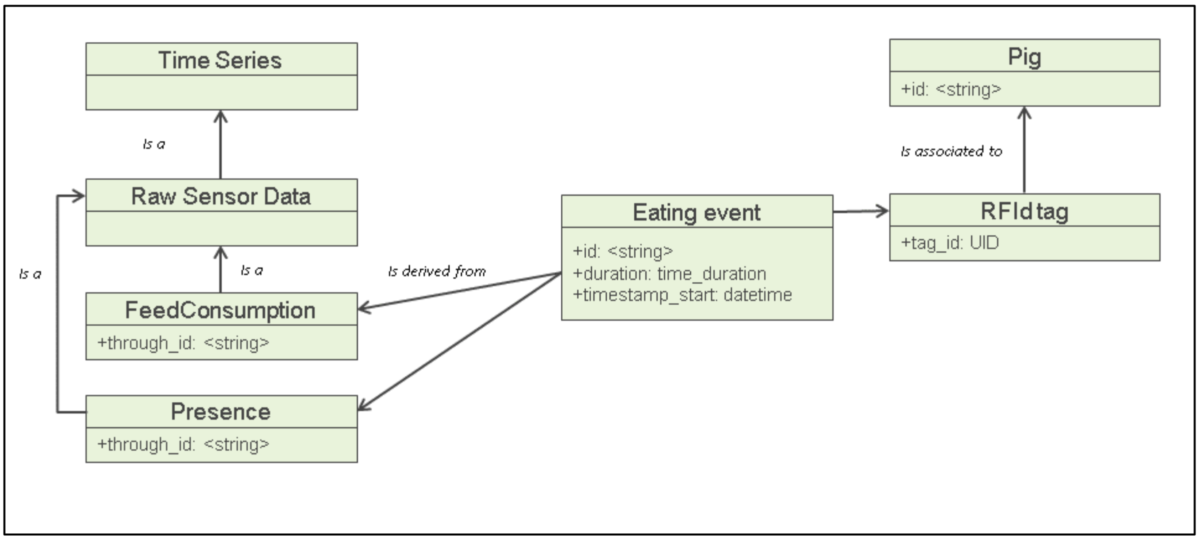

Figure 11 - Information View Example (Chart)

Table 3 - Information View Example (Table)

Data ID	Measurement Technique	Physical Entity	Frequency of data collection	Associated data model and format
…	…	…	….	

3.8. SUMMARY OF GAPS

As IoF2020 follows an iterative approach, it is possible that parts of the UCs specifications may be defined only at high-level or unclear. To facilitate the collection of recommendations, a short bullet-list can be specified for each UC, describing in free format:

- Aspects that need further specification by the UCs (i.e. decisions still to be taken)
- Aspects for whom the UC team is explicitly asking for support by IoT experts to identify suitable solutions or develop features that do not yet exist.
- Aspects of the UC that hamper the realization of the IoT vision e.g. components, which are non-interoperable or closed by design

3.9. ASSETS IDENTIFIED FOR RE-USE

A number of components adopted in specific UCs, may have potential for re-use or integration in other UCs. To facilitate the identification of such components, a table shall be prepared for each UC, following the template described in Table 4.

Table 4 - Assets identified for re-use

Component name	Short Description and role in the Use Case	Functional role	License
Name + web link	A concise description outlining the key feature of the component and its role in the UC	Mapping towards the ITU-T Y.2060 IoT Reference Model (e.g. IoT device, network, etc.) Mapping towards the Agri-food Functional Model (e.g. Farm Management, PLC, etc…)	Add link to license conditions unless they are standard (e.g. well-known open-source licenses)
…	…	…	….

3.10. SECURITY, PRIVACY AND TRUST ANALYSIS

For the Security, Privacy and Trust analysis of all 19 UCs, a STRIDE analysis was realised for each individual use case. NXP provided sufficient introduction and training sessions for all trial participants to conduct a STRIDE analysis by themselves, while being supported by the WP3 teams. NXP collected the findings and started to create a security, privacy and trust scorecard for the entire project IoF2020. The goal is to provide UC owners and participating partners with technology insights throughout the 2nd, 3rd and 4th year of the project to improve their stand on security, privacy and trust. At the end of the project, NXP will once again ask partners to conduct the same STRIDE analysis to collect findings on security improvements being made in the course of the project.

The STRIDE methodology is a threat classification model developed by Microsoft for thinking about computer security threats. It provides a mnemonic for security threats in six categories.

- **Spoofing identity**. An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
- **Tampering with data.** Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet.
- **Repudiation.** Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise — for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Nonrepudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package.
- **Information disclosure.** Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it — for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.

- **Denial of service.** Denial of service (DoS) attacks deny service to valid users — for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability.
- **Elevation of privilege**. In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defences and become part of the trusted system itself, a dangerous situation indeed.

The STRIDE was initially created as part of the process of threat modelling. STRIDE is a model of threats, used to help reason and find threats to a system. It is used in conjunction with a model of the target system that can be constructed in parallel. This includes a full breakdown of processes, data stores, data flows and trust boundaries.

A tutorial was prepared that was used for the training session provided to the use cases. To document the STRIDE analysis, a document template and a presentation template were provided accordingly.

4. CONCLUSIONS

This document has described the overall methodology and guidelines for Use Case Analysis and Design, developed within the IoF2020 project.

Following the overall methodology of the IoF2020 project, the proposed methodology has been necessarily iterative. The following steps have been planned for the next 9 months of the project:

- Continuous update and refinement of the analysis currently described in D3.2, leveraging feedback and interaction with the various UC teams.
- Finalization and structuring of the “IoT offering” i.e. components, platforms, standards that can benefit UCs and that can help creating synergies and added-value IoT services across different UCs and also across domains.
- Identifications of the common architectural aspects and the interoperability end-points that can be exploited to foster interoperability and collaboration with initiatives in other IoT domains e.g. other Large Scale pilots.

5. APPENDIX AND REFERENCES

5.1. APPENDIX A - INITIAL INTERVIEW TEMPLATE

The questions below have been used to guide the initial collection of inputs for architectural inputs during the early phases of the project, starting at the KOM.

High-priority questions to be solved in the early days of the project

1. *What is the core idea of the Use Case (in a sentence)?*
2. *What are the main business processes targeted at by the use case?*
3. *Which are the main actors using the envisaged system for which purpose?*
4. *What are the objects of the business processes (e.g. land parcels, stabled, machinery, crops, animals, containers, returnable trade items, etc.)?*
5. *What are the main functionalities/ services provided to the end-users?*
6. *Which are the main IoT technology related components to be used (e.g. sensors, actuators, autoID devices, wireless networks, gateways, FIWARE generic enablers, servers, etc.)?*
7. *Which components that you are reusing need to be enhanced/extended and could the resulting solution also add-value to other's future deployments?*
8. *Will you develop functionalities/services that can be so generic as to be reused in other cases (e.g. weather service, localization, rule engine, security, privacy & trust module)?*
9. *Is there a potential for services/features on top of the existing use case (i.e. also reusable by other use cases) that might add significant value, but is not yet foreseen in the use case planning?*
10. *Choose one: the use case already clearly specified in details (**5**) / a few implementation details are still to be solved (**4**) / the features are clear, but we need still to choose components and implementation details (**3**) / major technical features are still to be designed (**2**) / everything has still to be planned (**1**).*
11. *Please draw a sketch (block diagram) of the architecture deployed for this Use Case.*
12. *Do you want to add some additional information that could be useful for Architecture analysis activities in general?*

Lower-priority questions to be elaborated after the first days of activity

13. *For each component, please specify the name (brand/model) of the component, if available.*
14. *How are the main components connected with each other?*
15. *Which communication standards/formats are used in each link of the system?*
16. *Identify the reusable/generic software in relation to the specific use case components.*
17. *How is data flowing in this scenario? Which parameters are measured, exchanged, aggregated, processed in this use case? By which component? Using which format? Are* ***open standards*** *and* ***data formats*** *already in use in this scenario? For what purpose? In which component?*
18. *In your opinion, Is there some piece of data handled in this case which may be of interest for some actor also beyond this specific Use Case? Which one?*
19. *Are there some constraints related to data privacy and security in this Use Case? Which ones?*

20. *What are the main aspects in the Use Case that are not fully specified yet, or it will likely be subject to change during the course of the project?*

21. *Do you want to add some additional information that could be useful for Architecture Analysis?*

5.2. APPENDIX B - USE CASE ANALYSIS TEMPLATE

The template below have been used to collect a second round of inputs from UCs for architectural inputs during the early phases of the project, starting at the KOM.

Architecture sketch

Please draw a sketch (block diagram) of the systems deployed in this Use case.

Involved Actors (Users)

Please fill in the table below outlining the main actors using the systems deployed in this Use Case.

For each actor, please explain which kind of input/output data is accessed by the actor (and when) – and which are the main user interfaces (e.g. web-based dashboards, mobile apps, warnings received by e-mail or SMS, etc.).

Actor Name	Main features provided to the actor	Main data input/output actions	Main User interface(s) used

Deployed Components

Please fill-in below the main components deployed in this use case. Please be consistant with names used in the sketch (Section 4.1).

Name	Description	Supplier (brand) + Model	Number of units	Deployment Site(s)
[Please name deployed technology]	*[Please describe deployed technology]*	*[Please provide deployed technology supplier/brand and model]*	*[Please insert the number of used units]*	*[Please refer to No. from Area/Facilities Deployed table (see Section 4.3 of this document)]*

Re-usable components

Please specify, among the components above, if any of the components above are available/appropriate for re-use. If so, please specify the web page of the component, its license (for Software) and explain its potential for re-use.

Name	Web Page	License	Potential for re-use

Communication standards and formats

Please list all the communication standards and formats used in the use cases. *Please be consistant with names used in the sketch (Section 4.1).*

Interface name	Standard (s)	Notes

Data gathering

Data That Will Be Gathered							
No.	Data	Measurement Technique	Deployment Site(s)	Crops/animals used for task	Frequency of Data Collection	Task No.	Associated data model/format
1			*[Please refer to No. from Area/Facilities Deployed table (see Section 4.3 of this document)]*			*[Please refer to the Task number from the section 3 of this document]*	*[e.g. for each reading : weight measured (Kg) – serialized in a custom JSON frame + ISO 8601 timestamp]*

Are there some constraints to data privacy and security?

[free format]

5.3. LIST OF ACRONYMS

HLA	High Level Architecture
IoF2020	Internet of Food and Farm 2020
IoT	Internet of Things
RFID	Radio Frequency IDentification
UC	Use Case
UML	Unified Modeling Language

5.4. LIST OF FIGURES

5.5. LIST OF TABLES

5.6. BIBLIOGRAPHY AND WEB REFERENCES

[i] ISO/IEC/IEEE Systems and software engineering -- Architecture description," in ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000) , vol., no., pp.1-46, Dec. 1 2011 - doi: 10.1109/IEEESTD.2011.6129467

[ii] AIOTI WG03 – IoT Standardization, High Level Architecture (HLA), Release 3.0, June 2017, available on-line at https://aioti.eu/aioti-wg03-reports-on-iot-standards/

[iii] Fowler, Martin. Patterns of Enterprise Application Architecture. Addison Wesley, 2003, p. 116.

[iv] Internet of Things – Architecture IoT-A, Deliverable D1.5 – Final architectural reference model for the IoT v3.0, available at http://www.meet-iot.eu/iot-a-deliverables.html

[v] C. Roussey et al, Ontologies in Agriculture, available at http://liris.cnrs.fr/Documents/Liris-4759.pdf

[vi] VDMA, ISO, ISOBUS Data Dictionary, 2017-09-21, available at https://www.isobus.net/isobus/

[vii] Deployment diagrams show "the allocation of Artifacts to Nodes according to the Deployments defined between them." Unified Modeling Language, Superstructure, V2.1.2 p. 202

[viii] https://www.docker.com

[ix] ITU-T, Telecommunication Standardization sector of ITU, Recommendation ITU-T Y.2060, IoT Reference Model, Overview of the Internet of Things, June 2016, available at http://www.itu.int/itu-t/recommendations/rec.aspx?rec=Y.2060

[x] ANSI/ISA-95.00.01-2000, Enterprise-Control System Integration Part 1: Models and Terminology, available at https://isa-95.com/

[xi] Y. Tina Lee, Information Modeling: From Design to Implementation, Proceedings of the Second World Manufacturing Congress, 1999