Computers and Electronics in Agriculture 165 (2019) 104939

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Architecture framework of IoT-based food and farm systems: A multiple case study

Cor Verdouw[a,b,*], Harald Sundmaeker[c], Bedir Tekinerdogan[a], Davide Conzon[d], Teodoro Montanaro[d]

[a] *Information Technology Group, Wageningen University and Research, P.O. Box 35, 6700 AA Wageningen, the Netherlands*
[b] *Mprise Agriware, P.O. Box 598, 3900 AN Veenendaal, the Netherlands*
[c] *ATB – Institut für angewandte Systemtechnik Bremen GmbH, Wiener Straße 1, 28359 Bremen, Germany*
[d] *LINKS Foundation, Via Pier Carlo Boggio, 61, 10138 Torino, Italy*

ARTICLE INFO

Keywords:
Internet of Things
Architecture
Architectural framework
System of systems
Smart farming
Food supply chains

ABSTRACT

The Internet of Things (IoT) is expected to be a real game changer in food and farming. However, an important challenge for large-scale uptake of IoT is to deal with the huge heterogeneity of this domain. This paper develops and applies an architecture framework for modelling IoT-based systems in the agriculture and food domain. The framework comprises a coherent set of architectural viewpoints and a guideline to use these viewpoints to model architectures of individual IoT-based systems. The framework is validated in a multiple case study of the European IoF2020 project, including different agricultural sub sectors, conventional and organic farming, early adopters and early majority farmers, and different supply chain roles. The framework provides a valuable help to model, in a timely, punctual and coherent way, the architecture of IoT-based systems of this diverse set of use cases. Moreover, it serves as a common language for aligning system architectures and enabling reuse of architectural knowledge among multiple autonomous IoT-based systems in agriculture and food.

1. Introduction

Agriculture has a vital role in feeding the world in a healthy way. In recent decades, the agri-food sector has already realized big achievements in meeting critical challenges concerning food security, food safety, sustainability and health. These improvements mainly have been accomplished with non-digital technologies, such as mechanisation of field operations, animal and plant breeding and more eco-friendly farming methods. However, still a radical increase of productivity is needed to feed the ever-growing world population and to deal with challenges such as climate change, resource efficiency, animal welfare, waste reduction, food safety, and healthier consumer lifestyles.

The Internet of Things (IoT) is a very promising paradigm to drastically improve productivity and sustainability because it has the potential to achieve new levels of control (Sundmaeker et al., 2010; Porter and Heppelmann, 2014; Sarni and Kaji, 2016). IoT comprises smart webs of connected and context-sensitive objects that can be identified, sensed and controlled remotely (Atzori et al., 2010; Kortuem et al., 2010; Porter and Heppelmann, 2014; Verdouw et al., 2016b). As such it enables (AIOTI, 2015; Jayaraman et al., 2016; Verdouw et al., 2016a; Talavera et al., 2017):

- Better sensing of farming and food processing operations, including usage of inputs, crop growth, animal behaviour, food spoilage and resource utilization;
- Improving quality management and traceability by remotely monitoring the location and conditions of shipments and agricultural products;
- Better understanding of specific production circumstances, such as climate conditions, animal welfare, microbiological quality, pest pressure, and better knowledge about optimal interventions;
- More advanced and remote control of operations, enabled by actuators and robotics, e.g., precise application of pesticides and fertilizers, autonomous harvesting, or adjusting ambient conditions of food during transportation;
- Increasing consumer awareness of sustainability and health issues by personalised nutrition advices, health wearables and home automation.

However, the application of IoT in agriculture is challenging, especially because of a high uncertainty of business processes (Sundmaeker et al., 2016; Verdouw et al., 2016a). Agri-food products are living objects and farming is depending on natural conditions, such

[*] Corresponding author at: Wageningen University and Research, Information Technology Group, Hollandseweg 1, 6706 KN Wageningen, the Netherlands.
E-mail address: cor.verdouw@wur.nl (C. Verdouw).

https://doi.org/10.1016/j.compag.2019.104939
Received 12 April 2019; Received in revised form 30 July 2019; Accepted 4 August 2019
Available online 27 August 2019

as climate, weather, and diseases. In addition, there are many different types of production, e.g., arable farming, greenhouse cultivation or livestock farming. Farms and food processors also have to deal with many interrelated objects such as: (i) farm inputs, including seeds, feed, fertilizers or pesticides, (ii) farm resources, including farm land, stables, tractors and equipment, (iii) agricultural products including living animals, processed food and fresh produce; and (iv) logistic objects such as crates, containers and trucks. Moreover, IoT-based systems must not only support farmers, but also various stakeholders around the farm, including, e.g., contractors, agronomists, veterinarians, certification and inspection companies, authorities, consumers and input suppliers. Moreover, food supply chains are complex networks where many small companies do business with large multinationals.

This high variety and variability of processes, objects and stakeholders result in a large heterogeneity of IoT applications, which hampers a large-scale uptake of IoT in the agri-food domain. IoT-based systems are often fragmented, use different data platforms with limited interoperability and in particular more advanced applications are still in an early stage of development (Sundmaeker et al., 2016; Verdouw et al., 2017).

In order to overcome this situation, autonomous IoT systems should function as interoperable nodes within a well-aligned software ecosystem that maximizes reuse and synergies across multiple IoT systems. In such an ecosystem, technology companies can concentrate on the development of components that fit best to their core competencies (Manikas and Hansen, 2013; Kruize et al., 2016). Next, users can configure customized software systems from standardized components, which are supplied by multiple vendors that interact via common technological platforms (Verdouw et al., 2014). As such, IoT-based systems are no longer isolated, but integrated in a coherent way, leading to a System-of-Systems (SoSs) (Jamshidi, 2008; Nielsen et al., 2015; Tekinerdogan, 2017). In a SoSs, individual systems have managerial and operational independence, whereas the overall purpose of a system is to provide a function or service that cannot be provided by individual systems independently (Maier, 1998). Speaking the same architectural language is a key starting point to adapt systems to the SoSs' overall purpose (Fitzgerald et al., 2013; Nielsen et al., 2015).

This paper proposes such a common language for aligning system architectures and for enabling reuse of architectural knowledge among multiple autonomous IoT systems in agriculture and food. The objective is to develop and apply an architecture framework for modelling IoT-based systems in the agri-food domain. More specially, the paper will define a coherent set of architecture viewpoints and a guideline to use these viewpoints to model the architecture of individual IoT-based systems. The framework is validated in a multiple case study of the European IoF2020 project. The case study includes a diverse and coherent set of 19 IoT-based use cases that each provides a dedicated solution for a specific domain challenge.

The paper is structured as follows. Section 2 first provides some background information of IoT and software architecture, while Section 3 describes the research methodology. Section 4 introduces the designed framework, including a definition of the architecture viewpoints addressed and a guideline to apply the framework. Section 5 subsequently summarizes the application of the framework to the use cases by presenting the architectures of three representative systems into more detail. Finally, the main findings are summarized and discussed in Section 6.

2. Background

2.1. IoT for food and farming

The Internet is a global system of interconnected computer networks that uses the Internet protocol suite (TCP/IP) to link billions of devices worldwide. Nowadays, over 46% of the world population uses the Internet (InternetWorldStats, 2015). It has had a revolutionary impact on culture and commerce, including the rise of near-instant communication by electronic mail, instant messaging, Voice over Internet Protocol (VoIP) telephone calls, two-way interactive video calls, social networking, and online shopping sites. Moreover, Internet connectivity became the norm for many business applications and is today integral part of many enterprise, industrial and consumer products to provide access to information. However, the Internet usage still primarily focuses on human interaction and monitoring through apps and interfaces. IoT is a next stage of the Internet in which also physical things communicate.

IoT combines two concepts "Internet" and "Thing" and can therefore semantically be defined as "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols" (Infso and EPoSS, 2008). The concept was first introduced by Massachusetts Institute of Technology (MIT) Auto-ID Center to label the development towards a world where all physical objects can be traced via the internet by tagging them with RFID transponders (Schoenberger, 2002). In the meantime, its meaning is expanded towards a world-wide web of smart connected objects that are context-sensitive and can be identified, sensed and controlled remotely by using sensors and actuators (Atzori et al., 2010; Kortuem et al., 2010; Porter and Heppelmann, 2014; Verdouw et al., 2017). In the IoT domain every 'thing' is uniquely identifiable, equipped with sensors and connected in real-time to the Internet. As a result, Internet will be deeply embedded in the daily life of consumers and businesses. Invisible technology operates behind the scenes, dynamically responding to how people want "things" to act. IoT is expected to be the next Internet revolution. To date, the world has deployed about 5 billion "smart" connected things. Predictions indicate that there will be up to 50 billion connected devices by 2020 and in our lifetime we will experience life with a trillion-node network (Castaneda, 2015).

The agri-food is a challenging domain for IoT from a technical and organisational perspective (AIOTI, 2015; Jayaraman et al., 2016; Verdouw et al., 2016a; Talavera et al., 2017; Verdouw et al., 2017). 'Things' are often living and natural objects, such as plants, animals, square meters of soil and perishable food products. This means that IoT devices (e.g., microprocessors, sensors, antennas) cannot be easily embedded in products themselves. Furthermore, agricultural production is depending on natural conditions, such as climate (day length and temperature), soil, pests and diseases and weather. This results in a large variety and variability of agricultural things. Moreover, IoT devices have to operate in harsh environments (open air, cold storage, hot cleaning treatments, etc.) and remote areas (fields, stables, etc.). As a consequence, they need to be energy-autonomous and able to deal with Internet connectivity problems in rural areas. There is also a temporal IoT challenge, because the growth of food products is a relatively slow, seasonal process with many uncertainties (e.g., weather conditions) on the one hand, while at the other hand consumers ask for safe, healthy and fresh food, a whole year round and just-in-time delivery, minimizing waste and long best-before dates. A major organisational issue is the high number of small and medium sized enterprises, especially at agricultural production, trade and food industry, resulting in a lack of (financial) resources, technical expertise and management skills to invest successfully in IoT solutions. It also impacts user concerns among other about data ownership, privacy and security.

Consequently, current IoT applications and technologies in the agri-food domain are still fragmentary, lack seamless integration and especially more advanced solutions are in an experimental stage of development (Sundmaeker et al., 2016; Verdouw et al., 2017). Operational applications are mainly used by a small group of innovators and still focus on basic functionalities at a high granularity level. A large-scale uptake of IoT in agriculture is prevented among others by a lack of interoperability, user concerns among others about data ownership, privacy and security, and appropriate business models that are also suitable for (very) small companies.

2.2. Software architecture

Designing software architecture is a key activity in developing software-intensive systems. In fact, every software-intensive system has a software architecture, whether it is complex or simple. A software architecture describes the components of a system, interactions among components, and the interaction of a system as a whole with its environment (Tekinerdogan, 2014). A software architecture is an abstract representation that identifies the higher-level structure of a system and is important for supporting communication among stakeholders, for guiding design decisions, for supporting the subsequent development process, and for analysis of an overall system.

In this study the authors adopt the recommended practice for architecture description of ISO/IEC 42010 that defines concepts for modelling software architectures (ISO/IEC/IEEE, 2011). In this context, software architecture is assumed to meet specific stakeholder concerns. A stakeholder is defined as an individual, team, or organization with interests in, or concerns relative to, a system. A concern is defined as a matter of interest in the system that could be functional or related to quality issues.

Modelling architecture has been initially done in an informal manner, which often led to ambiguous representations of a system. It is now common practice to model software architecture using proper well-defined modelling approaches, which provide either visual notations or textual descriptions to represent an architecture in a precise unambiguous manner.

The architecture, is usually not drawn in one diagram but separated in multiple so-called architecture views each of which describes an architecture according to specific stakeholders' concerns (Clements et al., 2010). An architecture view is a representation of a set of system elements and relations associated with them to support a particular concern. Having multiple views helps to separate concerns and as such support modelling, understanding, communication and analysis of software architecture and the business processes to be supported for different stakeholders. Architecture views are defined for a particular system and need to conform to viewpoints that represent the conventions for constructing and using a view. An *Architecture Framework* is defined as a coordinated set of viewpoints that are used to define views. A more precise definition of architecture framework is given in the ISO standard: *"Conventions,* principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders" (ISO/IEC/IEEE, 2011).

Different architecture frameworks have been proposed in literature (Clements et al., 2010). Well-known examples include the Zachman framework (Sowa and Zachman, 1992), Kruchten's 4 + 1 view model (Kruchten, 1995), the Views and Beyond approach (Clements et al., 2010), the Reference Model of Open Distributed Processing (RM-ODP; ISO/IEC/IEEE, 2009) and the Open Group Architecture Framework TOGAF (Josey, 2018). Furthermore, several reference architectures have been developed for industrial automation, including ISA-95 (ISA, 2010a) and OPC Unified Architecture (OPC, 2017).

Initially, architectural frameworks were proposed with a fixed set of viewpoints. Because of different concerns that need to be addressed for different systems, it is currently widely recognized that a set of views should not be fixed but multiple viewpoints might be introduced instead. As such recent architectural frameworks, such as the Views and Beyond approach, provide mechanisms to adapt existing viewpoints, or to add new viewpoints.

2.3. Architecture of IoT-based systems

The architecture of IoT-based systems is similar to other information systems, but with special requirements concerning the remote identification, sensing and control of smart objects by using sensors and actuators. There are several initiatives working toward standardized architectures to overcome fragmentation in IoT development (Weyrich and Ebert, 2016). The Internet of Things—Architecture (IoT-A) provides a detailed view of IoT's information technology aspects (Carrez et al., 2013; Gubbi et al., 2013). The International Telecoms Unions (ITU) has developed an IoT Reference Model which provides a high level capability view of an IoT infrastructure (ITU-T, 2016). The Alliance for IoT Innovation (AIOTI) has defined a High Level IoT Architecture to achieve IoT semantic interoperability (AIOTI, 2018).

The focus of these IoT reference architectures is on technical aspects of IoT. They include viewpoints to visualize how objects are sensed and controlled by IoT technologies, but do not sufficiently cover a complete information modelling process from requirements definition and business modelling to detailed implementation models. The framework presented in this paper will add these viewpoints to address the completeness required for IoT-based systems in the agriculture and food domain.

3. Methodology

3.1. Case study setup

The development of reference architectures is typically design-oriented research that aims at solving a certain problem by constructing a new artefact (Hevner et al., 2004; Van Aken, 2004; March and Storey, 2008). Artefacts for real-life problems are influenced by many factors. Case studies can deal with such complex phenomena, which cannot be studied outside their rich, real-world context (Benbasat et al., 1987; Eisenhardt, 1989; Yin, 2002).

The present research has conducted a developing multiple-case study, which is a type of action research that develops and tests technological rules in close collaboration with people in the field (Van Aken, 2004). In such research, an individual case is primarily oriented at solving a local problem. Following a reflective cycle, after each case a researcher develops knowledge that can be transferred to similar contexts on basis of reflection and cross-case analyses (Eisenhardt, 1989; Hevner et al., 2004; Van Aken, 2004). To this aim, the present study has defined architecture viewpoints based on literature regarding existing architectural frameworks. These viewpoints served as a theoretical basis for abstracting replicable knowledge from case study findings (Yin, 2002).

For the purposes of this paper, the cases were selected to reflect the diversity of the food and farming domain, i.e. an heterogeneous selection based on theoretical replication logic (Eisenhardt, 1989; Yin, 2002). In total 19 cases were selected as being representative for different agricultural sub sectors, conventional and organic farming, early adopters and early majority farmers, and different supply chain roles, including farming, processing, logistics and consumption. The approach for the cases is a combination of a lean start-up methodology that focuses on the development of Minimal Viable Products (MVPs) in short iterations and a multi-actor approach that stresses an active involvement of various stakeholders. Each case focused on the development of IoT-based solutions for specific business needs, done by a dedicated team of agri-food users (e.g. farmers, processing companies, or logistic service providers) and IoT companies (integrators, app/service developers, infrastructure/technology providers) with a clear commercial drive, supported by R&D organisations.

3.2. Overview of the cases

The case study was carried out as part of the European IoF2020 project in close interaction with the involved business partners [www.iof2020.eu]. It included 19 use cases that are organized in 5 coherent trials that aim to address the most relevant challenges for the concerned sub sector.

The *Internet of Arable Farming (trial 1)* integrates operations across the entire arable cropping cycle combining IoT technologies, data acquisition (soil, crop, climate) in growing and storage of arable crops (potatoes, wheat and soya beans). These are linked to existing sensor networks, earth observation systems, crop growth models and yield gap analysis tools and external databases (e.g., economic/environmental impact) as well as translated into farm management systems. The trial aims to result in increasing yields, less environmental impact, easier cross-compliance and product traceability and more use of technology by farmers.

The *Internet of Dairy Farming (trial 2)* implements, experiences and demonstrates the use of real-time sensor data (e.g., neck collar) together with Global Positioning System (GPS) location data to create value in a chain from 'grass to glass', resulting in a more efficient use of resources and production of quality foods, combined with a better animal health, welfare and environment implementation. The trial focuses on feeding and reproduction of cows through early warning systems and quality data that can be used for remote calibration and validation of sensors.

The *Internet of Fruits (trial 3)* demonstrates IoT technology that is integrated throughout a whole supply chain from farm, logistics, processing to retail. Sensors in orchards and vineyards (including weather stations, multispectral/thermal cameras) are connected through the cloud and used for monitoring, early warning of pests and diseases and control (e.g., variable rate spraying, selective harvesting). Traceability devices (including Radio Frequency IDentification - RFID, multi-dimensional barcodes) and smart packaging allows for condition monitoring during storage, processing, transportation and on shelves. Big data analysis will further optimize all processes in a whole chain. This is intended to result in reduced pre- and post-harvest losses, less inputs, higher (fresh) quality and better traceable products (including Protected Designation of Origin, PDO).

The *Internet of Vegetables (trial 4)* focuses on a combination of environmental control levels: full-controlled indoor growing with an artificial lighting system, semi-controlled greenhouse production and non-regulated ambient conditions in open-air cultivation of vegetables. It demonstrates the automatic execution of growth recipes by intelligent combination of sensors that measure crop conditions and control processes (including lighting, climate, irrigation and logistics) and analysis of big data that is collected through these sensors and advanced visioning systems with location specification. This is intended to result in improved production control and better communication throughout a supply chain (including harvest prediction, consumer information).

The *Internet of Meat (trial 5)* demonstrates how animal growth (individual and group level) can be optimized and communication throughout a whole supply chain can be improved, based on automated monitoring and control of advanced sensor-actuator systems. The data generated by events is also be used for early warning (e.g., on health status) and for improving the transparency and traceability of meat. This will assure meat quality, reduce mortality, optimize labour and improve animal health and welfare leading to reduction of antibiotics. Table 1 provides an overview of the use cases of each trial, classified according to the selection criteria for addressing the diversity of the food and agricultural domain.

3.3. Research phasing

The research was organised in four steps: (A) definition of architecture viewpoints and modelling guidelines (framework), (B) requirements definition and generic analysis of the use cases, (C) modelling of the use case architectures and validation, and (D) overall analysis and update method.

Firstly, a review of literature on IoT in agri-food and software architecture was performed and various existing software architecture frameworks and IoT reference models were investigated. Based on the literature review, the viewpoints and modelling guidelines were defined and validated by a team of 39 system architects and information analysts.

Secondly, the general context and requirements of the use cases were identified and subsequently analysed through desk research of use case documentation and semi-structured interviews with the lead architect of every use case (thus involving 19 interviews). The interviews were conducted based on a questionnaire with a mix of open and closed questions. The most relevant questions were related to the core idea and objective of a use case, main business processes targeted, objects addressed, main actors using the envisaged system, the main functionalities/services to be provided to end-users, non-functional requirements, high-level technology components envisioned and available documentation. The interviews and desk research resulted in a definition of requirements to be satisfied by the architectural models and an update of the viewpoints and guidelines. At this, it was necessary to balance the expressiveness of analysis and the ease of interpretation by agri-food experts with no strong background in IoT, and, more in general in system engineering.

Thirdly, the architecture of each use case was modelled by analysts, who applied the viewpoints and guidelines to the cases based on information gathered in the previous phase. The resulting case-specific models were reviewed by the use case teams and iteratively refined by the use case analyst.

Finally, the team of system architects and information analysts conducted an overall analysis of the applied models to identify commonalities and differences of use case architectures and to identify lessons learned for the overall method. The results of this analysis were evaluated by the use case teams in three project workshops and giving bilateral feedback. This evaluation resulted in the final version of the architectural method, which is described in the present paper.

The remainder of this paper introduces the results following the research steps as described above.

4. Design architecture framework

4.1. Viewpoint definition

Before modelling software architecture, it is important to define an architectural framework of common conventions, principles and practices for addressing specific concerns of the corresponding systems (ISO/IEC/IEEE, 2011). A framework comprises a coherent set of viewpoints for its stakeholders and as such it serves as a common language and a basis for software development. In this study, the following basic requirements for selection of viewpoints are identified:

- R1: they must support a seamless translation of business process design to detailed information engineering models;
- R2: they must visualize how objects are sensed and controlled by IoT technologies;
- R3: they must support interoperability and reuse of system components;
- R4: they must provide insight in the essence of use case systems in a consistent, concise but also simple way, not overcharging the use case owners.

The authors have first analysed existing architecture frameworks that fulfil the requirements of IoT-based systems in the agri-food domain (see Section 2). It was concluded that none of the analysed architecture frameworks has a direct fit with the described scenario. Generic software architecture frameworks do provide a coherent set of viewpoints for business design to technical implementation (R1), but miss viewpoints for modelling the IoT dimension (R2). IoT-specific frameworks focus on technical IoT aspects (R2), but lack especially business viewpoints (R1). Both types of architecture frameworks include useful views for interoperability and reuse (R3). Furthermore, for modelling the essence of use case systems, excluding details that are not important for the purpose of this research, a limited set of viewpoints can be selected from both types of frameworks (R4).

Based on this analysis, the following six viewpoints are defined:

Table 1
Overview of the cases.

Trial/sector	Case	Challenge	Focal Country	Chain Role	Adopter Type	Conventional/Organic
Arable	1.1 Within-field management zoning	Defining specific field management zones by developing and linking sensing- and actuating devices with external data	NL	Farming, Logistics	Early adopters and majority	Both
Arable	1.2 Precision Crop Management	Smart wheat crop management by sensors data embedded in a low-power, long-range network infrastructure	FR	Farming	Majority	Conventional
Arable	1.3 Soya Protein Management	Improving protein production by combining sensor data and translate them into effective machine task operations	AT, IT	Farming	EarlyAdopters	Both
Arable	1.4 Farm Machine Interoperability	Data exchange between field machinery and farm management information systems for supporting cross-over pilot machine communication	DK	Farming	Majority	Conventional
Dairy	2.1 Grazing Cow Monitor	Monitoring and managing the outdoor grazing of cows by GPS tracking within ultra-narrow band communication networks	BE	Farming	Both	Both
Dairy	2.2 Happy Cow	Improving dairy farm productivity through 3D cow activity sensing and cloud machine learning technologies	NL	Farming	EarlyAdopters	Both
Dairy	2.3 Silent Herdsman	Herd alert management by a high node count distributed sensor network and a cloud-based platform for decision-making	UK	Farming	Majority	Conventional
Dairy	2.4 Remote Milk Quality	Remote quality assurance of accurate instruments and analysis & pro-active control in a dairy chain	NL	Processing, Consumption	Majority	Both
Fruit	3.1 Fresh table grapes chain	Real-time monitoring and control of water supply and crop protection of table grapes and predicting shelf life	IT, GR	Farming, Logistics	Early Adopters	Both
Fruit	3.2 Big wine optimization	Optimizing cultivation and processing of wine by sensor-actuator networks and big data analysis within a cloud framework	FR	Farming, Processing	Early Adopters	Both
Fruit	3.3 Automated olive chain:	Automated field control, product segmentation, processing and commercialisation of olives and olive oil	SP, IT	Farming, Processing, Logistics	Majority	Conventional
Fruit	3.4 Intelligent fruit logistics	Fresh fruit logistics through virtualization of fruit products by intelligent trays within a low-power long-range network infrastructure	GE	Logistics, Consumption	Majority	Both
Vegs	4.1 City farming	Value chain innovation for leafy vegetables in convenience foods by integrated indoor climate control and logistics	NL	Farming, Logistics	Early Adopters	Conventional
Vegs	4.2 Chain-integrated greenhouse production	Integrating the value chain and quality innovation by developing a full sensor-actuator-based system in tomato greenhouses	SP	Farming, Logistics, Consumption	Majority	Both
Vegs	4.3 Added value weeding data	Boosting value chains by harvesting weeding data of organic vegetables obtained by advanced visioning systems	NL, AT	Farming	Majority	Organic
Vegs	4.4 Enhanced quality certification system	Enhanced trust and simplification of quality certification systems by use of sensors, RFID tags and intelligent chain analyses	IT	Farming, Logistics, Consumption	Majority	Both
Meat	5.1 Pig farm management	Optimise pig production management by interoperable on-farm sensors and slaughter house data	BE, NL	Farming, Processing, Consumption	Both	Both
Meat	5.2 Poultry chain management	Optimize production, transport and processing of poultry meat by automated ambient monitoring & control and data analyses	SP	Farming, Logistics, Processing	Majority	Conventional
Meat	5.3 Meat Transparency and Traceability	Enhancing transparency and traceability of meat based on an monitored chain event data in an EPCIS-infrastructure	GE, NL	Farming, Logistics, Processing, Consumption	Majority	Both

- *Domain model viewpoint*: general view of key functional aspects of an IoT-based system concerning the key actors, main physical entities (e.g., animals, goods, equipment), main IoT components and its interactions, this viewpoint provides a common understanding and terminology for other views and is based on the logical view of the 4 + 1 framework (Kruchten, 1995);
- *Business process hierarchy viewpoint*: overview of business processes and their interrelations, including the physical product flow of input material to end products, the main objects (things) involved and the position of business processes in the production control hierarchy, ranging from operational control of physical objects to enterprise management level, based on the ISA95 reference model (ISA, 2010a);
- *IoT layer viewpoint*: classifies IoT functionalities into different technical layers ranging from device layer until application layer, as such it provides an overview of the technical architecture and allows to identify suitable technology providers; this viewpoint is based on the ITU-T Y.2060 IoT Reference Model (ITU-T, 2016);
- *Deployment viewpoint*: visualizes the location of hardware and software components and how they are deployed; as such it defines a detailed technical architecture, based on the physical view of the 4 + 1 framework (Kruchten, 1995);
- *Information model viewpoint*: depicts the data entities of an IoT-based system, including data models of databases used, specifications of raw data collected by deployed IoT sensors, standard identification schemas, data entities in communication protocols, etc.; this viewpoint is based on the RM-ODP framework (Raymond, 1994);
- *Interoperability endpoints viewpoint*: defines main interfaces for integration with external systems including standards and protocols to be used, derived from the information model viewpoint; it helps to identify potential technical synergies among IoT-based systems.

These viewpoints will be described in the following subsections: domain model viewpoint (Section 4.2), business process hierarchy viewpoint (Section 4.3), IoT layer viewpoint (Section 4.4), deployment viewpoint (Section 4.5), information model viewpoint (Section 4.6), and interoperability endpoints viewpoint (Section 4.7). Moreover, a guideline to apply these viewpoints will be introduced in Section 4.8.

4.2. Domain model viewpoint

The domain model view is used to provide a general view of main concepts and relationships for the case being analysed. The naming and identification of these concepts and relationships provide a common understanding and terminology for other views. More specifically, this viewpoint summarizes in one diagram the key functional aspects of the cases concerning: the key actors, main objects and physical entities involved (e.g., animals, goods, equipment), the main IoT components, and how these entities interact with each other and to obtain what. For example, such entities could be IoT architectural entities like the ones described in the Architectural Reference Model IoT-A adopted by AIOTI (Carrez et al., 2013; AIOTI, 2018) and ontologies for agriculture and farming (Roussey et al., 2019), as well as agricultural data dictionaries, like the ISOBUS Data Dictionary (VDMA, 2019). The domain model is an introductory viewpoint in many software architecture frameworks. In particular we adopted the logical view of the 4 + 1 architecture framework (Kruchten, 1995), which uses UML class diagrams for representation (OMG, 2011).

The key aspects addressed by the domain model viewpoint are provided in Table 2.

4.3. Business process hierarchy viewpoint

The business process hierarchy view shows an overview of business processes and their interrelations. Two dimensions are added to regular business process models to adapt it to the specific characteristics of IoT. First, the viewpoint visualizes the physical product flow of input material to end products and the main objects (things) involved. Second, the main business processes for planning and control of the physical flow are placed in different layers, based on their position in the production control hierarchy. These layers address different time horizons, ranging from *operational control of physical objects* to *enterprise management level*. The levels are defined using a standard approach, defined for the industrial domain, but also suitable for farming and food systems: the ISA-95 reference model (ISA, 2010a).

ISA-95, formerly known as S95, is a framework that focuses on integration of office automation and production automation and mechanization (ISA, 2010a; Verdouw et al., 2015). It is widely adopted in the international production industry, among others, in the pharmaceutical, petrochemical and food processing sectors. ISA-95 consists of models and terminology about: (i) information exchange between enterprise management systems and manufacturing operations systems; (ii) activities in manufacturing operations systems; and (iii) exchanged information within manufacturing operations systems. More specifically, ISA95 addresses four control levels, which are based on the Purdue Reference Model (Williams, 1994; ISA, 2010b):

- Level 0 & 1: the actual physical processes and its sensing and actuation;
- Level 2: manufacturing operations management systems that supervise, monitor and control physical processes, especially Supervisory Control and Data Acquisition systems (SCADA), Programmable Logic Controllers (PLC) and Distributed Control Systems (DCS);
- Level 3: systems, which manage the workflow of batch, continuous or discrete production operations, especially Manufacturing Execution Systems (MES);
- Level 4: business planning & logistics systems that manage business-related activities of production, including production planning and scheduling, material use, shipping and inventory management, especially in Enterprise Resource Planning (ERP) systems.

Fig. 1 shows the four layers of the Business Process Hierarchy View. The Management Information Layer contains processes related to the control of an entire enterprise (e.g., a farm), on an aggregate level and with the longest time horizon (months, weeks, days). The Operations Execution Layer contains processes related to the definition, control and performance of tasks, with an intermediate time horizon (days, hours, minutes). The Production Control Layer contains processes related to the execution of tasks by equipment and humans, with shortest time horizon (minutes, seconds, milliseconds). Finally, the Physical Object Layer shows the relation to objects in the physical world, eventually including IoT sensors and actuators. These objects can be fields, stables, animals, plants, farm equipment, processing facilities, containers, boxes, trucks, but also humans like employees or consumers.

The business process hierarchy viewpoint is defined in Table 3.

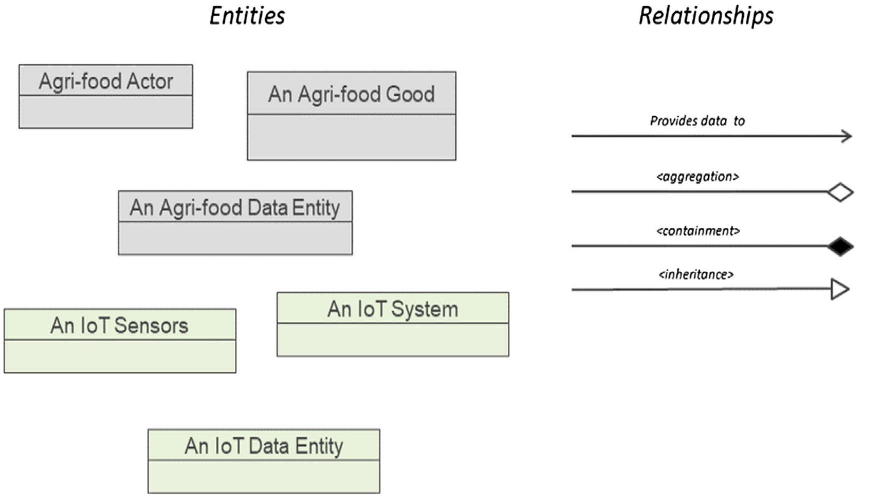
4.4. IoT layer viewpoint

The IoT layer view is used to classify each component from a technical IoT perspective. It supports categorizing each component of an use case in a way that allows to identify suitable providers of infrastructure or technology capable to offer such component. As such it shows a concrete overview of the mapping between adopted components and common IoT functionalities, indicating to which layer of the reference model, a particular component belongs to.

The IoT layer view is aligned with main on-going IoT trends and standardization efforts, like, for example, the recommendation by AIOTI WG03 (AIOTI, 2018). To do this, the main features of an IoT-based system are depicted inside the ITU-T Y.2060 IoT Reference Model (ITU-T, 2016) (see Fig. 2).

The viewpoint contains the following layers:

Table 2
Summary of the domain model viewpoint.

Name	Domain model
Concerns	functionality, usage, system purposes, system features, system properties, structure, behaviour, modularity, control, inter-process communication, complexity
Stakeholders	IoT related stakeholders: technology providers, infrastructure providers, integrators. Agri-food related stakeholders
Elements	Agri-food Actor, Agri-food Good, Agri-food Data Entity, IoT Sensors, IoT System, IoT Data Entity
Relations	Provides data to, aggregation, containment, inheritance
Constraints	No constraints, every relationship can be used for every entity
Notation	 Entities Relationships Agri-food Actor An Agri-food Good An Agri-food Data Entity An IoT Sensors An IoT System An IoT Data Entity Provides data to <aggregation> <containment> <inheritance> Existing elements in gray, components and entities to be developed in green

- Application layer, which contains IoT applications;
- Service support and application support layer, which contains capabilities that can be used by several IoT applications;
- Network layer, which provides control functions of network connectivity and connectivity for transport of application-specific data;
- Device layer, subdivided in device capabilities, which are network capabilities at device level, and gateway capabilities, which are network capabilities at gateway level, e.g., multiple interfaces support, protocol conversion;
- Management capabilities: cross-layer capabilities related to an IoT network, e.g., device management, traffic and congestion management;
- Security capabilities: cross-layer capabilities at application, network and device layer.

The IoT layer viewpoint is defined in Table 4.

4.5. Deployment viewpoint

The deployment view visualizes the physical deployment of an IoT-based system, i.e. location of hardware and software components and how they are deployed. The view is used to show a concrete explanation

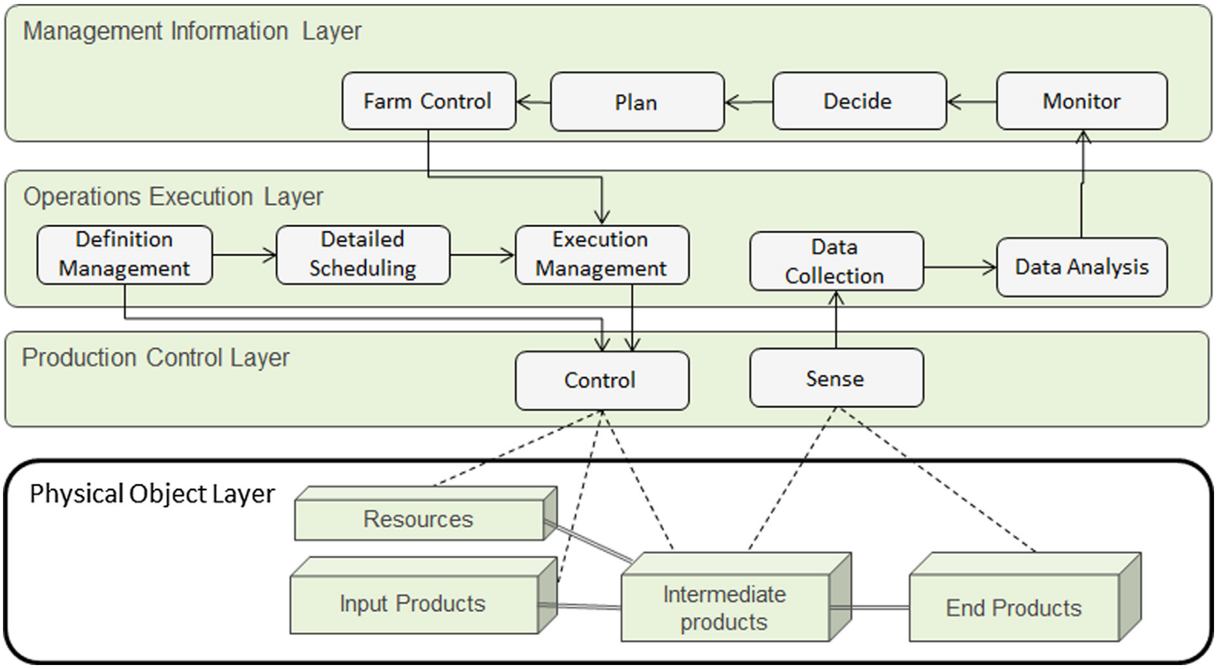
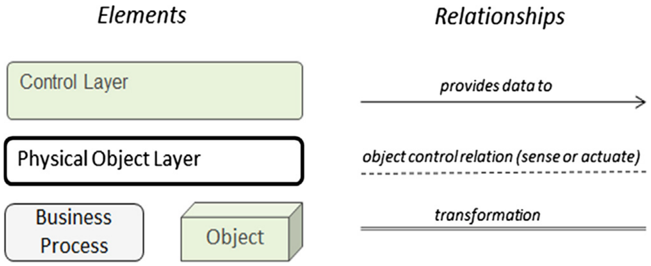


Fig. 1. Generic example of the process hierarchy view.

Table 3
Summary of the business process hierarchy viewpoint.

Name	Business Process Hierarchy
Concerns	System purposes, control activities undertaken by a system, objects addressed by an IoT system
Stakeholders	business decision-makers, product managers, business analysts, system architects, developers and integrators
Elements	Objects, Business Processes, Control Layers
Relations	Transformations, Information Flows, Object Control Relation (sense or actuate)
Constraints	● Objects can only be defined in the physical object layer ● Transformations can only connect objects ● Business processes can only be defined in Control Layers ● Information Flows can only connect Business Processes ● Object Control Relations have to connect objects and business processes of the Production Control Layer
Notation	 Elements Relationships Control Layer Physical Object Layer Business Process Object provides data to object control relation (sense or actuate) transformation

of the deployment of systems, their location and the way they are integrating in an use case scenario. Specifically, it depicts a list of the components (hardware and software), indicating where they are installed (for example identifying the specific software that is deployed on the related computer); and requirements in terms of connectivity for each component.

The deployment viewpoint is included in most software architecture frameworks. The present framework in particular adopted the physical view of the 4 + 1 architecture framework (Kruchten, 1995), which uses UML deployment diagrams for its representation (OMG, 2011).

The deployment viewpoint is defined in Table 5.

4.6. Information model viewpoint

The information model view (Lee, 1999) is used to model all data entities of an IoT-based system, including data models representing used databases, specifications of raw data collected by deployed IoT sensors, standard identification schemas, data entities in communication protocols, etc. The view is used to show in a single diagram the basic data-related aspects of UCs, indicating data entities present in this use case, their format and from which systems they are handled. Furthermore, it indicates if data can be mapped using standard ontologies or taxonomies, the relation among data entities of an use case and finally how they are transcoded, converted, mapped and elaborated.

The information model viewpoint is included in most software architecture frameworks. The present framework in particular adopted the information view of RM-ODP framework (Raymond, 1994), which uses UML Entity Relationship Diagrams for its representation (OMG, 2011).

The information model viewpoint is defined in Table 6.

4.7. Interoperability endpoints viewpoint

This interoperability endpoints view indicates main endpoints that can be leveraged for integration of different systems. The objective is to identify most suitable interfaces for interacting with legacy and IoT systems deployed in each use case and at the same time the set of standards and protocols to be used.

This view is complementing the information model viewpoint, because information is usually distributed across other views. The interoperability endpoints allow for the identification of potential technical synergies.

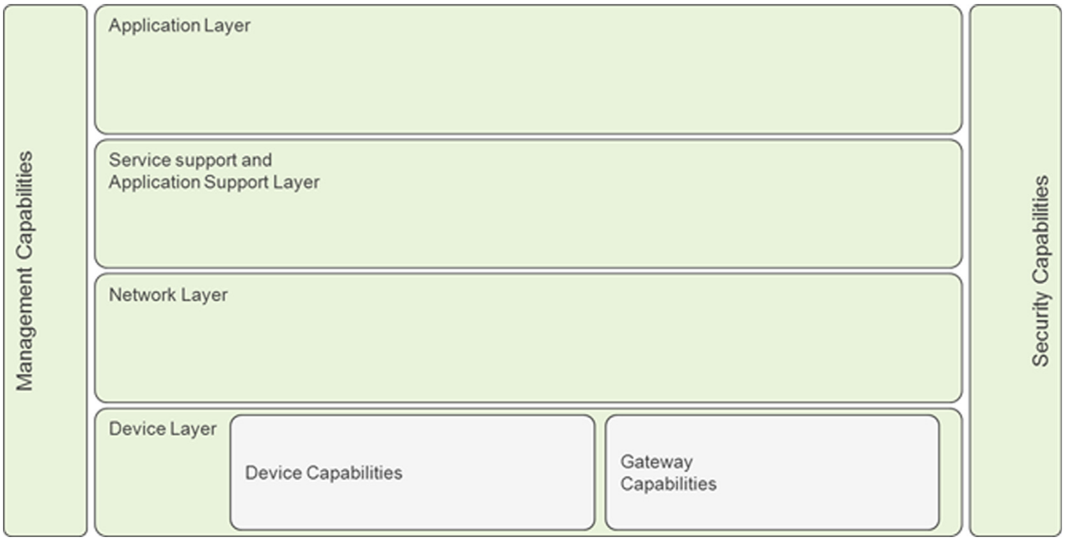
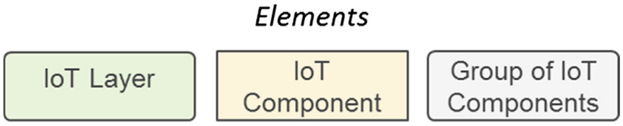


Fig. 2. The ITU-T Y.2060 reference model.

Table 4
Summary of the IoT layer viewpoint.

Name	IoT layer
Concerns	Map between adopted components and common IoT functionalities, system deployment, system localization, system integration
Stakeholders	IoT related stakeholders: technology providers, infrastructure providers, integrators
Elements	Management Capabilities, Application Layer, Service Support and Application Support Layer, Network Layer, Device Layer, Device Capabilities, Gateway Capabilities, Security Capabilities
Relations	n/a
Constraints	Entities cannot overlap more layers
Notation	 Elements IoT Layer IoT Component Group of IoT Components

The interoperability endpoints viewpoint is defined in Table 7.

4.8. Guideline for applying the architecture framework

In the previous subsections we have described six viewpoints that together form an architecture framework for modelling IoT based agri-food systems. In this section we introduce a guideline, i.e. a general method, for applying these viewpoints.

Fig. 3 shows the activities of this method in a Business Process Modelling Notation (BPMN) diagram. The first step is identifying an organization's objectives and stakeholder concerns, followed by developing a domain model view and a business process hierarchy view. The domain model addresses the objectives and concerns defined from the corresponding application domain perspective, while the business process hierarchy view represents its business and organizational concerns. The consistency of both models, including its ontology, must be checked carefully. For example: IoT sensors in a domain model should be compliant with sensing processes in a business process hierarchy model.

The subsequent step is the development of an information model view that focuses on data aspects. Information models must be consistent with domain and business process hierarchy models. Its elements should be compliant with the data entities included in a domain model and with the provide-data-to relations in a business process hierarchy model.

Subsequently, an IoT layer view is developed for addressing technical IoT concerns. Again, this view must be consistent with the previous models. Especially, a modeller should check if the IoT elements of the domain model are addressed. Concerning the business process hierarchy view, all processes must be aligned with the application layer components, while the sense and control processes should be addressed in the device layer.

The next view to be modelled is a deployment view for allocating software elements to hardware nodes, and in parallel the development of an interoperability endpoint view for addressing interoperability concerns. The components of a deployment diagram should especially be in accordance to the IoT system elements of a domain model and the IoT components of an IoT layer model. Interoperability endpoints should especially be aligned with the network layer of IoT layer views and the components of deployment models.

The final step is validation and a final consistency check of the developed views. This activity also checks if there are any updates in the viewpoints as defined in the framework. In case of inconsistencies, missing issues and/or updates of the framework, the process iterates to the first steps, which implies that the modelled views are improved.

The above represents a general method that is based on the COST-WORTH approach (Kirchhoff et al., 2004) and that is further developed from experiences in the project. For achieving this level, several

Table 5
Summary of the deployment viewpoint.

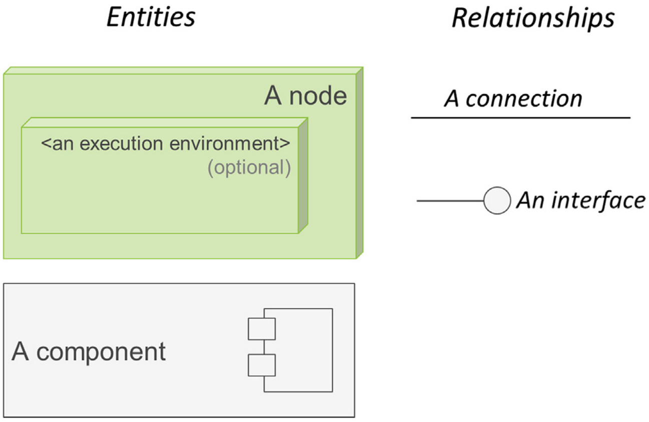
Name	Deployment View
Concerns	Functionality, usage, system features, structure, behaviour, resource utilization, reliability, complexity, evolvability, cost, flexibility, agility, modifiability, modularity, control, inter-process communication, subsystem integration, data accessibility, maintainability
Stakeholders	Technology providers, infrastructure providers, integrators, maintainers
Elements	Node, Execution Environment, Component
Relations	Connection, interface
Constraints	Execution environments need to be indicated only if there are two or more of them in one node
Notation	UML deployment Diagram (OMG, 2011)  Entities Relationships A node <an execution environment> (optional) A connection An interface A component

Table 6
Summary of the information model viewpoint.

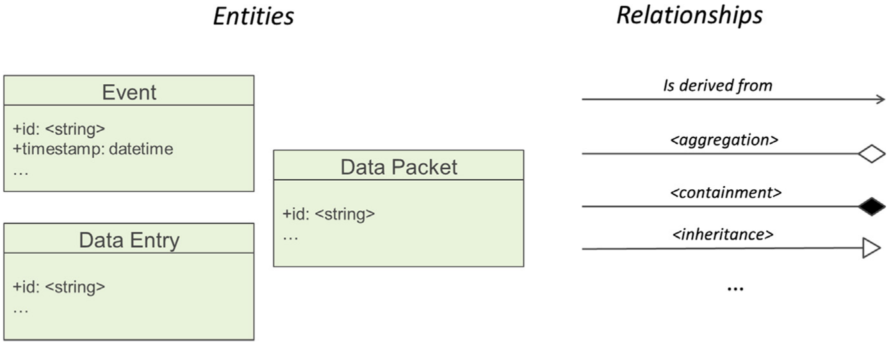
Name	Information Model
Concerns	Usage, system properties, structure, complexity, modifiability, inter-process communication
Stakeholders	Integrators, Developers, Maintainers
Elements	Events, Data Packet, Data Entry
Relations	Is derived from, aggregation, containment, inheritance
Constraints	There are no constraints, each relation can be used by each entity
Notation	UML diagram, accompanied by a descriptive table providing details about the most relevant data entities 

Table 7
Summary of the interoperability endpoints viewpoint.

Name	Interoperability Endpoints
Concerns	Inter-process communication, data accessibility.
Stakeholders	3d party system Integrators
Elements	n/a
Relations	n/a
Constraints	Each row of the table must contains a description of an interface and the associated protocol to be used.
Notation	Table with a list of interfaces

Interface name	Exposed by	Protocol	Notes

iterations were made in practice. The use case teams were assisted in specifying envisaged IoT-based solutions. In an early phase, when aiming at the set-up of the overall consortium of over 70 partner organisations, they were asked to prepare a basic use case description, outlining their underlying objectives and critical business processes to be improved clearly identifying stakeholder concerns. Based on the motivation to install a continuous ICT-based business improvement process, the modelling elements as described in the previous sections were selected and compiled to facilitate the realisation of a systematic and harmonised approach when describing all 19 use cases.

The initial use case description was a reference document to analyse and design the future realisation of related business processes. Specifically, the domain model and business process hierarchy viewpoints were used to understand the relation of different process and technical components that need to be reused, adapted, or developed. To ensure the relation of organisational objectives over different abstraction layers, validation and consistency checks of views are essential as well as facilitating an understanding of the relations between real, digital and virtual world aspects of IoT-based solutions. The elaboration of IoT layer and deployment views benefits from a realisation in different iterations, since each iteration cycle helps to also validate the technical feasibility and to plan specific iterations of envisaged minimum viable products (MVPs).

Next section will describe the application of the proposed framework to the cases.

5. Application of the framework to IoF2020 use cases

The presented framework is applied to 19 use cases in different regions of Europe (see Table 1) to demonstrate its effectiveness for modelling the architectures of IoT-based systems in a large spectrum of different agricultural domains and applications. The following paragraphs provide an overview of the application of the proposed architectural framework to the use cases of the present study.

5.1. Overview use case architectures

The framework is used to model use case architectures in the same way, resulting for every use case in a coherent set of standardized views, i.e. applications of the viewpoints as defined in the framework. Table 8 summarizes how many elements are used within each viewpoint to indicate the level of detail of the applied use case architectures. Generally, the IoT-based solutions, realized in the use cases, are quite heterogeneous due to differences in the agri-food sectors and business challenges addressed. Therefore, the individual architectures are including diverse sets of elements. For instance, a broad range of networking technologies and a large number of different cloud platforms are employed within the different use cases. Use cases implement diverse sensors and measure a vast array of data dimensions, including animal and crop features, outdoor and indoor conditions, as well as soil characteristics. The majority of use cases moves beyond passive portals with sensor data and introduce intelligent, task-specific decision support for agri-food professionals and/or fully autonomous control loops that automatically trigger actuators based on sensor data and statistical data processing. State of the art, IoT-specific networking protocols such as Lora and Sigfox are deployed by a large number of use cases.

For the purpose of this paper and for brevity reasons, the remainder of this section presents the architectures of three use cases (out of 19) to show the application of the modelling method for each addressed viewpoint. We have chosen use cases that illustrate the diversity of the

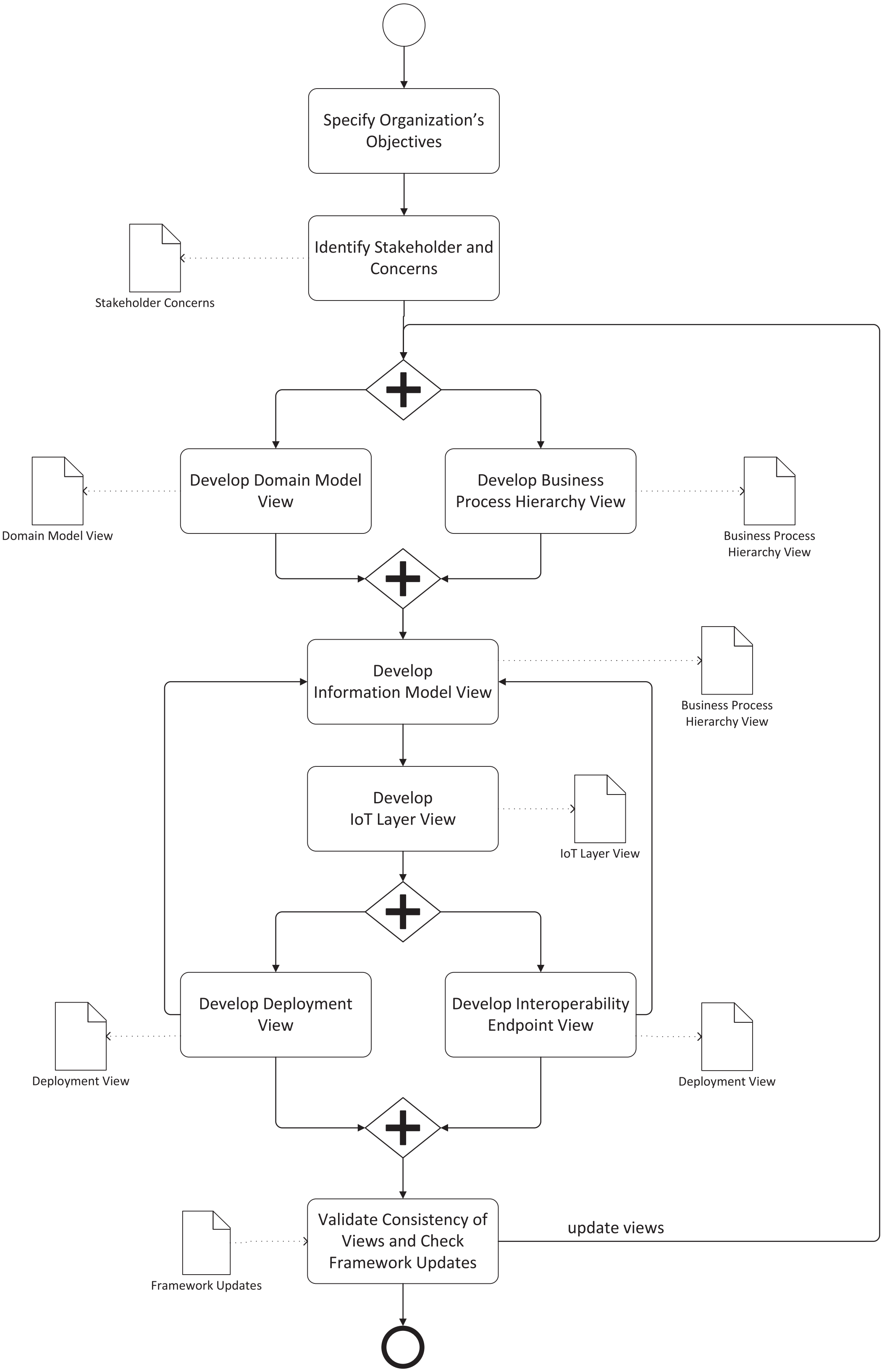

Specify Organization's Objectives
Identify Stakeholder and Concerns
Stakeholder Concerns
Develop Domain Model View
Domain Model View
Develop Business Process Hierarchy View
Business Process Hierarchy View
Develop Information Model View
Business Process Hierarchy View
Develop IoT Layer View
IoT Layer View
Develop Deployment View
Deployment View
Develop Interoperability Endpoint View
Deployment View
Validate Consistency of Views and Check Framework Updates
Framework Updates
update views

Fig. 3. Guideline for applying the Architecture Framework.

domain, especially concerning the subsectors, challenges and supply chain roles. The following Table 9 summarizes the three selected use cases. For a description of all use case architectures we refer to (Tomasi et al., 2018).

5.2. *Architecture use case 'Added Value Weeding Data'*

With the growth of organic vegetables, weeding represents one of the most important and frequent activities to control both the quality of

Table 8
Number of main elements addressed in each viewpoint.

Use case	Domain: concepts	Layer: IoT functions	Business Process Hierarchy: Objects/ Processes	Deployment: nodes/ components	Information: data elements	Interoperability Endpoints
1.1 Within-field management zoning	15	14	9/16	7/16	N.A.	2
1.2 Precision Crop Management	16	19	5/12	16/38	25	13
1.3 Soya Protein Management	12	9	4/10	5/10	N.A.	3
1.4 Farm Machine Interoperability	19	23	5/15	5/16	19	4
2.1 Grazing Cow Monitor	12	8	6/11	3/6	9	2
2.2 Happy Cow	16	9	4/10	5/11	19	3
2.3 Silent Herdsman	8	8	5/15	5/9	8	4
2.4 Remote Milk Quality	17	13	7/16	6/15	14	5
3.1 Fresh table grapes chain	21	13	9/17	6/26	18	1
3.2 Big wine optimization	37	23	12/21	12/22	7	N.A.
3.3 Automated olive chain:	16	12	7/16	5/11	9	5
3.4 Intelligent fruit logistics	19	18	11/18	6/16	18	9
4.1 City farming	16	10	N.A.	5/9	7	4
4.2 Chain-integrated greenhouse production	8	8	6/12	5/9	5	5
4.3 Added value weeding data	26	18	6/17	11/49	31	6
4.4 Enhanced quality certification system	20	22	9/18	9/22	19	11
5.1 Pig farm management	23	12	7/17	8/17	25	12
5.2 Poultry chain management	20	10	8/15	15/31	19	10
5.3 Meat Transparency and Traceability	10	10	0/6	5/9	11	3

a field and its produce. In recent years, automated intra-row weeding machines have entered the market, greatly facilitating the weeding process. The most advanced weeding machines use machine vision applications to distinguish crops from weeds. As camera systems' sensor data is a valuable information source, this use case collects location-specific camera data to provide insights on the number of vegetables growing on the field, plants' growth status and best harvesting moment, weed prevalence, nutrient shortages and drought stress.

5.2.1. Domain model view

The domain model of the use case 'Added Value Weeding Data' is depicted in Fig. 4.

The main components in the domain model are the Steketee weeder, a field, tractor and harvest machine. Besides, there are a few stand-alone elements, which are sources of information like a soil sensor or an smartphone app.

In the Steketee weeder, images are processed to extract field information like crop size, growing stage and weed pressure. The machine was developed to perform intra-row weeding. To extract more management information from fields, some elements are added to the Steketee weeder component. With this information the development team can make a better Decision Support System (DSS) for farmers.

The field component contains elements, which have direct influence on the total yield. Weed and crop images are used in the weeder as inputs to determine where to actuate the weeding elements. In the harvest machine a yield monitor will be added to get feedback on the performance of crops. Crop yield is the result of all actions performed in the last year and years before. This information will be used to predict an optimum harvesting date and the expected yield.

The tractor component contains GPS information which is used to process information from the field in a spatial way. This gives the opportunity to perform site-specific measures on the field. All these elements will provide input for two Decision Support Systems (DSS)s, one DSS for the machine settings for the Steketee weeder and one DSS for field/crop management.

5.2.2. Business process hierarchy view

The business process hierarchy model of the use case 'Added Value Weeding Data' is depicted in Fig. 5.

This model shows how sensing data of weed pressure, crop growth and harvest is used for precision weeding and farm management. In the Physical Object Layer, relevant objects of this case are depicted: weeds and growing crops in the field. The weeding machine detects and removes weeds. Once the crops are ready for consumption, they are harvested with a harvest machine. The other layers include main farm processes on different time horizons that are needed in this case to sense and control the physical objects. The Production Control Layer includes operational processes to (physically) sense and control the weeds, crops and harvest. The weed pressure and crop growth are sensed by cameras and sensors in the weeding machine. The harvest machine includes a sensor for yield monitoring. This data is collected and analysed in the Operations Execution Layer. Aggregated data are used in the Management Information Layer to monitor crop growth and weed pressure. Next, the location-specific weeding need is calculated based on the weeding pressure monitoring and specific weeding tasks are planned. Alongside, also expected yields are predicted, based on crop growth monitoring, and translated into a detailed harvest planning. Both weeding and harvesting plans may result in location-specific farm operations. The farm control triggers the execution of these actions by sending the specific machine requirements and task definition to the Operations Execution Layer. In this layer the settings of weeding and harvesting machines are defined, weeding or harvesting tasks are scheduled and machine-readable task instructions (precision task maps) are sent to the Production Control Layer. The machine control process of the weeder or harvester, then, implements the specific machine settings and executes specific instructions.

5.2.3. IoT layer view

The IoT layer view of the use case 'Added Value Weeding Data' is depicted in Fig. 6.

The Application Layer includes an IoT dashboard for farmers and the machine vendor (Steketee). The machine vendor's dashboard is used to setup machines, to optimize and update settings and for maintenance purposes. The farmer's dashboard is used for operational sensing of weed pressure, crop growth and harvest, as well as for monitoring the execution of farming tasks. The AgLeader system supports usage of sensing data for farm management, including yield prediction, planning of farm operations and triggering execution.

Table 9
Use cases of which the architectures are presented in the paper (subset of Table 1).

Trial/sector	Case	Challenge	Focal Country	Chain Role	Adopter Type	Conventional/Organic
Fruit	3.4 Intelligent fruit logistics	Fresh fruit logistics through virtualization of fruit products by intelligent trays within a low-power long-range network infrastructure	GE	Logistics, Consumption	Majority	Both
Vegs	4.3 Added value weeding data	Boosting the value chain by harvesting weeding data of organic vegetables obtained by advanced visioning systems	NL, AT	Farming	Majority	Organic
Meat	5.1 Pig farm management	Optimise pig production management by interoperable on-farm sensors and slaughter house data	BE, NL	Farming, Processing, Consumption	Both	Both

The Service and Application Support Services Layer comprises web services and cloud data management capabilities that are used by dashboards and farm management systems in the Application Layer. The Network Layer includes different kinds of wired and wireless interfaces for connecting the weeding machine, tractor and harvester and for communication of machines and sensors with cloud systems. The Device Layer includes generic Device Capabilities for the sensors and actuators that are embedded in the weeding and harvesting machines and for soil sensors in the field. The terminals on the weeder and harvester are the main gateways that include local data processing and storage (e.g., of the images). The tractor terminal is used as a GPS gateway.

5.2.4. *Deployment view*

The deployment view of the use case ‘Added Value Weeding Data’ is depicted in Fig. 7.

The main deployed components are machines on the field, i.e., a tractor, IC weeder and harvest implement, and components to store, access, process, analyse and use the data generated, i.e., cloud platforms, local PC and advisory services. For the weeding task, the processing of camera images will be done offline in the IC weeder and the processed information, like weed pressure and crop size, is directly available after weeding. Other data that are generated during operation are first logged on the machine. After operation they are transferred to cloud storage. Besides weed/crop data a settings file will be logged as well. This will be used by Steketee to improve machine settings by learning from all types of weed/crop/field circumstances. Initially it will be used for generating advised settings.

A cloud platform will be the storage for all data from the IC weeder, harvester, Farm Management Information System (FMIS), machine services from the vendor Steketee and advisory services from Wageningen Research, e.g. about crop growth and yield prediction. All nodes are in a way connected to the cloud to have one central data storage point. Data from the cloud can be downloaded by Steketee or Wageningen Research, processed and put back on the cloud to be used by farmers. Data can be used directly in the IC weeder (e.g., settings update) or via the FMIS for farmers to be used for decision support.

5.2.5. *Information model view*

The information model view of the use case ‘Added Value Weeding Data’ is depicted in Fig. 8.

The information model above is closely related to application functionalities in the deployment view and the information exchanged between them. It distinguishes: (i) operational data flows, i.e. commands, responses and actuals; (ii) expert rules data, e.g. algorithms; (iii) advice data, such as alerts and predictions; and (iv) management data, including definitions, schedules, capabilities, and performance.

5.2.6. *Interoperability endpoints view*

The interoperability endpoints of this use case, as listed in Appendix A, are interfaces to access data on the weeding and harvest machines, a GPS interface on the tractor to get position data and Application Programming Interfaces (APIs) to communicate with the farm and machine vendor cloud platforms and the yield prediction service.

5.3. *Architecture use case ‘Pig Farm Management’*

This use case is related to the management of pig farms via optimal use of data throughout the chain. Through automated collection, processing and sharing of crucial information it aims at providing feedback to farmers via an easy-to-use interface (a pig business intelligence dashboard). In addition, the use case also fosters the reuse of collected data to enable information transfer to other relevant stakeholders (breeders, food processors, feed suppliers, veterinarians, etc.).

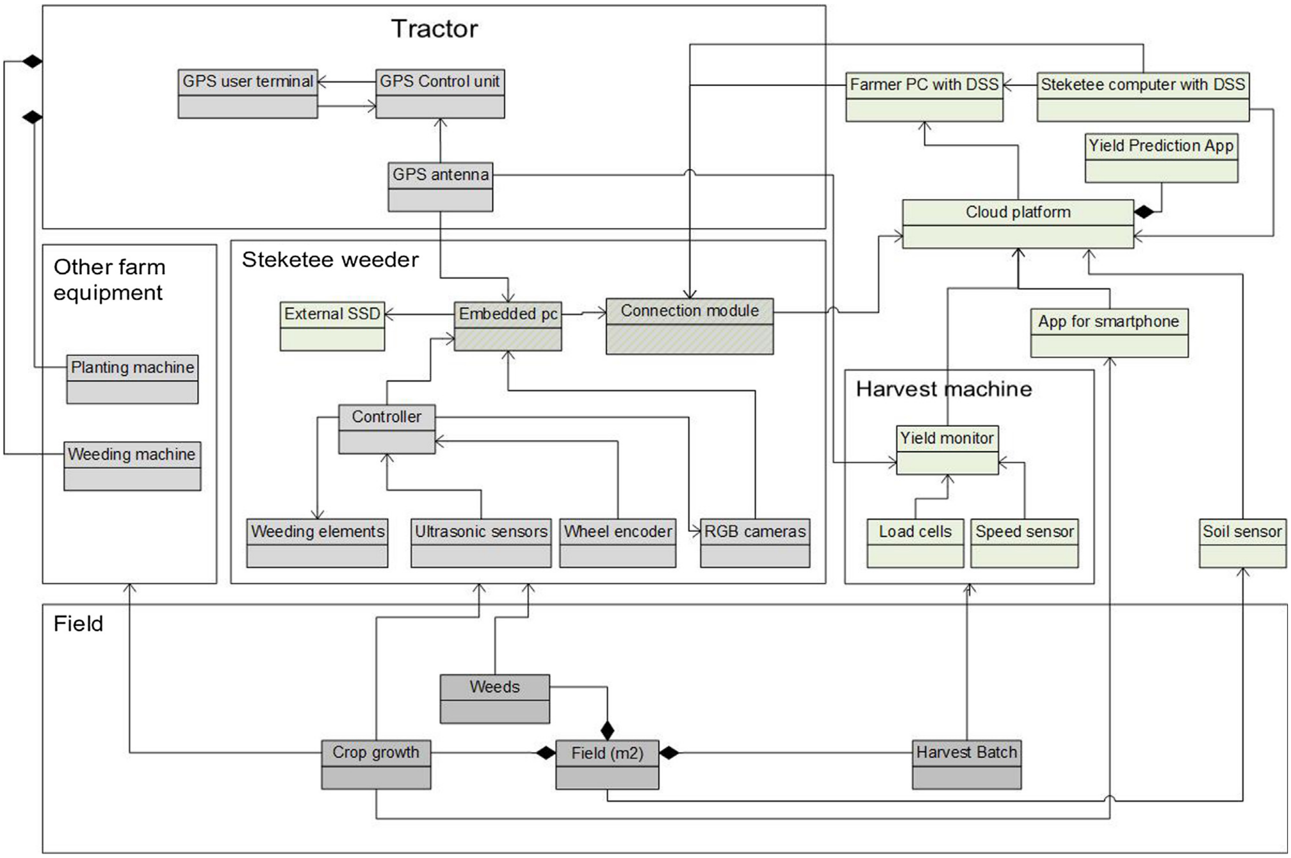


Fig. 4. Domain model view of the use case 'Added Value Weeding Data'.

5.3.1. Domain model view

The domain model for the use case 'Pig Farm Management' is depicted in Fig. 9.

In this use case, a farmer is monitoring and optimizing feed and water consumption, fattening performance and health parameters of both individual and group of pigs. Different IoT sensors, already deployed in the farm, will measure these parameters. Furthermore, a RFID reader can read the unique identification (tag) of each individual pig to

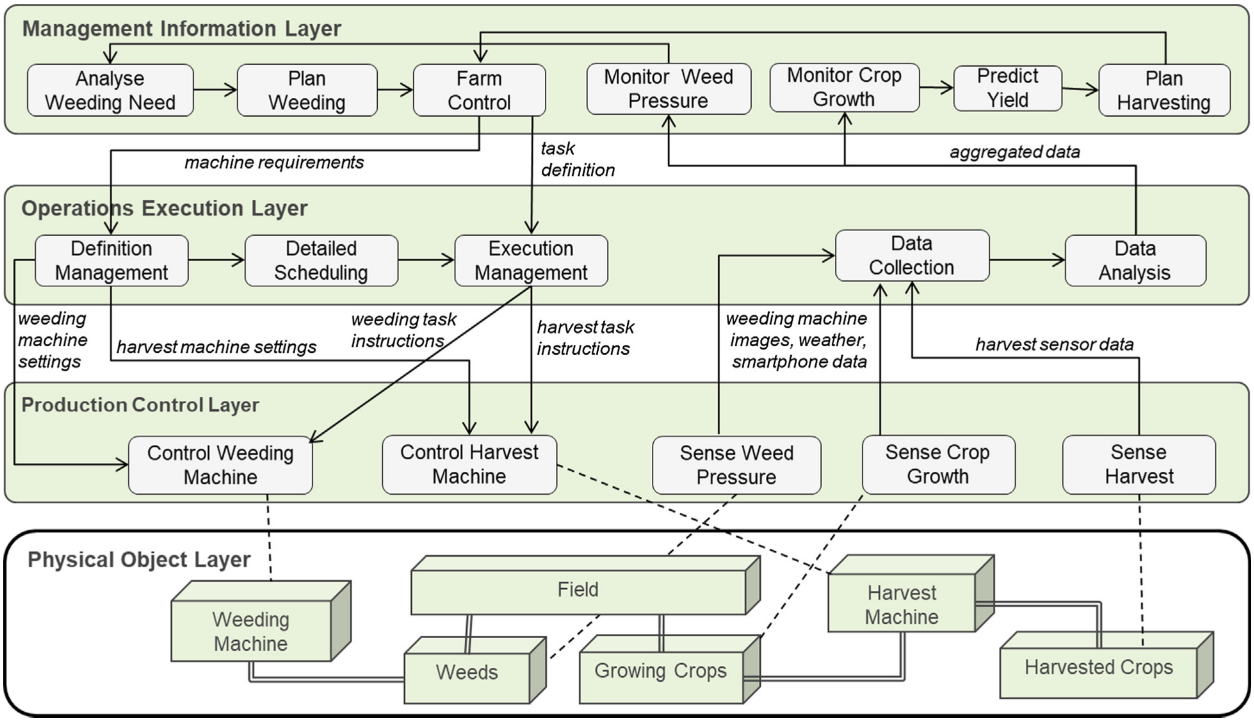


Fig. 5. Business process hierarchy view of the use case 'Added Value Weeding Data'.

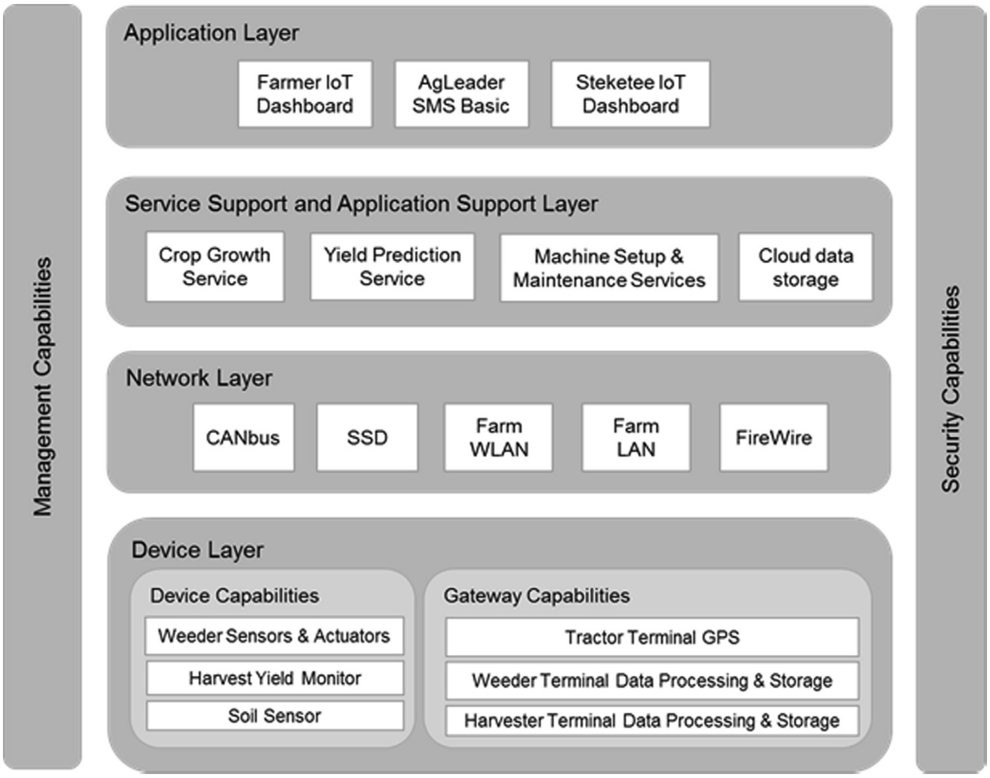


Fig. 6. IoT layer view of the use case ‘Added Value Weeding Data’.

track its movements in specific areas in which other sensors are active and, consequently, associate other collected information to the right pig. For example: if a flow of water is revealed and only one pig is near the tap, it is possible to deduce how much water was drunk by that specific pig. This possibility is assured by physically attaching unique RFID tags to the pigs’ ears. However, in some farms RFID tags are not used and/or some local laws do not allow it. In these circumstances, group level information is collected without storing the association to individual pigs. This group level monitoring is most commonly used in practice.

Other IoT sensors fostered in this use case are weight scales, water and feed consumption sensors that allow to measure the weight and the water and feed consumption for individual pigs. Moreover, temperature, relative humidity and light intensity sensors can be used to monitor climate and light in barns.

In addition to the data collected through IoT sensors, some other input data can be provided by farmers and/or experts about barns and pens through a dedicated web-based dashboard. An example of this information is called ‘boar taint’, which is an unpleasant odour that can occur in entire male carcasses and that can drastically negatively influence the final price of the produced meat. Boar taint is currently only detected by humans at the slaughter line.

All the described data is stored in a data storage system, which feeds dedicated algorithms suitable to extract eating behaviour, fattening performance and pig health figures. Such extracted inferences are shown to farmers through a dedicated web-based business intelligence dashboard.

5.3.2. *Business process hierarchy view*

The Business Process Hierarchy view for the use case ‘Pig Farm Management’ is depicted in Fig. 10.

The Business Process Hierarchy View comprises four layers: Physical Object Layer, Production Control Layer, Operations Execution Layer and Management Information Layer.

In the Physical Object Layer, the most relevant groups of information required by this use case are depicted: fattening performance, eating behaviour, health behaviour, that are sensed with IoT sensors and batch data. Farmer are especially interested in monitoring and optimizing of the eating behaviour, fattening performance and health of individual pigs. At the same time, experts are mainly interested in the detection of boar taint smell and slaughterhouse workers in data regarding carcass, boar taint, and transport.

The other layers include the main farm processes on different time horizons that are needed in this case to sense and control physical objects. The starting point is sensing of pig fattening and growth in the Production Control Layer based on IoT sensor data. This data is collected and analysed in the Operations Execution Layer. Subsequently, aggregated data is used in the Management Information Layer to monitor pig growth and fattening. Next, an overview of farm status and pig health is calculated based on pig growth and fattening monitoring, providing information for farmers, experts and slaughterhouse workers. Finally, this analysis is used to update the planning and to control its execution.

5.3.3. *IoT layer view*

The IoT layer viewpoint of this use case (depicted in Fig. 11) is structured as follows:

- *Application layer*: in this layer, a web-based dashboard is provided to visualize collected IoT data.
- *Service support and application support layer:* both generic support capabilities and specific support capabilities are provided within this use case. Specifically, a Business Intelligent Dashboard allows to analyse data and provide an overview of farm status and warnings in case status is not optimal, while the Fusion Engine Service allows to elaborate the data. In addition, there are common capabilities, which can be used by different IoT applications, such as Cloud Data Storage and Data & Context Broker.
- *Network layer:* in this layer both Networking Capabilities and Transport Capabilities are provided. Respectively, the first provide

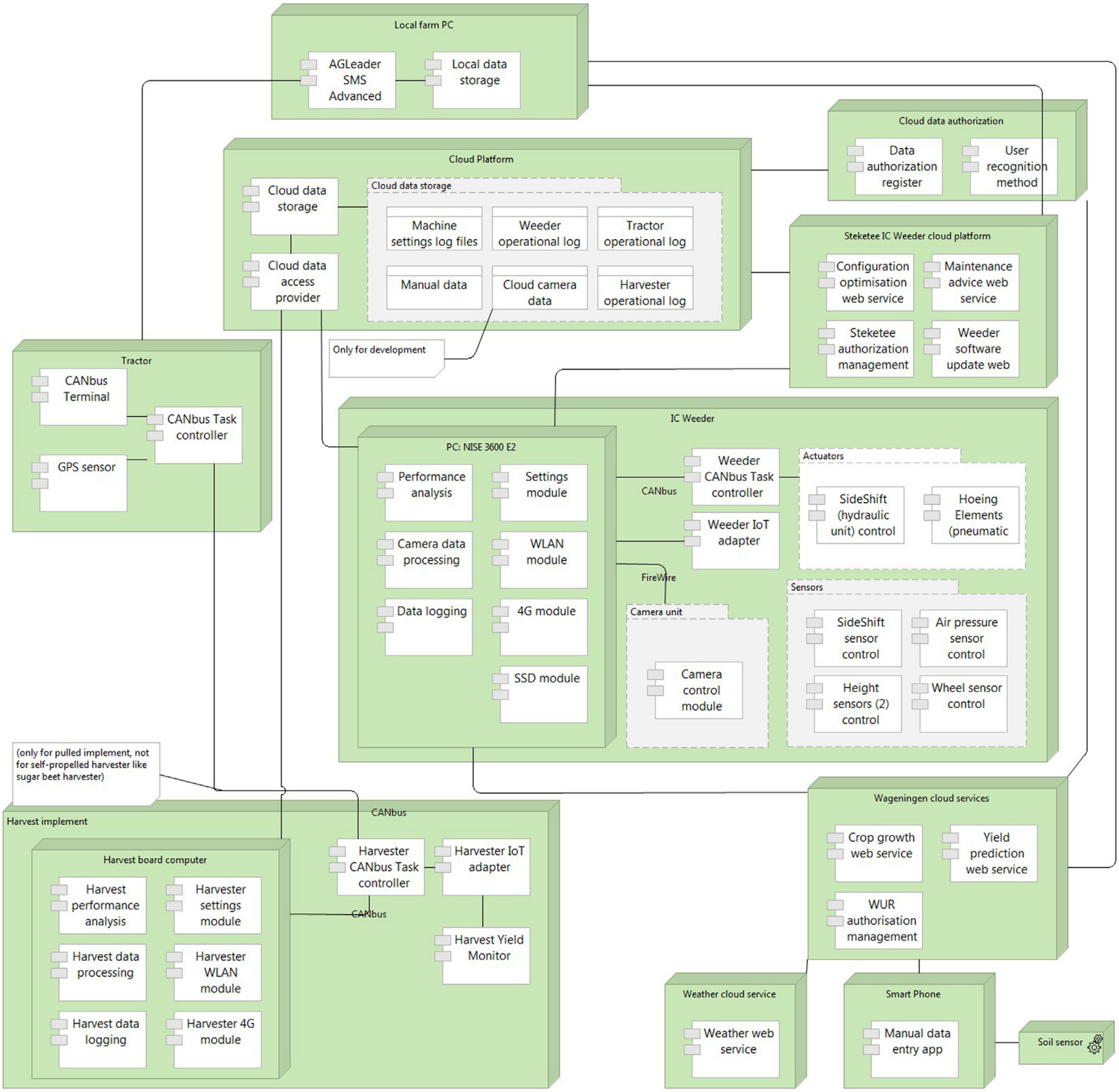


Fig. 7. Deployment view of the use case 'Added Value Weeding Data'.

relevant control functions of network connectivity while the second focuses on providing connectivity for the transport of IoT service and application specific data information. Network and transport connectivity are provided through specific technologies such as Wi-Fi, GPRS, XMPP-IoT, and NGSI (Cantera et al., 2018). The same happens for Mobility Management Capabilities that use specific protocols based on the used technology.

- *Device layer*: this UC includes general functions provided by existing devices and gateways. Specifically, Device Capabilities includes (a) sleeping and waking-up of devices to reduce energy consumption, (b) the possibility of sensors and actuators to gather and upload information directly or indirectly to the communication network, (c) the capacity of devices to construct networks in an ad-hoc networking based on a specific technology. Gateway Capabilities include supported devices that are connected through different kinds of wired or wireless technologies (multiple interfaces) and protocol conversion.
- *Management capabilities:* in this use case these capabilities are especially demanded for existing infrastructures and systems already installed in the farm.
- *Security capabilities:* this use case mainly leverages security functions of specific technologies that are already present in the farms.

5.3.4. Deployment view

The deployment view for the use case 'Pig Farm Management' is depicted in Fig. 12.

Components in this use case are deployed either locally (i.e., in a farm and or slaughterhouse) or remotely (i.e., in the cloud or in a self-hosted cloud server).

In a farm, five different physicals dedicated sensor platforms are deployed, namely a RFID reader platform, a water consumption platform, a feed sensor platform, a daily growth platform and a climate control platform. Each platform corresponds to a node composed by a dedicated stand-alone PC installed in a protected location in the farm

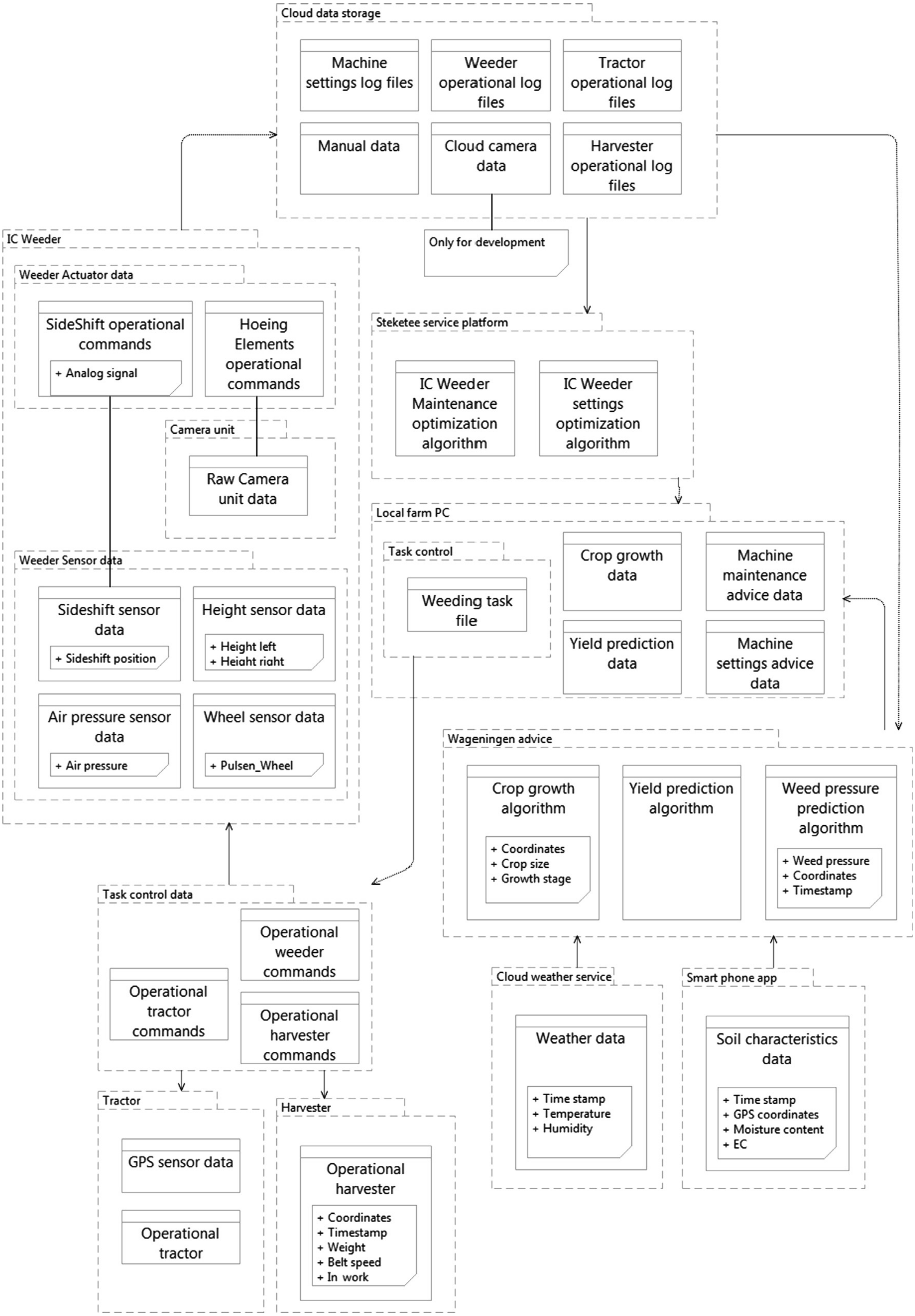


Fig. 8. Information model view of the use case ‘Added Value Weeding Data’.

that is connected to a local farm Local Area Network (LAN) used to inter-connect these nodes to a farm server.

The Farm Server is a general-purpose ruggedized x86-64 PC running Windows 10 IoT Enterprise, which hosts five 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 Virtual Private Network (VPN), which allows secure communications towards a

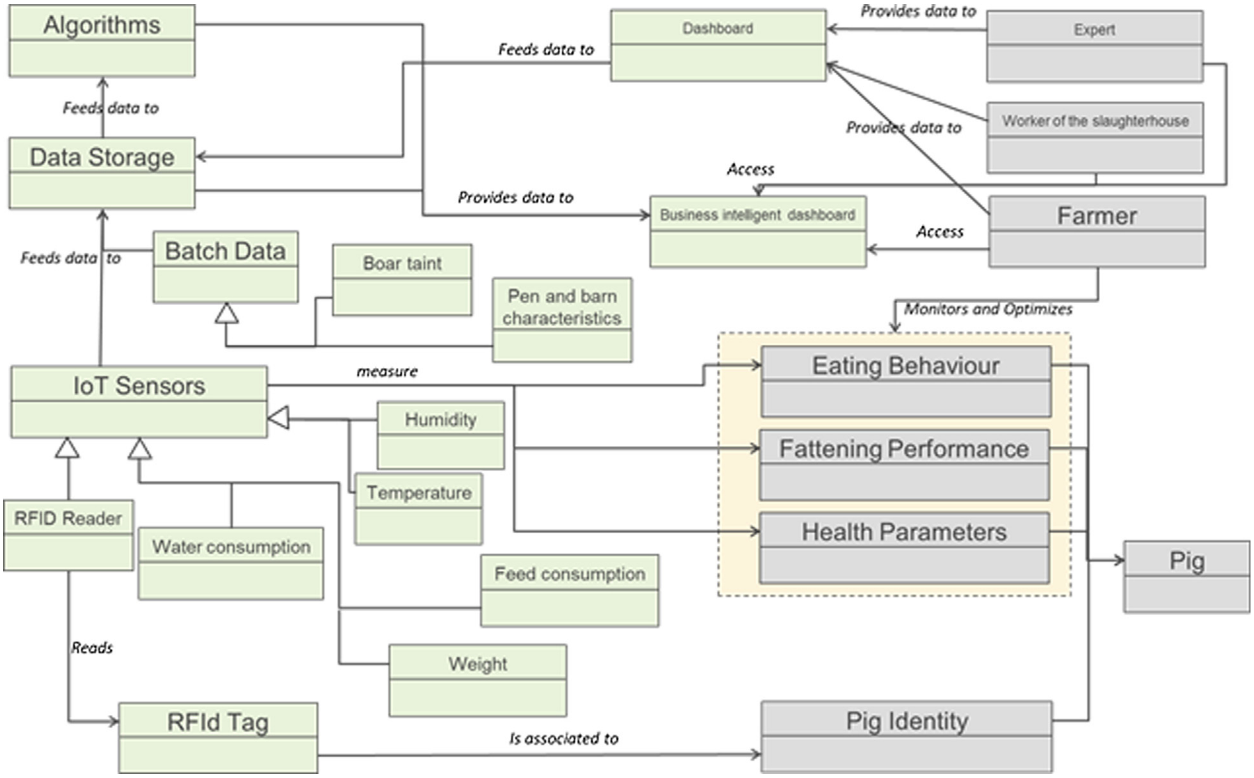


Fig. 9. Domain Model of the use case ‘Pig Farm Management’.

private Cloud Service platform. The cloud platform runs: (a) a Cloud Data Storage service and a Data & Context Broker, which receive data via XMPP-IoT or NGSI from farm servers; (b) a Business Intelligent Dashboard, accessible via HyperText Transfer Protocol over Secure Socket Layer (HTTPS); and (c) a Fusion Engine Service running algorithms.

5.3.5. Information model view

The Information model for the use case ‘Pig Farm Management’ is depicted in Fig. 13.

On the left side, there are sensor data entities that have at least a unique identifier. Those data are classified as ‘raw sensor data’ and include humidity, temperature, water consumption, feed consumption and presence sensor data. The entity ‘drinking event’, which is derived from water consumption and/or presence, also has a unique identifier.

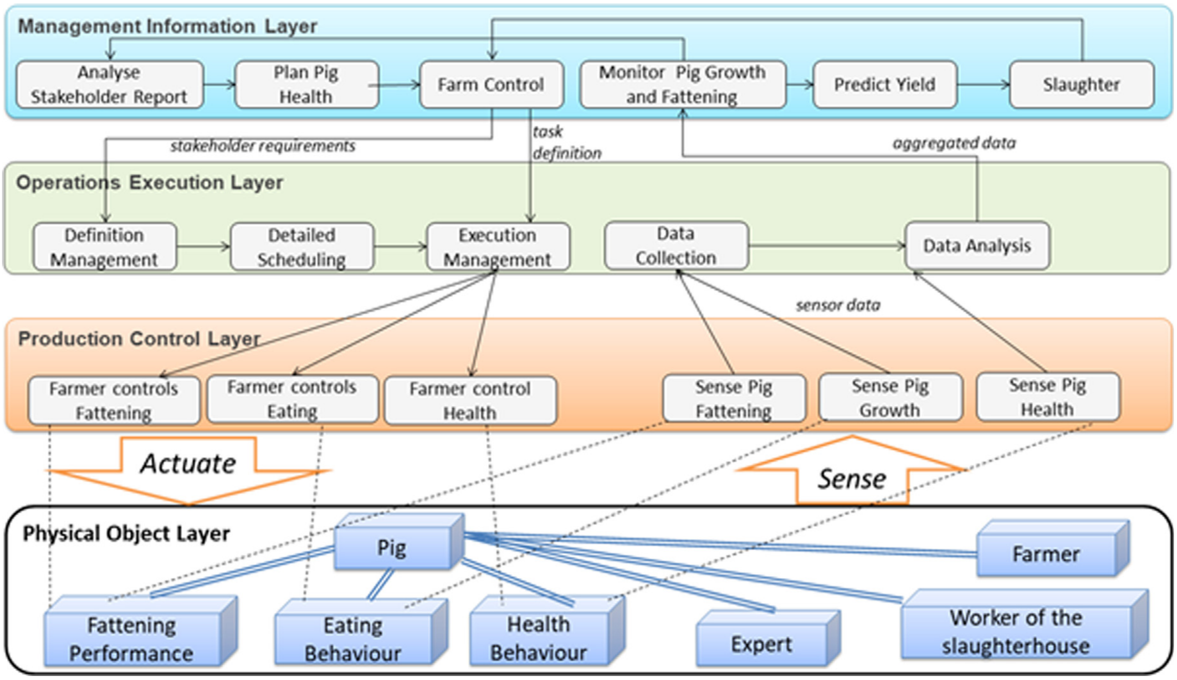


Fig. 10. Business Process Hierarchy view of the use case ‘Pig Farm Management’.

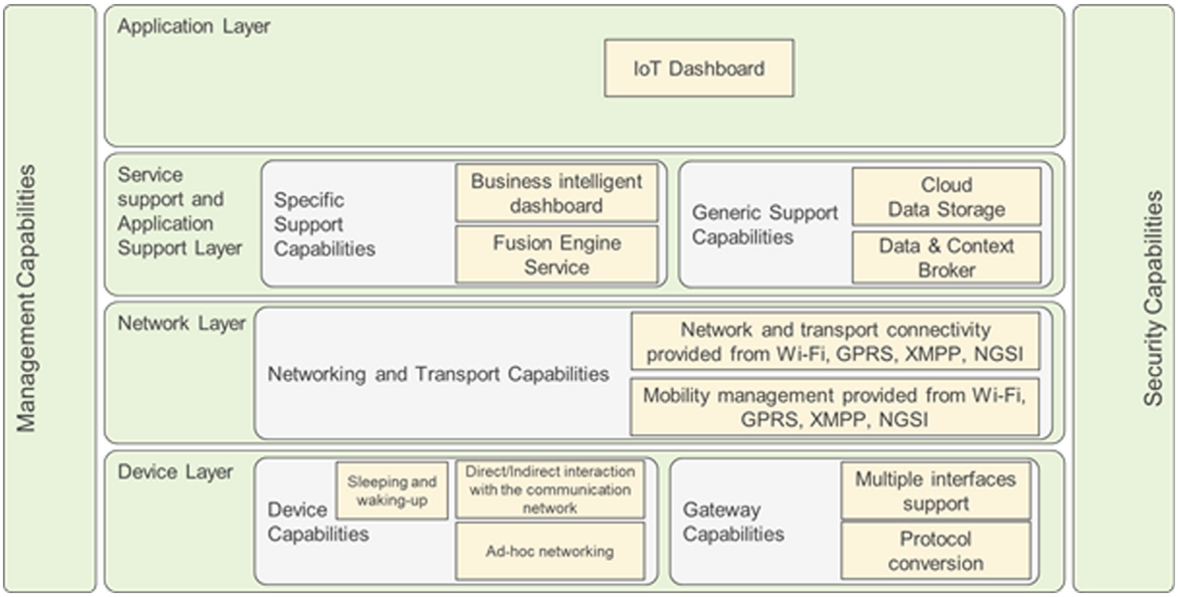

Management Capabilities
Application Layer
IoT Dashboard
Service support and Application Support Layer
Specific Support Capabilities
Business intelligent dashboard
Fusion Engine Service
Generic Support Capabilities
Cloud Data Storage
Data & Context Broker
Network Layer
Networking and Transport Capabilities
Network and transport connectivity provided from Wi-Fi, GPRS, XMPP, NGSI
Mobility management provided from Wi-Fi, GPRS, XMPP, NGSI
Device Layer
Device Capabilities
Sleeping and waking-up
Direct/Indirect interaction with the communication network
Ad-hoc networking
Gateway Capabilities
Multiple interfaces support
Protocol conversion
Security Capabilities

Fig. 11. IoT layer view of the use case 'Pig Farm Management'.

The attribute 'duration' is used for describing the period of event and 'timestamp_start' shows the time when an event is started. The same values are also collected for the 'eating event' that it is derived from feed consumption and/or presence.

An RFID tag, uniquely identified by a tag_id, is associated to each pig. Every drinking event and eating event are in correlation with an RFID tag, so that it is possible to link each event to a pig. Furthermore, daily growth, health and genetics data are batch data linked with the pig. The climate entity is a batch data derived from humidity and temperature entities, linked to a specific barn or pen.

Other batch data, useful for the system, are boar taint, carcass parameters and transport & waiting time that are linked with slaughter. Finally, pen and barn characteristic and survey are batch data that are linked with farm and uniquely identified by a unique identifier.

5.3.6. *Interoperability endpoints view*

The interoperability endpoints of this use case, as defined in Appendix A, are interfaces to access data collected and provided by the present use case. These data include water consumption, feed consumption, daily growth and climate control. Other sensor interfaces provide data acquired through infrastructures already present in farms that can be accessed through a farm server interface. The RFID reader and tag allow to uniquely identify each pig and to monitor its behaviour in the pens. The slaughterhouse database interface and the slaughterhouse server interface provide access to all data gathered in the slaughterhouse. The cloud service database interface and the data &

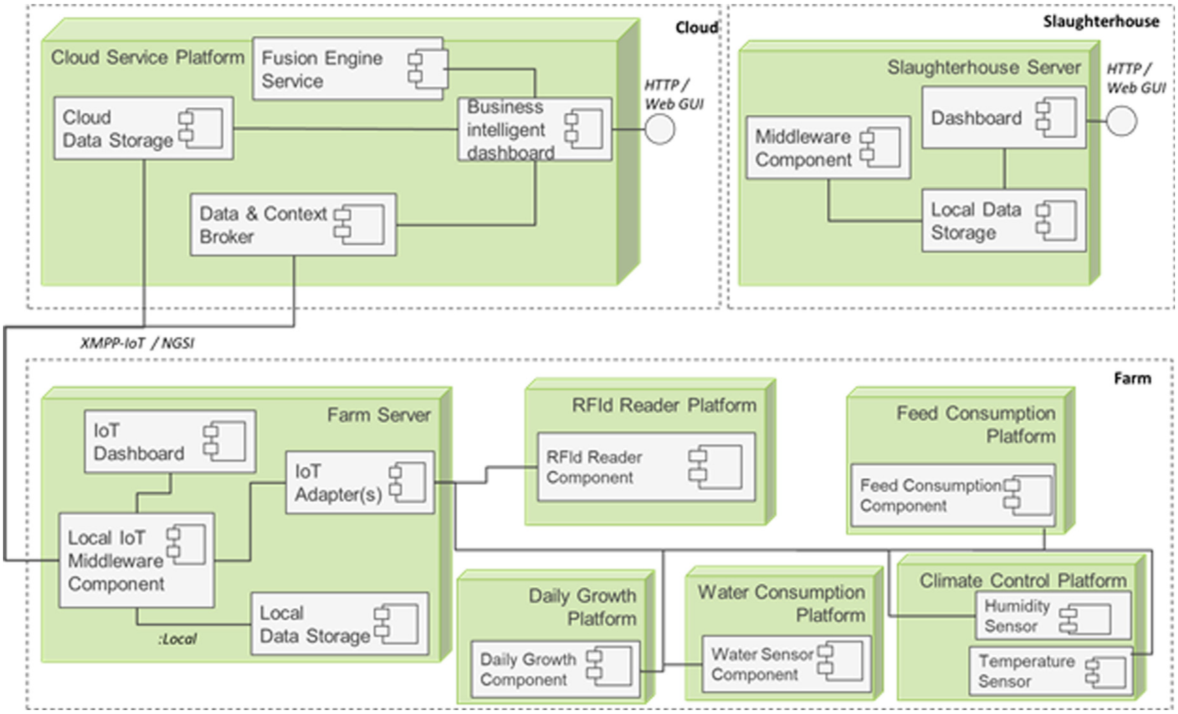


Fig. 12. Deployment view of the use case 'Pig Farm Management'.

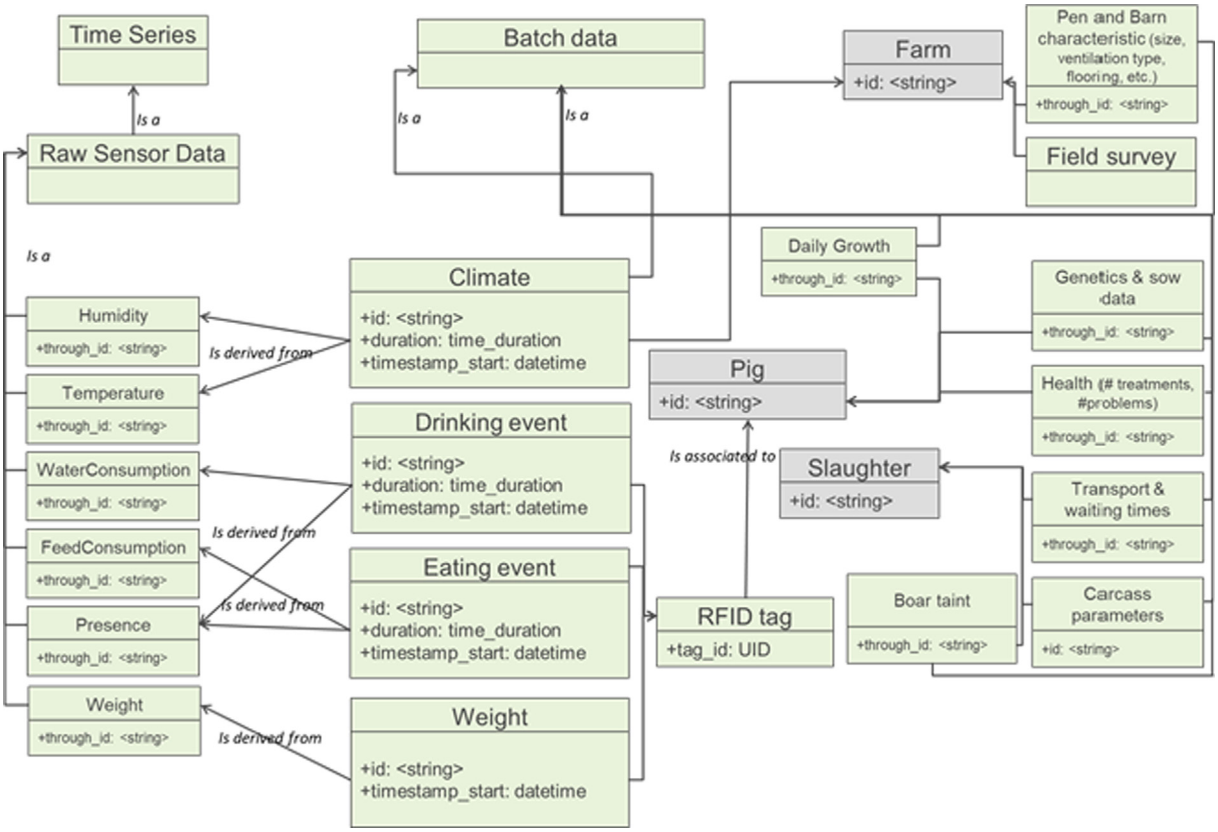


Fig. 13. Information model of the use case ‘Pig Farm Management’.

context broker expose all information collected in this use case.

5.4. Architecture use case ‘Intelligent Fruit Logistics’

The strategic objective of this use case is to connect IoT-enabled Returnable Trade Items (RTIs) with smart applications, to open a new dimension of added value services in multi-stakeholder fruit and vegetable supply networks. The main stakeholder and use case leader is Euro Pool System, a returnable packaging pool provider and European market leader for returnable packaging in the fruit and vegetable sector. Existing Euro Pool System RTIs are equipped with IoT devices to enable the RTIs to transmit information during their usage by Euro Pool System’s customers.

The key stakeholders are located in Germany and the Netherlands. GS1 Germany supports the use case realisation as expert regarding identification and Electronic Product Code Information Services (EPCIS). From an IoT perspective, semiconductor manufacturer NXP facilitates the experimentation with advanced IoT technology. ATB Bremen as research and innovation partner develops smart application for different MVPs, while Mieloo & Alexander is customising diverse ICT solution components, bridging gaps between diverse systems when providing their expertise as system integrator.

5.4.1. Domain model view

The main stakeholder is operating a pool of around 140 million RTIs that are used by its customers to transport produce (i.e. fruit and vegetables), usually from farm to retail. Each RTI is uniquely identified by a Global Returnable Asset Identifier (GRAI) number, which is represented by a decimal number, a barcode and a QR-code on the RTI. By equipping an amount of RTIs with IoT devices, the objective is to turn passive boxes into active things. This shall facilitate RTI tracking and tracing as well as an optimal usage and free circulation of RTIs. This shall also help to guarantee a proper operation of the overall open RTI pool. It is foreseen to use a long range low bandwidth communication protocol to maximize life-time of the IoT devices and possibly to enable determination of the position. However, initial experiments with this technology came to the conclusion that the accuracy of the geo-localisation is currently not yet precise enough for the requirements of this use case. Therefore, also a GPS chip is used to increase the accuracy of the RTI geo-localisation.

This real-time monitoring of RTIs is integrated with existing systems for monitoring and planning their usage as well as combined with the classical RTI identification technologies. Furthermore, the current ERP and External Places APIs can be used to identify locations and to manage Global Location Numbers (GLN)s in relation to the RTI users. In a later phase, it could be foreseen to offer an EPCIS interface and additional customer services to enable RTI users to access event data that is relevant for their specific business processes.

The domain model of the use case ‘Intelligent Fruit Logistics’ is depicted in Fig. 14.

5.4.2. Business process hierarchy view

The main purpose of the solution is to facilitate a free circulation of RTIs from the pool as well as to guarantee a proper operation of the pool and avoid misuse. Therefore, the management information layer is representing the key supportive activities when the RTIs are circulating from the depot to users and backwards to the depots for an appropriate cleaning. An example of a related customer chain are farmers providing produce to traders, which are shipping produce to retailers. Those retailers are distributing the produce within the RTIs to their points of sale (e.g. supermarket) and collect empty RTIs for being able to return them to a depot.

Therefore, on the operations execution layer, the pool operator is taking care for a timely processing and shipment of customer orders. At the same time, data is gathered to facilitate the operational preparation and load balancing in the different depots.

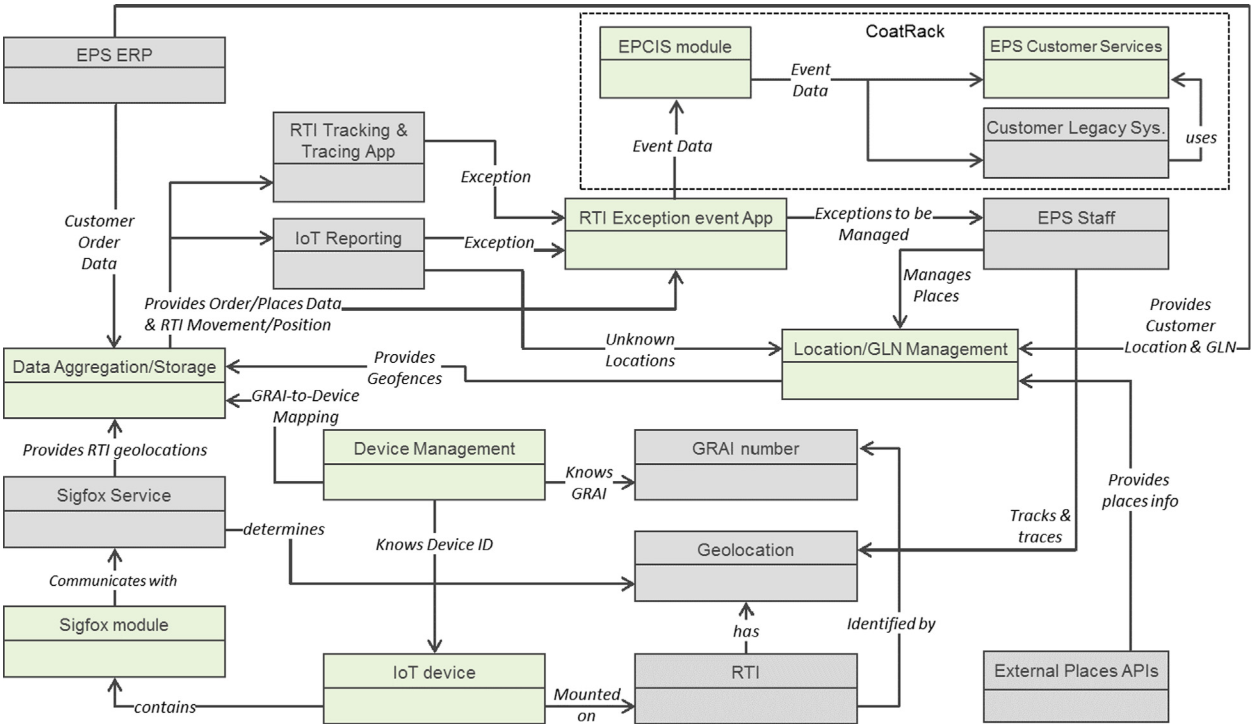


Fig. 14. Domain model view of the use case 'Intelligent Fruit Logistics'.

The specific tasks are initiated on the production control layer, to realise the cleaning process itself as well as to organize the shipment in between depots as well as to the customers, while this includes different interactions with customers as well as logistic service providers. At the same time, the equipment is operated for gathering all the required data.

The physical object layer is presenting the circular process in relation to the RTI usage (i.e. from depot storage, over usage up to the reception and cleaning of used RTI in the depots) as well as potential exceptional situations that might cause issues and need to be handled.

The business process hierarchy model of the use case 'Intelligent Fruit Logistics' is presented in Fig. 15.

5.4.3. IoT layer view

In the device layer, a prototype of the IoT device mounted on the RTI will include a module for GPS positioning. In a later version of the device, additional capabilities/sensors could possibly be added. A Sigfox module will be included in the IoT device prototype, enabling it to directly interact with the communication network.

At a later point in time, it will be decided which sensors are finally required or how to possibly modularize the IoT device, enabling a later selection of which data is required. This is also the base for making a cost-benefit analysis as well as to balance battery capacity with required energy for operational usage.

On the network layer, the Sigfox communication network will be used for communication between the IoT device and the Sigfox cloud, while the Data Aggregation/Storage module in the EPS cloud will provide a call-back interface to get RTI position information from Sigfox via the Internet.

The service support and application support layer comprises external services to get data about positions/places and EPS services to provide data to customers. The application layer comprises all applications to be used by the end-users.

The IoT layer view of this use case is presented in Fig. 16.

5.4.4. Deployment view

The IoT devices are mounted on RTI's that will freely circulate between depots and RTI end-users. Each RTI is a unique and autonomous object. They will communicate via a long range low bandwidth communication protocol. At the current moment, Sigfox was selected, due to its regional availability. The Sigfox service shall be used via a subscribe/call-back mechanism to regularly gather data sent by RTIs. The IoT Platform will be installed on an external application server to gather data as well as to facilitate the analysis of acquired position data, providing customized reports. The specific interfaces to the RTI exception event app need to be agreed, while it will operate its own database to facilitate the analysis. The location and exception management will be realised on an application server, also facilitating the integrated usage of mobile apps, usually running on Android devices.

In a later phase, it could be foreseen to offer an EPCIS interface and additional services to enable end-users to access event data that is relevant for a specific purpose in their business processes. The CoatRack module, developed in the scope of IoF2020, could be used as infrastructure to make these interfaces/services available to customers, facilitating access control.

The deployment model of the use case 'Intelligent Fruit Logistics' is depicted in Fig. 17.

5.4.5. Information model view

Main elements of the information model are the unique identification of orders, pallets and RTIs as well as the geographical information that is required to identify the position of RTIs. The position is represented by geo-locations, geo-fences as well as places, while the latter represents the position in relation to a stakeholder that is currently using, storing or handling the RTI (e.g., depot, farm, warehouse, retail location). Relevant for the application will also be information about places outside of RTI handlers' sites, where RTIs sometimes stay for a longer time. This is especially important when aiming at the provision of added-value services for being able to authorise the provision of information.

The information model of the use case 'Intelligent Fruit Logistics' is presented in Fig. 18.

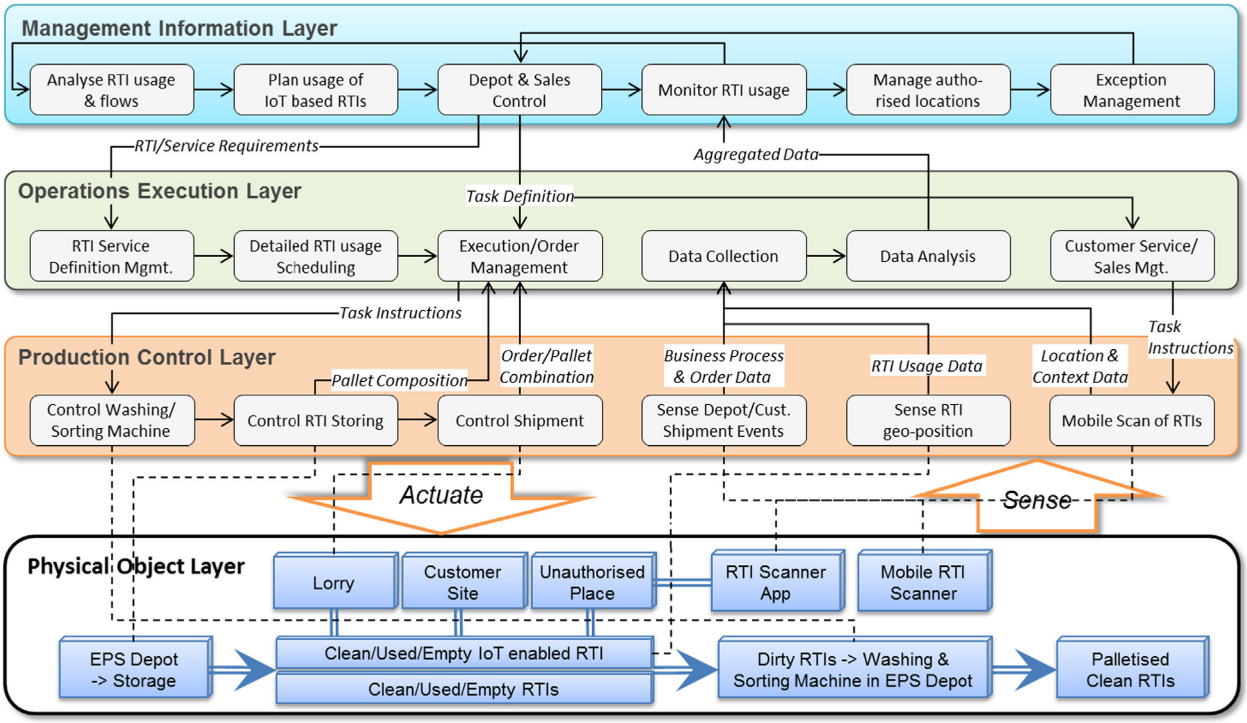


Fig. 15. Business process hierarchy view of the use case 'Intelligent Fruit Logistics'.

5.4.6. Interoperability endpoints view

In Appendix A the interoperability endpoints of the Use case 'Intelligent Fruit Logistics' are listed. They are specifically related to the RTI and Sigfox usage as well as in relation to the event messaging with the IoT platform and related systems to initiate actions in the business processes.

6. Discussion

This paper has developed an architecture framework for modelling IoT-based systems in the agriculture and food domain. It was found that related existing frameworks either lack viewpoints for modelling the IoT dimension or have a too technical focus excluding business viewpoints. For this reason, the framework combined viewpoints from different frameworks that together: (i) support a seamless translation of business design to detailed information engineering models, (ii) visualize how objects are sensed and controlled by IoT technologies, (iii) support the interoperability and reuse of system components and (iv) provide insight in the essence of the use case systems in a consistent, concise but also simple way, not overcharging the use case owners. Subsequently a guideline, i.e. a general method, for the application of these viewpoints has been introduced.

By using the guideline, the framework is applied to in total 19 use cases that reflect the diversity of the agri-food domain, including different sub sectors, conventional and organic farming, early adopters and early majority farmers, and different supply chain roles, including

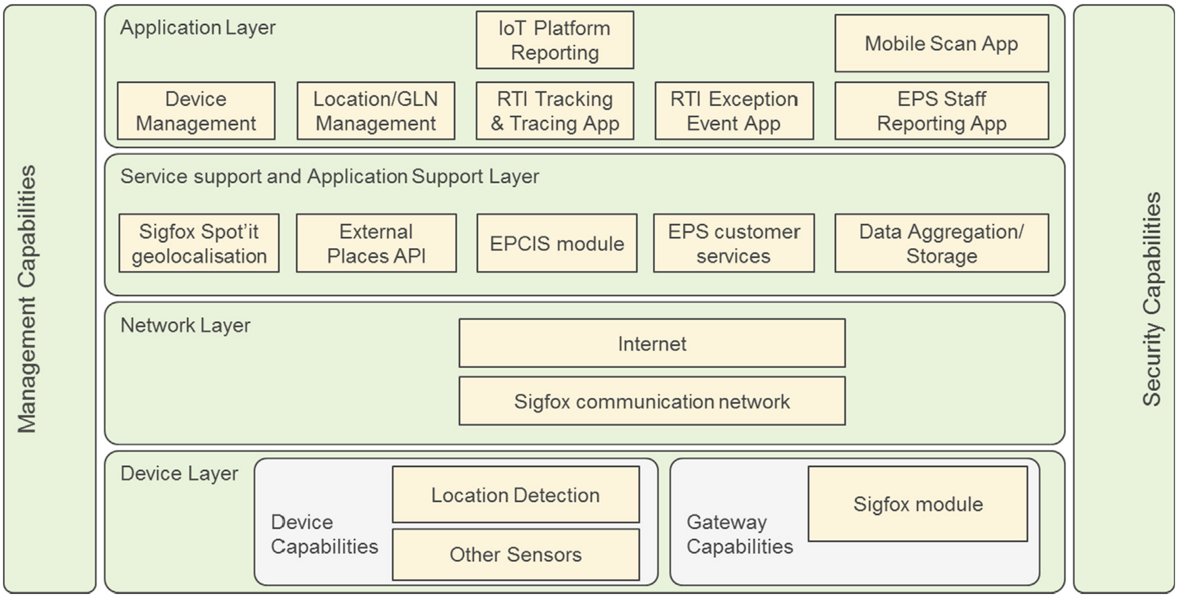


Fig. 16. IoT layer view of the use case 'Intelligent Fruit Logistics'.

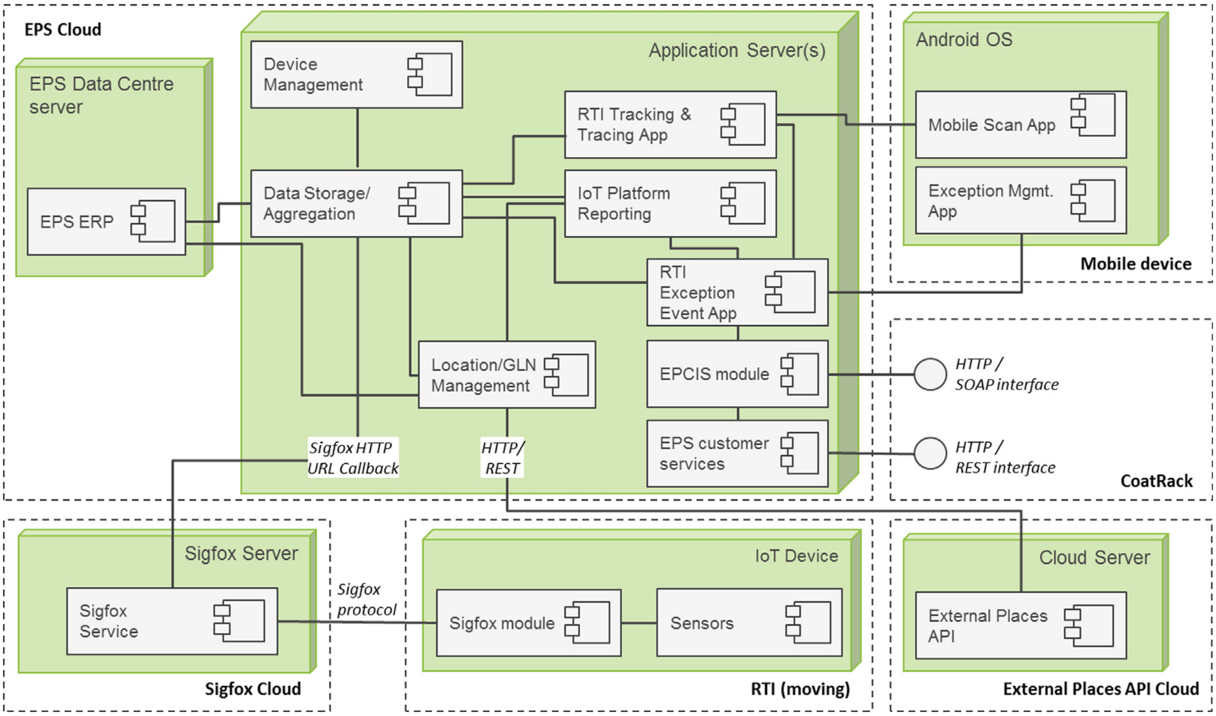


Fig. 17. Deployment view of the use case 'Intelligent Fruit Logistics'.

farming, processing, logistics and consumption. It turned out that a major advantage of the framework is that it helps to model, in a timely, punctual and coherent way, the architecture of IoT-based systems of this diverse set of use cases. Moreover, it serves as a common language for aligning the system architectures and enabling the reuse of architectural knowledge among multiple autonomous IoT systems in agriculture and food.

A main theoretical novelty of the framework is that it combines viewpoints of several generic software architecture frameworks and specific IoT frameworks in order to support the modelling of IoT-based systems. Such systems not only include technical capabilities for sensing and actuation, but also support the usage of operational sensing data in the planning and control of business processes. Existing generic frameworks however miss viewpoints for modelling IoT capabilities, while IoT-specific frameworks focus on technical IoT aspects, but lack especially business viewpoints. Our framework combines the strengths of both worlds since it includes both business and technical viewpoints.

Furthermore, the framework goes beyond the selection of viewpoints from existing frameworks, but also adapted viewpoints for the making it appropriate for the purpose of the research. At this, we found

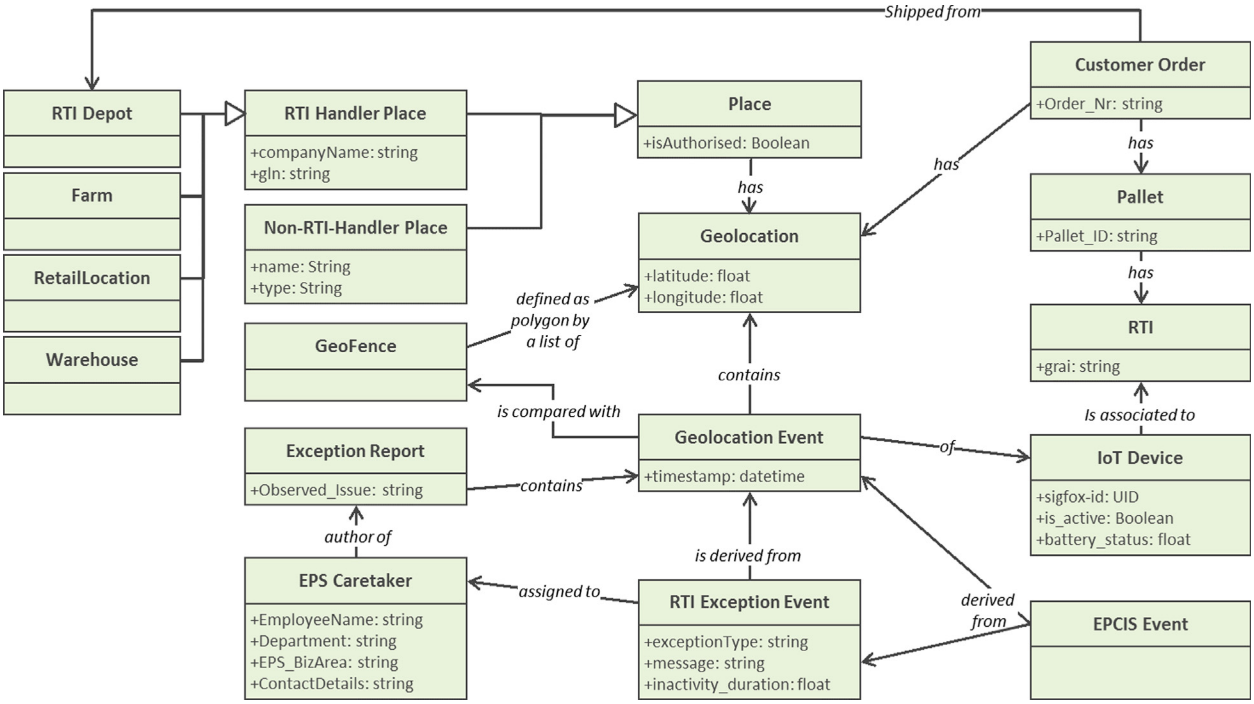


Fig. 18. Information model view of the use case 'Intelligent Fruit Logistics'.

the industrial ISA95 reference architecture very beneficial. Especially the classification of control layers based on time horizon is not only important for industrial automation, but also for IoT-based farming systems. However, this perspective is lacking in existing IoT architecture frameworks. For this reason, we translated the industrial control layers (i.e. ERP, MES, SCADA and PLC) into our business process hierarchy viewpoint. Moreover, we have added a layer for physical objects and transformations, as well as the object control relations (sense or actuate) with the production control layer, in order to explicitly address the 'things'-dimension of IoT. As a result of these changes, our framework is appropriate for modelling complete IoT-based systems, including monitoring, decision making, planning and execution control.

Another important scientific contribution is the application of our framework to the agri-food domain in a large-scale multiple-case study. As such, the paper provides unique and in-depth descriptive knowledge of a broad and diverse set of IoT-based systems for smart farming and food supply chains. Also other sectors can gain from our framework and learn from how it is applied in agri-food. They can use it as a basis to develop a reference architecture for their own domain or they can adopt specific viewpoints to complement existing sector-specific architectures. For example, to the best of our knowledge many IoT reference architectures for Smart Cities focus on technical aspects and could use our framework to strengthen its business views.

Considering the current literature on architecture viewpoints the question could raise whether other viewpoints could have been introduced and are needed in the framework. Obviously, the framework has been derived from a real practical setting in which both the selection and the specific implementation of viewpoints have been shaped by practical constraints. An important requirement was that the viewpoints must provide insight in the essence of use case systems in a consistent, concise but also simple way, not overcharging the use case owners. For this reason a limited set of key viewpoints was identified that provides just enough representation power to model the essence of the use case systems. As such, the research is an illustration of the trade-off between complexity of an architecture framework and its usefulness for a particular practical objective. A drawback of this approach is that the framework might be not sufficient for other purposes. For example, we expect that for software programming the framework is still too generic and it is also lacks viewpoints for modelling the (commercial) business model of an IoT system.

In our future work, we intend to apply the framework to other agri-food cases, to further enrich it with more detailed architectural modelling knowledge and to extend its scope. More specifically, three challenges are addressed.

First, the present research has designed and applied an architecture framework and consequently it is limited to architecture viewpoints. It has not yet developed reference architectures, i.e. concrete architectures that formalise recommended practices (i.e. standardized models) for a certain domain. Such a reference architecture can be an application of our framework, not for a particular use case, but for a class of IoT-based systems. The broad application of the framework provides an excellent basis for such a reference architecture, but future research is needed to abstract reference architectures from the use case-specific architectures.

Second, as stated before, our framework focuses on a limited set of key viewpoints for the purpose of the present research. We expect that additional viewpoints will be needed to support the complete software lifecycle. In case other viewpoints are needed, they can be borrowed from existing architecture frameworks or novel viewpoints can be added. Hereby it should be noted that a new viewpoint should be consistently integrated in the current architecture framework to ensure that it will remain a coherent set of viewpoints for the defined scope and goals, that is, an architecture modelling approach for the agri-food domain.

Finally, in the introduction it was argued that IoT-based systems should function as interoperable but autonomous nodes within a system of systems. We envision that our framework could serve as a common language for aligning the individual architectures and enabling the reuse of architectural knowledge within a system of IoT-based systems. However, we did not yet explicitly address the system of systems approach in our research. It should be further researched how architecture frameworks and reference architectures can effectively enhance a system of IoT-based systems in the agricultural and food domain.

7. Conclusion

This paper has provided an architecture framework consisting of a coherent set of viewpoints for modelling IoT-based systems in the agri-food domain. In addition, it developed a guideline, i.e. a general method, for applying the architecture framework. The framework has been applied and validated by 19 use cases within the large-scale industrial context of the European IoF2020 project. The framework has been directly used by practitioners while it has elaborated on the current state of the art on architecture modelling within the context of IoT. From this perspective both the architecture framework and the experiences from this large scale project in architecting IoT-based systems can be considered as unique. The architecture framework with its set of viewpoints aims to support communication among stakeholders, to guide design decisions and to evaluate designed architectures. Within all cases considered in our study, we could indeed observe a practical need and necessity of a proper architecture framework. The set of architecture viewpoints, together with the method, directly supported the development of IoT systems for the corresponding case studies. Besides of the end-results, that is, the architecture views, we can state that also the process of documenting the architecture helped to understand the systems, to make explicit and discuss the design decisions, to evaluate the fitness of the architectures for stakeholder needs, and subsequently to guide the development of the IoT systems.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 731884. The authors greatly acknowledge the valuable contributions from colleagues of the IoF2020 consortium.

Appendix A. Interoperability endpoint views of the use cases

Interoperability endpoints of the use case 'Added Value Weeding Data':

Interface name	Exposed by	Protocol	Notes
Weeding Machine interface	Steketee Weeding machine	WLAN, 3G/4G and manual (SSD)	Interface to get image data of the weeding machine and to control machine setting and task instructions
Harvest Machine interface	Harvest machine	csv	Interface to get harvest data of the harvesting machine
GPS	Tractor	NMEA	GPS interface on the tractor to get position data

Farm Cloud Data interface	Farm cloud data platform	To be specified	API to access farm cloud platform for camera and sensor data, machine settings and task instructions
Weeder Cloud Data interface	Steketee Cloud data platform	To be specified	API to access weeder cloud for maintenance and optimisation of machine settings
Yield prediction interface	Yield prediction service WPR	To be specified	API to access app for yield prediction

Interoperability endpoints of the use case 'Pig Farm Management'

Interface name	Exposed by	Protocol	Notes
Water Consumption Sensor Interface	Water Consumption Sensor	GHM Meettechniek Flowmeter MID1-008AP001E with LABO-MID1-008UNS	It will probably be a wired sensor
Feed Consumption Sensor Interface	Feed Consumption Sensor	to be specified	It will probably be a wired sensor
Daily Growth Sensor Interface	Daily Growth Sensor	to be specified	It will probably be a wired sensor
Climate Control Interface	Climate Control	Monnit Wireless Temperature & Humidity Sensor - Commercial Coin Cell Powered	It will probably be a wireless sensor
Other sensors interface	to be specified	to be specified	It will probably be a wired sensor
RFId Reader Interface	RFId Reader	LLRP (over IP, local)	Global EPC Standard
RFID Tag interface	RFID Tag	UHF or HF	(These Standards RFID radio protocols)
Slaughterhouse DB interface	Slaughterhouse Local Data Storage	Standard SQL End-point (over IP)	Only available on local network.
Slaughterhouse Server interface	Slaughterhouse Middleware Component	XMPP/Virtus or MQTT/Linksmart	Application-level profiles to be further specified during developments
Farm Server interface	Farm Server	XMPP/Virtus or MQTT/Linksmart + ICE 2	Application-level profiles to be further specified during developments
Cloud Service DB Interface	Cloud Data Storage	XMPP/Virtus or HTTP/Virtus	Application-level profiles to be further specified during developments
Data & Context Broker	Cloud Service Platform	NGSI	Connection between Farm Platforms and Porphyrio Platform through FIWARÉs context Broker

Interoperability endpoints of the use case 'Intelligent Fruit Logistics':

Interface name	Exposed by	Protocol	Notes
Sigfox callback interface	Sigfox	Sigfox HTTP URL callback	Service provided by Sigfox to get device information via a callback URL
Order/Places and RTI position data	Data Aggregation/Storage Module	to be specified	Aggregation of all RTI positions and RTI scans as well as geofences to make them available for diverse systems
IoT Platform	IoT Event System	to be specified	Interface to get exception information (location, place, # of RTIs, customer, flows and other??)
RTI_TT	RTI tracking & tracing Application	To be agreed	Interface to provide information about discovered RTIs
Places and Geofences	Location/GLN Management	EPCIS messages via HTTP	Provision of all locations of current and potential customers as well as other places
RTI Device ID	Device Management	To be agreed	Providing the mapping of IoT device ID to the GRAI of the related RTI
EPCIS events (t.b.d.)	EPCIS database?	To be agreed	EPCIS events about RTI movements/exceptions
RTI GRAI scans	Mobile Barcode Reader	To be agreed	Scan of GRAIs that were acquired in combination with an order, shipment and/or customer ID
Customer Interface (t.b.d.)	EPS customer service	To be agreed	Provide information about RTI movements/exceptions to EPS customers

References

AIOTI, 2015. Smart Farming and Food Safety Internet of Things Applications – Challenges for Large Scale Implementations. Alliance of IoT Innovation WG06, pp. 49. https://aioti.eu/wp-content/uploads/2017/03/AIOTIWG06Report2015-Farming-and-Food-Safety.pdf.

AIOTI, 2018. High Level Architecture (HLA) Release 2.1. Alliance for Internet of Things Innovation WG03, pp. 20. https://aioti.eu/wp-content/uploads/2017/03/AIOTI-WG3-IoT-High-Level-Architecture-Release_2_1.pdf.

Atzori, L., Iera, A., Morabito, G., 2010. The Internet of Things: a survey. Comput. Netw. 54 (15), 2787–2805.

Benbasat, I., Goldstein, D.K., Mead, M., 1987. The case research stratey in studies of information systems. MIS Quart. 11 (3), 369–385.

Cantera, J.M., Issa, J.S., Vlugt, P.v.d., Klaeser, S., Bartram, T., Kassahun, A., Neira, I., Milin, T., 2018. IoF2020 D3.3 Opportunities and Barriers in the Present Regulatory Situation for System Development. https://www.iof2020.eu/deliverables/d3.3-opportunities-and-barriers-in-the-present-regulatory-situation-for-system-development-v1.2.pdf.

Carrez, F., Bauer, M., Boussard, M., Bui, N., Jardak, C., Loof, J.D., Magerkurth, C., Meissner, S., Nettsträter, A., Olivereau, A., Thoma, M., Walewski, J.W., Stefa, J., Salinas, A., 2013. Internet of Things – Architecture IoT-A, Deliverable D1.5 – Final Architectural Reference Model for the IoT v3.0. European Commission, pp. 482. https://iotforum.org/wp-content/uploads/2014/10/D1.5.pdf.

Castaneda, C., 2015. Internet of Things to Become Cornerstone of Excellent Customer Service. http://ww2.frost.com/news/press-releases/internet-things-becomecornerstone-excellent-customer-service-finds-frost-sullivan/.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., Stafford, J., 2010. Documenting Software Architectures: Views and Beyond, second ed. Addison-Wesley.

Eisenhardt, K.M., 1989. Building theories from case-study research. Acad. Manage. Rev. 14 (4), 532–550.

Fitzgerald, J., Larsen, P.G., Woodcock, J., 2013. Foundations for model-based engineering of systems of systems. In: Fourth International Conference on Complex Systems Design & Management, pp. 1–19.

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. 29 (7), 1645–1660.

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design science in Information Systems research. MIS Quart. 28 (1), 75–105.

Infso, EPoSS, 2008. Internet of Things in 2020: A roadmap for the future. European Commission DG Infso & European Technology Platform on Smart Systems Integration, pp. 32.

InternetWorldStats, 2015. World Internet Usage and Population Statistics. http://www.internetworldstats.com/stats.htm.

ISA, 2010. ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod) Enterprise-Control System Integration - Part 1: Models and Terminology.

ISA, 2010. ANSI/ISA-95.00.02-2010 (IEC 62264-2 Mod) Enterprise-Control System Integration - Part 2: Object Model Attributes.

ISO/IEC/IEEE, 2009. ISO/IEC 10746-3:2009 Information technology – Open distributed processing – Reference model: Architecture. https://www.iso.org/standard/55724.html.

ISO/IEC/IEEE, 2011. ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description. https://www.iso.org/standard/50508.html.

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

Jamshidi, M., 2008. System of Systems Engineering: Innovations for the Twenty-First Century. Wiley, Hoboken, NJ.

Jayaraman, P.P., Yavari, A., Georgakopoulos, D., Morshed, A., Zaslavsky, A., 2016. Internet of Things platform for smart farming: experiences and lessons learnt. Sensors 16 (1884), 1–17.

Josey, A., 2018. An introduction to the TOGAF® Standard, Version 9.2 Overview The Open Group, pp. 22.

Kirchhoff, U., Sundmaeker, H., San Martin, F., Wall, B., Campos, J., Xeromerites, S., Terziovski, M., 2004. How to Speed-up the IMSS Related Innovation in Manufacturing SMEs. International IMS Forum, Como, Italy.

Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V., 2010. Smart objects as building blocks for the Internet of things. IEEE Inter. Comput. 14 (1), 44–51.

Kruchten, P., 1995. Architectural blueprints — the "4 + 1" view model of software architecture. IEEE Softw. 12 (6), 42–50.

Kruize, J.W., Wolfert, J., Scholten, H., Verdouw, C.N., Kassahun, A., Beulens, A.J.M., 2016. A reference architecture for Farm Software Ecosystems. Comput. Electron. Agric. 125, 12–28.

Lee, Y.T., 1999. Information modeling: from design to implementation. Second World Manufacturing Congress.

Maier, M.W., 1998. Architecting Principles for Systems-of-Systems Systems Engineering 1 (4), pp. 267–284.

Manikas, K., Hansen, K.M., 2013. Software ecosystems – a systematic literature review. J. Syst. Softw. 86 (5), 1294–1306.

March, S., Storey, V., 2008. Design science in the information systems discipline: an introduction to the special issue on design science research. MIS Quart. 32 (4), 725–730.

Nielsen, C.B., Larsen, P.G., Fitzgerald, J., Woodcock, J., Peleska, J., 2015. Systems of systems engineering: basic concepts, model-based techniques, and research directions. ACM Comput. Surv. 48 (2), 1–41.

OMG, 2011. OMG Unified Modeling LanguageTM (OMG UML), Superstructure: Version 2. 4.1. Object Management Group, pp. 732.

OPC, 2017. Unified Architecture, OPC 10000-1 - Part 1: Overview and Concepts, version 1.04. https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/.

Porter, M.E., Heppelmann, J.E., 2014. How smart connected objects are transforming competition. Harvard Bus. Rev. (November), 65–88.

Raymond, K., 1994. Reference model of open distributed processing (RM-ODP): introduction, In: Raymond, K., Armstrong, L. (Eds.), Open Distributed Processing.

Roussey, C., Soulignac, V., Champomier, J.-C., Abt, V., Chanet, J.-P., 2019. Ontologies in Agriculture. https://liris.cnrs.fr/Documents/Liris-4759.pdf.

Sarni, W.M., Kaji, J., 2016. From Dirt to Data, the second green revolution and the Internet of Things. Deloitte Rev. 18, 4–19.

Schoenberger, C.R., 2002. The internet of things. Forbes, issue 3/18. https://www.forbes.com/forbes/2002/0318/155.html#8167674559d4.

Sowa, J.F., Zachman, J.A., 1992. Extending and formalizing the framework for information-systems architecture. IBM Syst. J. 31 (3), 590–616.

Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S., 2010. Vision and Challenges for Realising the Internet of Things. European Union, Brussels.

Sundmaeker, H., Verdouw, C., Wolfert, S., Freire, L.P., 2016. Internet of Food and Farm 2020. In: Vermesan, O., Friess, Peter (Eds.), Digitising the Industry. River Publishers, pp. 129–150.

Talavera, J.M., Tobón, L.E., Gómez, J.A., Culman, M.A., Aranda, J.M., Parra, D.T., Quiroz, L.A., Hoyos, A., Garreta, L.E., 2017. Review of IoT applications in agro-industrial and environmental fields. Comput. Electron. Agric. 142, 283–297.

Tekinerdogan, B., 2014. Software architecture. In: Gonzalez, T., Díaz-Herrera, J.L. (Eds.), Computer Science Handbook, second ed. Taylor and Francis.

Tekinerdogan, B., 2017. Engineering Connected Intelligence: A Socio-Technical Perspective, Inaugural lecture. Wageningen University.

Tomasi, R., Sundmaeker, H., Rizzo, F., Conzon, D., Montanaro, T., Hovest, G.G., Vyas, A., Berg, J., Manoel, F., Campos, R., 2018. The IoF2020 Use Case Architectures and Overview of the Related IoT Systems. IoF2020 WP3, pp. 221. https://www.iof2020.eu/deliverables/d3.2-uc-architectures-v2-final-1.3.pdf.

Van Aken, J.E., 2004. Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. J. Manage. Stud. 41 (2), 219–246.

VDMA, 2019. ISOBUS Data Dictionary according to ISO 11783-11. https://www.isobus.net/isobus/.

Verdouw, C., Robbemond, R., Kruize, J.W., 2015. Integration of production control and enterprise management systems in horticulture. In: 7th International Conference on Information and Communication Technologies in Agriculture, Food and Environment (HAICTA 2015), Kavala, Greece, pp. 124–135.

Verdouw, C., Wolfert, S., Tekinerdogan, B., 2016a. Internet of Things in agriculture. CAB Rev. 11 (35), 1–12.

Verdouw, C.N., Beulens, A.J.M., Wolfert, J., 2014. Towards software mass customization for business collaboration. In: Annual SRII Global Conference. IEEE, Silicon Valley, San Jose, California, USA, pp. 106–115.

Verdouw, C.N., Wolfert, J., Beulens, A.J.M., Rialland, A., 2016b. Virtualization of food supply chains with the internet of things. J. Food Eng. 176, 128–136.

Verdouw, C.N., Wolfert, J., Beers, G., Sundmaeker, H., Chatzikostas, G., 2017. Fostering business and software ecosystems for large-scale uptake of IoT in food and farming. In: PA17 - The International Tri-Conference for Precision Agriculture in 2017, Hamilton, pp. 1–7.

Weyrich, M., Ebert, C., 2016. Reference architectures for the Internet of Things. IEEE Softw. 33 (1), 112–116.

Williams, T.J., 1994. The purdue enterprise reference architecture. Comput. Ind. 24 (2–3), 141–158.

Yin, R.K., 2002. Case Study Research: Design and Methods, third ed. Sage Publications, Inc.