D3.5 GUIDELINES FOR THE USE OF IOT RELATED STANDARDS IN SMART FARMING AND FOOD SECURITY

WP3

31st March 2021

Guideline towards Synergies across Use Cases based on Data Models in the Meat Trial

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

Document Identification

Project Acronym	**IoF2020**
Project Full Title	Internet of Food and Farm 2020
Project Number	731 884
Starting Date	January 1st, 2017
Duration	4 years
H2020 Call ID & Topic	IOT-01-2016
Date of the DoA	2017-2021
Website	www.iof2020.eu

File Name	D3.5 Guidelines for the use of IoT related Standards in Smart Farming and Food Security
Date	March, 2021
Version	04
Status	Final
Dissemination level	PU: Public
Author	Annette Breemer, Capucine Soum (Wageningen University & Research)
Review	Review Sabine Kläser (GS1/UC5.3), Daan Goense (rmAgro)
Contact details of the coordinator	George Beers george.beers@wur.nl

PROJECT SUMMARY

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

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

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

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

Led by the Wageningen University and Research (WUR), the 100+ members consortium includes partners from agriculture and ICT sectors, and uses open source technology provided by other initiatives (e.g. FIWARE). IoF2020 is part of Horizon2020 Industrial Leadership and is supported by the European Commission with a budget of €30 million.

EXECUTIVE SUMMARY

IoF2020 was implementing diverse IoT based solutions in a total of 33 use cases. Along their implementation, work package 3 was analysing key aspects that are facilitating the reuse of results as well as the replication of solutions. In the second half of the project implementation, the use cases provided feedback with respect to their challenges when managing different data sources as well as how to facilitate interoperability between different systems from IoT devices up to cloud-based services.

Therefore, IoF2020 work package 3 focused work on the use of IoT related standards and specifically data models that are facilitating data and information exchange. This was also accompanied with a workshop series focusing on semantic interoperability organised in close collaboration with the innovation projects ATLAS, DEMETER, NGIoT and OpenDEI. The initiated work is planned to be continued also after the end of IoF2020, taken over by the other projects involved.

At the same time, this deliverable D3.5 can be considered as a practical showcase of relevant (reference) data models that can help developers to learn about the technical environment as well as the agri-food domain, aiming to provide a guideline towards the generation of synergies across use cases when developing specific data models in a use case.

On the one hand, the work was using as reference the data models of the SmartDataModel initiative, providing data models that are free to use and open-licensed where interested parties can evolve them for their own needs. On the other, rmAgro was used, a model to support projects in covering all situations for an intended practical solution and to take ideas from, to make a quick and robust head start with a solution. In practice all those details described by rmAgro are rarely needed and more importantly copying the level of normalisation done in the reference model in a physical model would result in a model with too many entities and too many joins which will negatively affect the performance of the system and it is unnecessary complex. Therefore, in practice the reference model will almost never be implemented as such. However, to enhance interoperability between use cases, the reference model could be used to compare use case model terms to create a kind of a common vocabulary.

Based on the work accomplished, the following recommendations are given:

- Future use cases and initiatives shall carefully analyse their needs towards complexity, interoperability, reuse and replicability of results up front of their work. If these non-functional requirements are prioritised, teams shall aim at a reuse and adaptation of available data (reference) models and standards, finally being able to create a common vocabulary.
- Developers of IoT based solutions shall evaluate and share their solution with regard to registering identifying devices with animals, so new initiatives can learn from it and may come up with a good general solution to this problem, especially when large volumes of animals are involved. A robust solution to this problem is a prerequisite for achieving traceability and transparency of meat. Regarding this topic also GS1's (S)GTIN standard might be investigated. The standard doesn't have allocation rules for livestock yet.
- Reuse, contribute and/or define an international agreed set of values and international recognized code lists for use in the animal domain. If (new) use cases can refer to these standards interoperability will be improved.
- It would help future initiatives to have a pre-selected choice of reference data models and expert data models to choose from to give them a head start in creating a data model. They could be

published in the data marketplace[1] which is described in deliverable D3.6. Each of these pre-selected data models should have

    - A clear description of their scope and purpose and clear and easily accessible usage guidelines;
    - Easy and quick access to the parts of interest for new initiatives.
    - A version control system in place and information about changes and compatibility levels between versions of the (sub) model. After all, the pre-selected data models will be subject to continuous enhancements and improvements. In usages it should be known to which version of the model is referred.

- In order to enhance and improve pre-selected (reference) data models it would be helpful to incorporate knowledge from new initiatives into these data models. It is therefore desirable that collaboration of these new initiatives with administrators of such pre-selected (reference) data models will be made simple.
- The use of a common vocabulary by all solutions would facilitate interoperability, but solutions often have to adhere to stakeholders' terminology. Use case vocabulary (i.e. stakeholders' terminology) might be matched with common concept vocabularies provided by reference data models and expert data models using linked data and semantic web technology. In order to achieve such a linked data solution, it might be helpful if a reference data model could offer a service to create RDF files from its (sub) models. This would enable easy linking to proprietary use case data models by means of semantic web technology.

On top of that, the IoF2020 team highly recommends to make use of openly available resources like the SmartDataModels and rmAgro. The IoF2020 website offers links to such initiatives, additional tools as well as offering further details in the IoT catalogue about the IoT devices deployed in the use cases. Finally, the IoF2020 team started an effort to collect available resources (i.e. data models, standards, code lists) in a market place. Those resources shall be further maintained beyond the runtime of IoF2020, currently handing over the results to the SmartAgriHubs project that will further operate an innovation portal, hosting information and links to relevant resources.

[1] https://market.ioflab.opplafy.eu/

Table of contents

List Of Figures

1. Introduction

IoF2020 was implementing diverse IoT based solutions in a total of 33 use cases. Along their implementation, work package 3 was analysing key aspects that are facilitating the reuse of results as well as the replication of solutions. In the second half of the project implementation, the use cases provided feedback with respect to their challenges when managing different data sources as well as how to facilitate interoperability between different systems from IoT devices up to cloud-based services. Therefore, IoF2020 work package 3 focused work on the use of IoT related standards and specifically data models that are facilitating data and information exchange. This was also accompanied with a workshop series focusing on semantic interoperability organised in close collaboration with the innovation projects ATLAS, DEMETER, NGIoT and OpenDEI. The initiated work is planned to be continued also after the end of IoF2020, taken over by the other projects involved. At the same time, this deliverable D3.5 can be considered as a practical showcase of relevant data models that can help developers to learn about the technical environment as well as the agri-food domain, aiming to provide a guideline towards the generation of synergies across use cases.

1.1. Background

Within the several trials and use cases of IoF2020 lots of data is exchanged between organizations and systems. Due to the bottom-up approach of this project each use case made its own choices regarding the use of existing available data models and/or the development of their own suitable data models. This way a lot of expertise about data modelling has been developed, but at present this expertise is to a large extent individually present in use cases. Bringing this expertise together gives the opportunity to future developments to learn from this gained experience and facilitate future data modelling in agrobusiness.

With respect to standardisation and pre-normative contributions, specifically trials on arable and meat were mostly active. However, having a relation to the dairy trial in terms of unique identification of cattle, tracking & tracing of pasture and information exchange along the chain (specifically taking into account similarities in UCs making use of collars, pedometers and rumen bolus). The fruit and vegetables trials were also collaborating in terms of tracking and tracing (i.e., EPCIS related solutions) as well as farm equipment related work. At the same time, collaboration was ongoing with respect to the interoperability of weather stations and their north-bound integration. Specific contributions to standards and pre-normative work were accomplished with respect to the ISOBUS (ISO 11783), EPC IS (ISO 19987) and CBV (ISO 19988). At the same time, FIWARE Foundation was actively contributing to the evolution of ETSI NGSI-LD specs based on detected needs, taking into account the requirements identified in IoF2020. Previous work in IoF2020 was also resulting in an overview of existing data models that are applicable in arable agribusiness, as is presented in deliverable 3.6 chapter 5. The goal of the present activity is to gain an overview of the used data models in use cases of the Meat trial.

Before starting with this overview and analysis, it is necessary here to clarify exactly:

1. What is meant by data models
2. How data models can be positioned in the IoT reference architecture

Ad 1: There are mainly three different levels (types) recognized in data modelling:

- The conceptual data model defines what the system contains. Its purpose is to organize, scope and define business concepts and rules.
- The logical data model describes the data of a certain domain in as much detail as possible, without regard to how they will be physically implemented. Its purpose is to develop a technical map of rules and data structures.
- A physical data model also describes the physical means by which data is stored (e.g. processing and space capacity allocation). The purpose is an actual implementation of the database.

Ad 2: The IOT reference architecture as described in deliverable D3.3 and D3.6 is depicted in Figure 1. The interoperability points between layers in the reference architecture in combination with related data models are promoted as MIMs (Minimal Interoperability Mechanisms). When we look into data models, the Mediation Layer and the Information Management Layer are of most interest because the use cases developed data models which can be positioned in those two layers:

- **Mediation Layer:** This layer transforms raw data coming from devices or other external services, into curated, harmonized and possibly aggregated data. In addition, this layer is also capable of sending actuation commands to the IOT service Layer.
- **Information Management Layer:** This layer serves mainly as a data hub, enabling publication, consumption, subscription and processing of all the information relevant to a food and farming solution. The information present at this layer can be current or historical and may have been aggregated from different sources, not only IoT. In addition, this layer may offer complex event processing, storage or analytics services, which can generate insights, prescriptions or predictions.

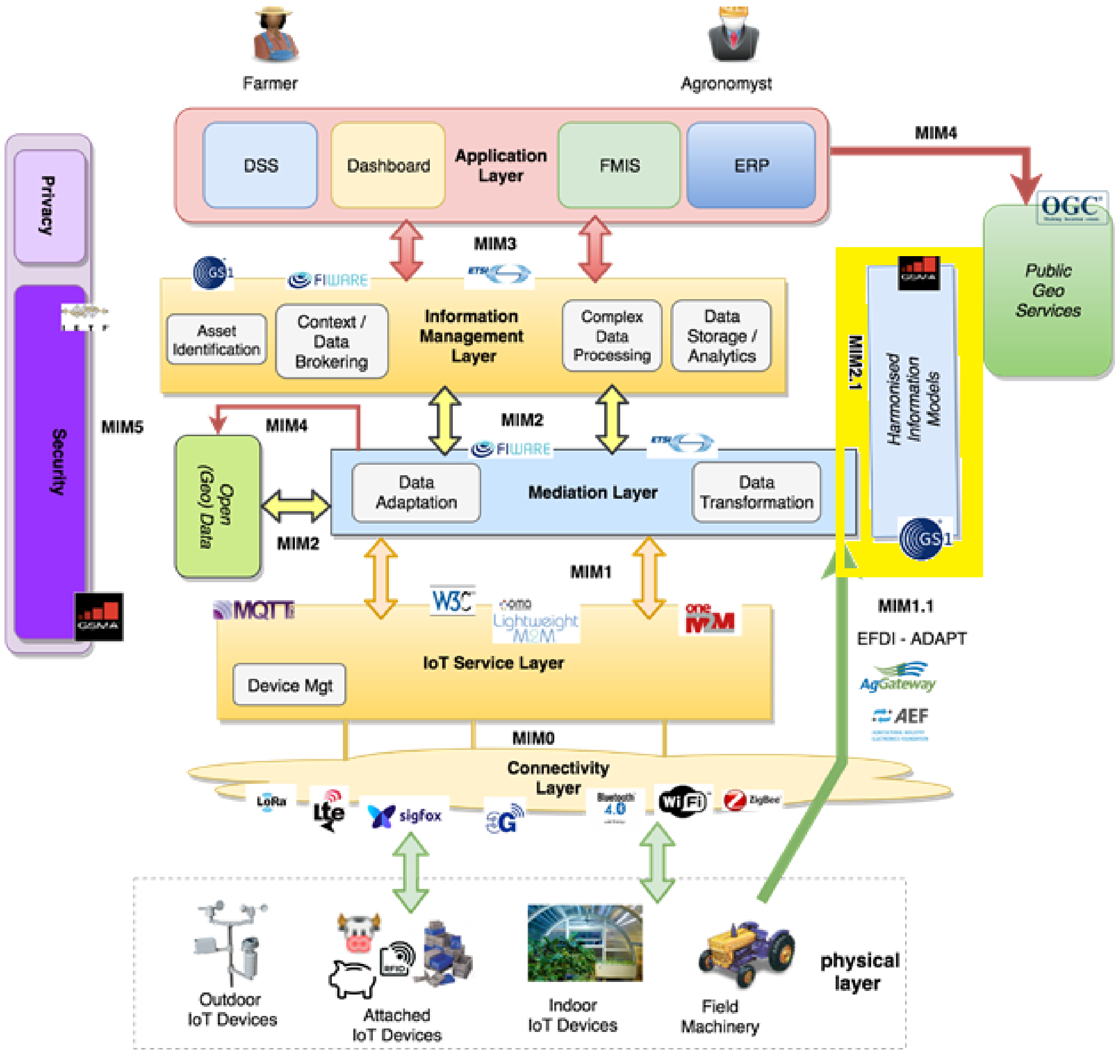


Figure 1: Smart farming Reference Architecture as described in deliverables D3.3. and D3.6. The focus in this document is on Harmonised Information models.

Additionally, MIM2.1 Harmonised Information Models (yellow marked in the figure) are of interest because they are the main enabler for the MIM2 interoperability points which are situated between the Information Management Layer and the Mediation Layer. Between those layers transformation, aggregation, harmonization of data coming from IoT Devices, agricultural machinery or other sources of information (open data portals, web services providing contextual data, etc.) takes place. MIM2.1 Harmonised Information Models allow for using the same meta-model, data representation formats and conventions (units of measurement, etc.) in transforming and publishing smart farming information. As can be seen from the figure, it serves the Mediation Layer as well as the Information Management Layer.

Whereas the implemented data models of the use cases can be regarded as physical data models, the MIM2.1 Harmonised Information Models can be regarded as logical models; the latter models serve as a reference model to which physical models can relate in order to create a common conceptual framework and to reuse data modelling knowledge from.

There are many reference models. In the analysis we will refer to rmAgro[2], a model that have been previously described in Deliverable D3.6. It is a logical data model for the agricultural domain that is divided in several subdomains. Additionally, we will refer to the Smart Data Models[3], which are harmonised and published data models in a joint collaborative program led by FIWARE Foundation and TMForum. These data models can be seen as logical data models, but they go a step further since they can be used as a blueprint for physical implementation using a FIWARE Context Broker Generic Enabler (GE)[4].

The investigation in the meat trial provides insights about the used data models that are described in the following chapters. These models are physical models which may to a certain extend reflect the logical models as described in rmAgro and/or in the Smart Data Models.

1.2. Objectives of this deliverable

This deliverable has the following objectives:

1. Providing an overview of data models used in the IoF2020 meat trial use cases.
2. Identifying the knowledge from the IoF2020 meat trial use cases data models that can be of added value to reference models like rmAgro and FIWARE Smart Data Models
3. Highlighting the knowledge from a reference model that could be of additional value for the use case data models in order to better address needs in upscaling the use case solution

Ad 1: At the use case level, we often can distinguish several physical data models. For example, there is often context information (the state of a real-world object at a given moment in time) that can be gathered by the IoT solutions, managed by a Context Broker GE. In addition, there is a need of storing the historical data, sometimes structured differently than in the Context Broker GE. Also, a physical data model may be necessary to process data locally because the performance of the network is not sufficient (e.g. algorithm to calculate the food volume in a silo based on pictures). Finally, a different interest and usage of the data can lead to different data models. For instance, in the case of food traceability in the meat value chain, only the necessary data of relevant events is considered.

Ad 2/3: To be able to interchange data easily it is important that the physical data models can relate to commonly accepted logical data models, i.e. reference models, in order to create a common conceptual framework. In that way they can be easily interpreted and used by other parties thus improving interoperability. In this deliverable we look into how well the physical data models of the use cases map on the logical data model of rmAgro and the Smart Data Models of FIWARE. This will give an indication of the interchangeability of data between the use cases. Based on this comparison it also can be established, which additional knowledge from the reference model could be relevant for the use cases and which use case knowledge could be used to enrich the reference models. In the latter case the knowledge gained by the use case will be better accessible and reusable by future use cases.

---

2 Wageningen University & Research, ftp://pragmaas.com/rmCrop/rmAgro_SNAPSHOT/

3 https://github.com/smart-data-models & https://smartdatamodels.org/

4 For more information, please consult the different guidelines of the Context Broker and the Smart Data Models e.g. https://fiware-datamodels.readthedocs.io/en/latest/howto/index.html, https://www.fiware.org/developers/catalogue/

1.3. Approach

In this analysis the six use cases of the IoF2020 meat trial were taken into account. It concerns the following six use cases:

- UC 5.1 Pig Farm Management:
- UC 5.2 Poultry Chain Management
- UC 5.3 Meat Transparency and Traceability
- UC 5.4 Decision-making Optimisation in Beef Supply Chain
- UC 5.5 Feed Supply Chain Management
- UC 5.6 Interoperable Pig Health Tracking

Please refer to the IoT catalogue (https://www.iot-catalogue.com/) and the use case progress reports for an extensive description of the use cases. However, also the following public deliverables provide further information about the IoF2020 use cases:

- D1.2 – Catalogue of Use Cases – available via the IoF2020 website – www.iof2020.eu
- D2.7 & D2.12 – Scale-Up Demonstration Reports
- D3.2 – The IoF2020 Use Case Architectures and overview of the related IoT Systems
- D3.10 – Open Call Realisation

Three of the use cases in this trial were already quite elaborated and therefore good candidates to analyse the data models used. In addition, many of the use cases in the meat trial used FIWARE or FIWARE compliant solutions, so it could be beneficial to explore how well FIWARE Smart Data Models can benefit from use case expertise and vice versa.

Since the availability of the use cases was limited, the analysis of the data models was basically performed by interviewing the six use cases from the Meat Trial. The following activities were performed:

- *Study of already available IoF2020 deliverables with regard to the meat trail uses cases*:
  The first three uses cases already shared earlier in the project details of their data architecture in D3.2 Use Case Architectures and D3.3 Reference Architecture for Interoperability. Besides, a few details of UC5.4 were described in D3.6 Enhancement and Configuration of Open Platforms and Reusable Components. Finally, synergy ideas as mentioned in document D3.9 Progress Report on Synergy Analysis, decisions and Coordination of Work were used as input for this analysis.
- *Elaboration of a questionnaire as basis for discussion*:
  Based on already available information a questionnaire was prepared as basis for discussion in an interview.
- *Interview with each use case*:
  The six use cases from the Meat trial were all interviewed based on the present questionnaire. During the interview some overall information was gathered. However, the main focus was on gathering information about the data models used in those use cases.
  The interviews were performed in collaboration with Alberto Abella from FIWARE who gave recommendations on the used data models and discussed the possibilities to publish use case data models as FIWARE Smart Data Models.

- *Gathering of domain and data model term definitions*:
  To better understand the use cases and the data they were dealing with we started a glossary and a data dictionary for two use cases, i.e. UC5.1 and UC5.2.
- *Addressing of additional questions by email*

In the next chapter the results of the data model analysis are described and in the last chapter some general guidelines and recommendations are suggested to use for future initiatives and projects.

2. Analysis of the Data Models in the Meat Trial Use Cases

With the limited access to the use cases and some use cases still very much in development it was difficult to get an appropriate understanding of the used data models. We asked for glossaries and data dictionaries to get a better understanding of the terms used in the data models, but they mostly[5] were not available. In this task we started one for UC5.1 and UC5.2, two use cases for which we received sufficient information to create an initial population. The use cases themselves supplemented and corrected them as necessary. Since it was very labour intensive and time consuming to complete the glossaries and data dictionaries for getting the content right, there was an imbalance identified considering effort versus usefulness in the context of an innovation exercise. Therefore, this initiative wasn't continued for the other use cases. Instead, we focussed for this data model analysis on the general information we were able to gather in the interviews supplemented with information gathered by e-mail.

The information gathered about the data models used in the IoF2020 Meat Trial is summarised in the Appendices. For each use case an appendix is added.

Due to the bottom-up approach of the IoF2020 project, each use case was able to make its own choices. However, most use cases used an entity-based type of data modelling. Even when an event driven type of data model is used, the data sources for the event model are entity based. In order to enhance interoperability between use case data, it is important to be able to refer to a common vocabulary. As mentioned in the IoT reference architecture (Figure 1) for MIM2.1 Harmonized Data Models, there are a lot of options to choose from, taking into account the reference model, as there is for instance the GS1 Standards for data exchange (e.g. EPCIS based on the Core Business Vocabulary) and identification (GTIN, GLN), the Smart Data models from FIWARE but also data reference model like ADAPT and rmAgro (see also IoF2020 deliverable D3.6 for a description of these models).

Most use cases in the meat trial use FIWARE technology and NGSI, a simple yet powerful open API for context information management published as ETSI Specification[6]. UC5.3 didn't, but they developed an NGSI-EPCIS gateway (OLIOT-gateway)[7] as a proof of concept to prove the ability to automatically obtain data from the FIWARE Orion Context Broker into the EPCIS based EPCAT system they use. Therefore, FIWARE was involved in this analysis to advice the use cases on re-use of existing FIWARE models and to look for opportunities to enhance the FIWARE Smart Data Models with knowledge of the use cases, may be even publish some use case data models as new Smart Data Models.

To harmonize the data models, and thus enhancing the interoperability, a mapping on references data models like ADAPT and rmAgro can be performed. Both reference data models have their roots in arable, but lately rmAgro has evolved in the animal husbandry domain by cooperating with stakeholders during different projects. The last project contributing to rmAgro was DATA-FAIR, in the trial 'Carbon footprint pig production'. Since already some projects in animal husbandry contributed to rmAgro, this

---

5 In UC 5.3 the GS1 Core Business Vocabulary was used for the data model.
6 https://www.etsi.org/deliver/etsi_gs/CIM/001_099/004/01.01.01_60/gs_CIM004v010101p.pdf
7 https://public.3.basecamp.com/p/2de58uLTzXWwfG9zhQrg2pop

reference model seems a good candidate to map the IoF2020 meat trial use cases on and to identify what the use cases could learn from rmAgro and what knowledge form the use case could be incorporated in rmAgro, in order to be of use for other projects.

Based on the above the following analysis activities were defined:

- Do different types of data modelling have added value?
  Basically, two different types of data modelling are used by the use cases, namely entity based modelling and event based modelling. Do they have their own benefits?
- Common challenges to address in data models.
  Particularly when creating data dictionaries, a number of issues emerged that probably apply generally to the use cases.
    - For technical reasons, data often has to be stored in different places. Uniquely identifying the correct animals and locations with IoT readings is often a challenge.
    - In addition, a code list, a predefined list, is often desired with certain registrations, for example the breed of the cow, the phenological condition or reproductive condition and the health status.
- Comparison of the use case data models with FIWARE Smart Data Models.
  This activity was performed by FIWARE. The use cases were advised on re-use of existing FIWARE models and opportunities were looked upon to enhance the FIWARE Smart Data Models or even publish some use case data models as new Smart Data Models.
- Comparison of the use case data models with reference model rmAgro.
  What could use cases learn from rmAgro and what knowledge from the use cases could be incorporated in rmAgro, in order to be of use for other projects?

2.1. Different types of data modelling

When looking at the Appendices we notice that basically two different kinds of data modelling were used by the meat trial use cases, see for instance the data models used in UC5.1 (see appendix UC5.1 Database in IOT Platform) and the data models used in UC5.3 (see appendix UC5.3 Event details). UC5.1 uses an entity-based data model as most use cases do. UC5.3 however, uses an event based data model. Also the block chain solution of UC5.4 is event driven.

Figure 2 depicts the connected solutions model by Porter (2014), also described in IoF2020 deliverable D3.9. This model illustrates the growing interconnectedness when IoT solutions mature. The information gathered by IoT solutions is used in many processes and therefore leads to integration with the applications supporting these processes. Over time interconnectedness, it will lead to a so-called system of systems. When combining information from different systems (i.e. a system of systems) one mostly is interested in only certain aspects of the original system. This is the case in UC5.3 where an event-based data model is chosen. It is also true for the blockchain part of the solution in UC5.4, where it was intended that a blockchain solution provides objective data about calf life cycle to consumer.

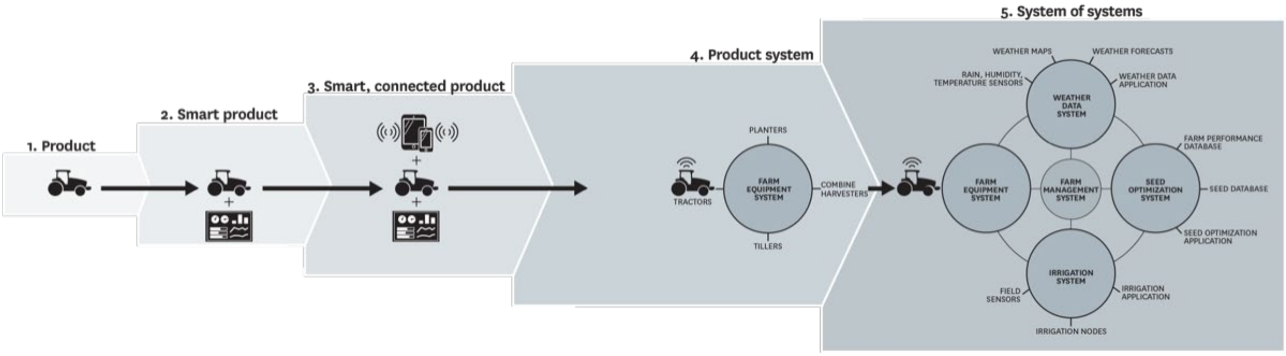


Figure 2: The Connected Solutions model by Porter[8]

From the above we can conclude that the choice for an event model mostly is prompted by the need to combine information from different systems (system of systems) into final information with a clear objective. This is the case in UC5.3 and UC5.4 in the need to create insight in meat transparency and traceability for the consumer. A different data need will arise when it concerns the monitoring and control of a certain (farm) process (a system) where all available data is needed to manage the process appropriately. This might include data from sensors, data about animals and animal housing, machines and maintenance, supply and inventory and so on.

When looking at the connected solutions model by Porter and the data model solutions chosen by the use cases we might conclude the following:

1. An entity-based data model is chosen for a single (sub) system (e.g. a farm system or a feed supply system). Such a system needs a detailed data model able to hold details about all objects, situations and events in order to manage that system well. For such a system, it is often necessary to integrate different expert domains like sensors, animal husbandry and expert knowledge of the animal species.
2. An event-based data model is chosen when a certain information need across multiple systems has to be met, like in the case of tracking and tracing animals and meat over the meat processing chain in order to keep track of what happened to the meat until hitting the consumers' plates. In those cases, only certain information aspects from source systems are relevant, for instance environmental, health and/or sustainability aspects. Individual systems as mentioned in the previous point are used as a source to capture that specific information. As we saw in UC5.3, and the blockchain in UC5.4 an event-based model meets the needs of such requirements very well.

2.2. Common challenges to address in data models

When creating data dictionaries, a number of issues emerged that probably apply generally to all the use cases. In this analysis two of these issues are looked into in more detail:

- Uniquely identifying the correct animals and locations with IoT readings. For technical reasons, data often has to be stored in different places, but still must be processible and unambiguous

[8] Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, *92*(11), 64-88.

beyond local systems. Registration of animals and animal housing mostly will take place in a farm management system. This information has to be combined with the measurements coming from the IoT sensors.

- Often a code list, a predefined list, is desired with data registration. For example, to register the breed of the cow and the phenological and reproductive condition one would like to choose from a predefined list of values. Often these lists already are standardized on a global level.

2.2.1. Uniquely identifying the correct animals and locations with IoT readings

In general, in IoT applications it is mainly about observations and the objects that are observed. In case of the meat trial the objects to be observed are the animals and those animals are subject to the specific environment they are housed in. So apart from the measurement value itself, additional context information is required. It has to be known to which animal or animal group the IoT based measurement applies and often at which location this measurement was taken to be able to manage housing conditions and/or to determine the animals held at that location.

There are several reasons for requiring animal identification. In addition to the need to be able to uniquely identify an animal on a farm in order to optimize animal care, there are also legal rules[9] that require such unique identification, such as rules regarding food safety and public health or to be able to control infectious diseases such as BSE for cattle. In IoT applications animals can be identified by means of a responder containing a microchip which transmit the required identification. ISO 11784[10] (structure of the radio-frequency identification code for animals) and ISO 11785[11] (technical concept of animal identification) provide standards for this kind of animal identification.

In the use cases of the IoF2020 meat trial when animals are individually identified, they are identified by transponders, attaching devices like an ear tag or a collar directly at the animal. These physical devices might need to be replaced when they are broken or when they are reused for new animals as the original animal is slaughtered. In those cases the use cases are faced with the challenge to register the correct physical device with the correct animal.

Sometimes animals are only identified as a group. That animal group is then bound to a certain animal housing. However, groups can change because of deaths and new group members. Also regrouping of animals can take place. Here also the use cases are faced with the challenge to register the correct animal group composition with an animal location.

One could say that for the meat trial use cases a common topic of interest is how to uniquely identify an animal or a location, since the registering of locations (animal housing) and animals in many cases is done in a database separated from the IoT solution itself. Animals and animal housing compartments are given unique id's generated by these separated databases and those unique ids are not known at the equipment level were the observations are done. The way the use cases solved this problem often is reflected in the data models in various levels of the use case solution. The context broker has to be able to put all this information together in order to trigger the right actuation or to provide the measured values in context of the measured object and place, so the problem must be solved before or during the

[9] https://ec.europa.eu/food/animals/identification_en
[10] https://www.iso.org/standard/25881.html
[11] https://www.iso.org/standard/19982.html

context broker processing. For more information on this topic see IoF2020 deliverable D3.9 where the typical feedback loop in IoT systems is explained in section 3.3. With regard to an event-driven model IDs of locations are uniquely identified and known across the relevant stakeholders. Locations are part of every event – either as a read point and/or a business location.

The specific situation in each of the meat trial use cases is presented in the following sections.

2.2.1.1. UC5.1 Solution – Pig Farm Management

UC5.1 recognizes two separate implemented data models, one for the context broker and one for the database, see Table 7 and Table 8 in the appendices. There are some differences between those data models, due to nature of the technical solution. The context broker data model doesn't have an entity Consumption and Measurement like in the database data model. Its solution had to be different because of a constraint about the amount of data that can be shared per second and the fact that only the last received data is stored. The database is organized different because it is meant e.g. to support queries about measurements in time.

When plotting these data models on the IoT reference architecture (Figure 1) both models can be placed in the Information Layer. The context broker model only holds the latest measurements and could be situated in the lower part of the information layer, in close connection with the Mediation Layer. The database contains the historical data (in addition to the latest measurements) and could be situated in the upper part of the Information layer, in close connection with the Application Layer. In the database each entity, for instance pig, is assigned an auto generated unique id. However, the sensors don't know about these unique database ids, they for instance can only read ear tags of a pig, hence a pig is recognized by its ear tag.

UC5.1 solves this problem before the context broker by enriching data gathered from the IoT devices with context data from local databases to enable the identification of the measured entities (e.g. pigs and or pens). This is accomplished by adding the attribute AdditionalInfo in the Context Broker entities (see Table 8 in the appendices). This attribute holds information to uniquely identify animals and/or housing locations, for instance the ear tag reference of a pig, hence enabling the look up of these objects in the database in order to store and provide IoT information in the right context.

This solution, however, needs manual maintenance as identifications of animals and housing may change in time because a sensor may be replaced or moved to another location or animal. If for instance a pig gets a new ear tag then this new ear tag must be registered with the pig in order to link new sensor data to the same pig.

2.2.1.2. UC5.2 Solution – Poultry Chain Management

The UC5.2 architecture and data models have been influenced by the existing tools and technology at the test farms. Each test farm is equipped with the Integral Farm Controller (IFC) designed by Exafan. It is the main controller of elements in farm to optimize animals' conditions (with communications capacity). This dedicated standalone computer collects data from the animal and silo scales and from the water consumption by a specific protocol developed by Exafan. The data is locally stored in the IFC. Several data models are defined to send the data to the Context Broker through the NGSI protocol. These FIWARE data models will enable third party components to be integrated in an easy way with the Poultry Cloud Services. The persistency of the data is ensured by a cloud mongoDB database and the FIWARE connector Cygnus. The list of Data Models is defined in Table 10.

The use case doesn't deal with individual animal identification but deals with herd (group of broilers) identification. The herds are uniquely identified with the combination of several attributes in the data models: idIntegrator, idFarm, idHouse and idFlock. These four attributes respectively inform on the owner of the broilers and where (farm and farm building) and when they are reared. Because these attributes are essential to interpret the data measured by the sensors, they can be found in each of the Context Broker data model dealing with herd information. The IFC stores all the enriched context information related to the farms and herds (flock, areas, scale etc.)

In summary UC5.2 solution to identify housing and animal groups with the measurements is by using the Integral Farm Controller, a system holding all necessary context information for the measurements. From the information gathered it is not possible to determine exactly how this context information is exchanged between IFC and sensors, but we assume the sensors are registered in the IFC as well and therefore when measurements of a uniquely identified sensor are coming in the relevant context information for the measurement can be extracted.

2.2.1.3. UC5.3 Solution – Meat Transparency and Traceability

UC5.3 is about an auditing program in pig meat transparency and traceability, see also the IoT catalogue of IOF2020[12] for more details. This use case does not consume raw data from IoT devices like sensors by itself but uses data from different data sources in the supply chain. The use case focuses still on the farmer, but also include data from a slaughterhouse and a feed company. They do not deal directly with sensor data but capture their information through interfaces with other systems, for instance a FMIS like LeeO or AgroVision and data sources at the slaughterhouse and the feed company.

In this system of systems environment UC5.3 faces an ever-greater challenge of identifying locations and animals uniquely. UC5.3 addressed this by using an event-based system based on GS1's EPCIS[13] standard. EPCIS is intended to be used in conjunction with GS1 standards for Identification like GTIN[14] to identify trade items (also used for animals) and GLN[15] to identify locations. By adapting this EPCIS standard the unique identification of animals and locations is explicitly part of each event and covered by GS1's GTIN end GLN standards, see also Figure 3 for the explanation of an EPCIS event.

GS1 ID keys provide access to master data and are globally unique because of the GS1 Company Prefix. The master data include all key information about the item, product, service, party or location. An example of this master data can be seen in Figure 19 where relevant data of the Pig is available through the GTIN, as pictured in the "What" dimension of the events. An example of master data for locations (GLN) in this use case is listed in Figure 25.

In summary the EPCIS standard enables trading partners to share information about events like physical movement and transformation and capture status of products as they travel throughout the supply chain – from business to business and ultimately to consumers, a system of systems solution. Due to the incorporated use of GS1 ID keys, like GTIN and GLN, unique identification of animals and products of animal origin and locations is part of this solution.

---

12 https://www.iot-catalogue.com/projects/5d95b18df02fdc9e36eaf447/usecase/5b4cbe74b8f13022bb5e6fcd
13 https://www.gs1.org/standards/epcis
14 https://www.gs1.org/standards/id-keys/gtin
15 https://www.gs1.org/standards/id-keys/gln

In UC5.3 GTIN is not yet used for livestock identification at the farm. Since GTIN identification could be used through the supply chain it might be worth to investigate how GTIN could be used for livestock as well. Such a GTIN, e.g. as LGTIN based on animal lots or SGTINs referring to single animals can be associated to a physical device like a collar or ear tag identified by the globally unique GIAI (Global Individual Asset Identifier) Of course, as described in the introduction of this paragraph, still the challenge remains to register the physical identification devices with the correct animal, since replacement of these devices often take place in a remote location without direct access to the registration system.

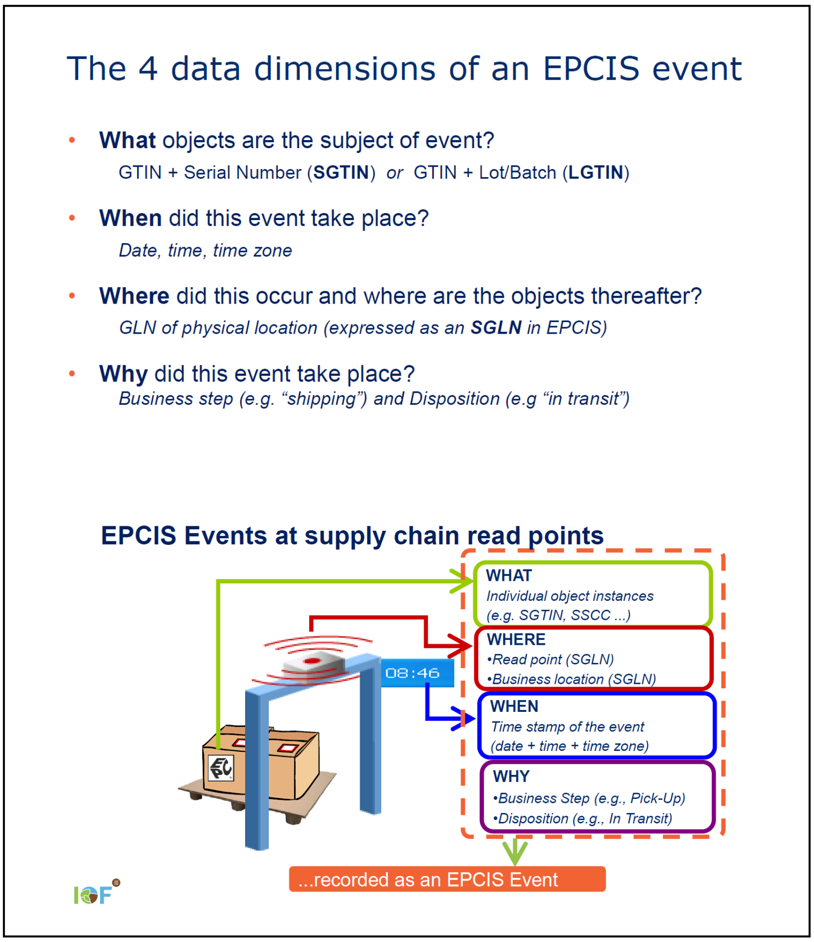


Figure 3: The four data dimensions of an EPCIS Event explained by Sabine Kläser (GS1) in IoF2020 Presentation about EPCIS and UC5.3, March 25th 2020.

In subsequent food chain steps towards slaughtering and processing of animals, the GS1 IDs can be further referenced (i.e. as reference to individual animals or to batches of animals), while asking for additional efforts that are due to the disaggregation of the animal parts as well as requirements for tracking when processing animals to new products.

2.2.1.4. UC5.4 Solutions – Decision-making Optimisation in Beef Supply Chain

UC5.4 Shared beef is about monitoring beef through the meat chain. The use case collaborated with Agricolus who was responsible of the crop monitoring and handled weather stations and field sensors, e.g. for decision making on crops like irrigation and fertilization. The use case also used satellite data to correlate data of the cow-calf system with e.g. pasture quantity. In the meat trial the focus is mainly on the monitoring of animals at the farm level. They collaborated with FIWARE to develop the Animal FIWARE Smart Model, see also appendix 4.4.

UC5.4 used two scenarios to monitor the growing of the animals at the farm, the cow-calf scenario and the feedlot scenario. In the cow-calf scenario they use ear tags for the calves and collars for the mothers. The collar monitors the mother's condition (temperature, activity) and location. The ear tag communicates via Bluetooth with the collar of the mother. This way the approximate location of the calves can be reported by knowing the location of the mother. The ear tag identifies the calf also in the feedlot scenario where the growth is monitored. Because location and time of the measurements is registered the information can be combined with for instance satellite data to correlate this data with for instance the pasture quantity.

From the gathered information it is not explained how the maintenance of ear tags with animals is handled in case of replacement or reuse of the ear tags. It is assumed that this is a manual registration of ear tags, animals and locations (link in UC5.1). Based on that information the right animal and location can be obtained for an IoT measurement.

In order to monitor and control the animal welfare from the birth to the slaughterhouse this use case also works on a blockchain solution to provide objective data about calf life cycle to the consumer. This identification starts with the identification of the calf at the farm by its ear tag.

2.2.1.5. UC5.5 Solution – Feed Supply Chain Management

UC5.5 is about feed supply chain management, see also appendix 4.5. This use case aims to optimise the integral feedstock management system in order to reduce costs and enhance labour efficiency. The farmer, the feed manufacturer and the nutrition expert are interested in monitoring feed stock, and consumption rate in order to achieve timely restocking of feed silos in combination of efficient truck route planning by the feed manufacturer.

In this use case the IoT solution is about measuring the amount of feed in the feed silos. The identification of animals and animal location is therefore not applicable.

2.2.1.6. UC5.6 Solution – Interoperable Pig Health Tracking

UC5.6 aims to optimize the wellbeing and health care costs of pigs with regard to pig farming by intensive monitoring trough IoT sensors, see also appendix 4.6.

The use case collects the information captured by IoT sensors in the pig ear tag in the data model PigHealthGWObserved. Additionally, they combine the pig health information with other information like air quality, noise etc. in order to extract knowledge. This is done based on timestamps and geo location of the measurements. There is a risk of duplicate data storage, when several gateways are installed in one building and are close to each other. It could occur to capture the same information. However, this is mitigated by using an Apache NIFI component. Apache NIFI can collect PigHealthGWObserved data from all gateways and combine data by the timestamp of capturing in order to avoid duplicate data storage.

The sensors themselves can identify the geo location of the measurement. The identification of the animal is based on the ear tag. At the time of interview, the use case didn't have an adequate solution in ear tag management, for instance when an animal gets a new ear tag because it was broken or when an ear tag is reused when the animal died or was slaughtered. Farmers have very many animals and to maintain registration of each animal with a unique identification will be very labour intensive. In case of replacement, it should be very easy for the farmer to register the new tag with the animal, for instance with a phone application which can be used at the location the replacement takes place.

2.2.2. (Potential) use of Standard Code Lists

To register certain characteristics, it is desirable to make use of a predefined code list. With regard to operability, it is desired to use a set of values agreed by an international standard.

In the data dictionaries (see appendices 4 and 4.2) we see this need in the column "Range of values". We also see examples in for instance UC5.3 and UC5.4 as depicted in Figure 4 and Figure 5.

pen	– SGLN	
	– area/pen	
	– capacity/pen	
	– barn SGLN	
	– feed storage SGLN	
	– lightning type	
	– floor type	
	– area/sqm	
	– feeding system	
	– Farming concept	
	– type of Farming	

Figure 4: Potential predefined lists in UC5.3 master data, see also appendix UC5.3 Relevant Master Data for GLN ID's as defined so far.

```
"phenologicalCondition": {
  "type": "string",
  "enum": [
    "lactatingBaby",
    "grazingBaby",
    "maleAdult",
    "femaleAdult",
    "maleYoung",
    "femaleYoung"
  ],
  "description": "Phenological condition of the animal"
},
"reproductiveCondition": {
  "type": "string",
  "enum": ["noStatus", "inactive", "inCalf", "inHeat", "active"],
  "description": "Reproductive condition of the animal"
},
"healthCondition": {
  "type": "string",
  "enum": ["healthy", "sick", "inTreatment"],
  "description": "Phenological condition of the animal"
},
```

Figure 5: Predefined lists in datamodel of UC5.4, see https://github.com/smart-data-models/dataModel.Agrifood/blob/66f5b46597e3562e925f965dfa76952795fa2db6/Animal/schema.json

Reference models like rmAgro and FIWARE Smart Data models strives to include standard set of values (enumerations) and international recognized code lists. In rmAgro the following distinction is made between identifiers, code types and enumerations:

- **IdentifierType** is used to represent objects to enable a common identification of objects. IdentifierType should be used in case of an infinite set of objects. A typical example is Products which are identified with an identifier.
- **CodeType** is used for all elements that are used in the communication between partners or systems to enable a common coded value representation. CodeType should be used in case of a finite case of allowed values. A typical example is the ISO country codelist and language code list.
- **Enumeration** defines a specified set of values, which are part of the reference model, while identifier lists and code lists can be provided by other parties, besides those parties responsible for the reference model itself.

Both reference models hold enumerations and references to code types. In paragraph 2.2.1 we already discussed the unique identification of animals. It is desirable that for the animal domain an international agreed set of values and international recognized code lists could be defined. If (new) use cases can refer to these standards, interoperability will be improved.

2.3. Comparison of the use case data models with FIWARE Smart Data Models

The Smart Data Models initiative was collaboratively initiated by the FIWARE foundation, TMForum and IUDX. The data models are free to use and open-licensed. Therefore, interested parties can evolve them for their own needs. In fact, the SmartDataModel initiative encourages the evolution in the actual repositories, while asking contributors to reach some consensus with other users, and to meet some data model guidelines. The data models have been harmonized to enable data portability for different applications including, Smart Cities, Smart Agrifood, Smart Environment, Smart Sensoring, Smart Energy, Smart Water, Smart Destination, and Starting Smart Robotics and Smart Manufacturing.

The Smart Data Models are available at https://github.com/smart-data-models and can be used with any development work, while asking for compliance to FIWARE NGSI version 2 and NGSI-LD. Contributions and the addition of data models shall take into account the data model development guidelines. This shall be assured to reach a unified approach to smart data, the Smart Data Models, along with contributions from GSMA and TMForum data models. The Smart Data models are stored in repositories. The lower-level repository is a Subject and every subject repository is aggregated into Domain repositories like Smart Farming. Domain repositories compile several subjects. At the same time, a subject could appear in several domains. Also csv examples of the payloads are provided for the data models. On top of that, also work started on some tools to facilitate the usage and test of data models. Most recently, a first version of a service was developed to automatically generate NGSI-LD payload based on a Smart Data Model to facilitate development and test.

In Table 1 the mapping of the data models of the meat trial use cases is summarized. In the following paragraphs per use case some additional explanation is given.

Table 1: Comparison of meat trial data models with FIWARE Smart Data Models

Use Case	**Use Case Data Models**	**FIWARE Smart Data Model comparable models**[16]	**Mapping comments**
5.1 Pig Farm Management	Farm Building Compartment Pen Pig SlaughteredPig Slaughterhouse Company See also Table 7 and Table 8 in the Appendices	Farm ≈ Agrifood.AgriFarm Building ≈ Building.Building Compartment ≈ Agrifood.Zone Pen ≈ Agrifood.Pen Pig ≈ Agrifood.Animal	Currently no relation between Agrifarm and Building (datamodel.Building) exists, because the use cases didn't need it. However Agrifarm has a hasBuilding[17] property which allows for such a connection. Zone is a new candidate model in FIWARE and does cover Compartment. Zone is the more general term for Compartment. Pen is a new candidate model in FIWARE Smart Data Models. SlaughteredPig, SlaughterHouse and Company cannot be mapped, but they do not contain many attributes in the use case model.
5.2 Poultry Chain Management	WeatherPrediction-Observation FarmPopulationObserved AirQualityObserved FarmConsumptionObserved PoultryWeight-Observed CurvesIOfBreeding SensorAirQualityObserved See also Table 10 in the Appendices	WeatherPrediction-Observation ≈ Weather.Forecast AirQualityObserved, SensorAirQualityObserved ≈ EnvironmentAirQualityObserved	FarmPopulationObservation, FarmConsumptionObserved, PoultryWeightObserved, and CurvesOfBreeding are not comparable with the current FIWARE Smart Data Models.
5.3 Meat Transparency and Traceability	Growth Event (Why part), see Figure 21		UC5.3 didn't use an entity based model, so the data models cannot be compared with the FIWARE Smart Data Models.

---

[16] https://github.com/smart-data-models
[17] https://github.com/smart-data-models/dataModel.Agrifood/blob/60755d5e970f18938a59dcb190a91ac5622c6c11/AgriFarm/schema.json

Use Case	Use Case Data Models	FIWARE Smart Data Model comparable models[16]	Mapping comments
5.4 Decision-making Optimisation in Beef Supply Chain	Animal model, see Table 11 AgriParcel AgriFarm	Animal ≈ Agrifood.Animal AgrParcel ≈ Agrifood.AgriParcel AgriFarm ≈ Agirfood.AgriFarm	UC5.4 originally developed the Animal data model in collaboration with FIWARE. They also used the AgriFarm
5.5 Feed Supply Chain Management	Silo Silo Model Device DeviceModel Farm See also Figure 29 and Figure 31 and further	Silo ≈ WastManagement.WaisteContainer SiloModel ≈ WastManagement.WaisteContainerModel Device ≈ Device.Device DeviceModel ≈ Device.DeviceModel Farm ≈ Agrifood.Farm	At the time of the interview UC5.5 was in the process of developing the data models using NGSI schema
5.6 Interoperable Pig Health Tracking	Device DeviceModel PigHealthGWObserved AirQualityObserved NoiseLevelObserved PigHealthObserved see Figure 37, Figure 38 and Figure 39	Device ≈ Device.Device DeviceModel ≈ Device.DeviceModel AirQualityObserved ≈ EnvironmentAirQualityObserved NoiseLevelObserved ≈ Environment.NoiseLevelObserved	At the time of interview PigHealthGWObserved and PigHealthObserved were in the process of being defined and not (yet) openly available as Smart Data models at FIWARE

2.3.1. Mapping with UC5.1 Data Models

UC5.1 didn't use the FIWARE Smart Data Models while developing the use case but they were willing to share their data models with the help of FIWARE under the Creative Commons open licences. Pen and Compartment will be included in the FIWARE Smart Data Model as Pen and Zone.

The Slaughterhouse entity is not used at the moment and Company and Farm identifications were only replicated because the organisation that is using the data asked for it. Otherwise, they had to consult every time the database to understand the connection with the Farm, Company. Because of the very few attributes in those entities mapping on existing FIWARE Smart Models is not applicable.

Health and treatments can be added directly to the dashboard but does not go over the IoT platform, so they are not included in the context broker data model.

FIWARE recommended them not to have too many compulsory fields in order not to break the system, if one field is missing.

2.3.2. Mapping with UC5.2 Data Models

In UC5.2 the data models had to be based on the data already available at the Exafan Farm Management Application. This Exafan controller was already in use at the farms before the project and contains information like the day of the flock, the number of chickens and the desired temperature of the animal housing. The models FarmPopulationObservation, FarmConsumptionObserved, PoultryWeightObserved, and CurvesOfBreeding are therefore not comparable with the current FIWARE Smart Data Models.

The AirQualityObserved and SensorAirQualityObserved model do at least in name resemble the FIWARE Smart Data Model Environment.AirQualityObserved but they are not yet the same. The idea is that there will be just one data model in the end. If necessary, attributes needed by UC5.2 will be added to the FIWARE Smart Data Model. This process is still going on.

2.3.3. Mapping with UC5.3 Data Models

UC5.3 didn't use an entity based model. However, in a proof of concept KAIST developed an NGSI-EPCIS gateway (OLIOT-gateway)[18] to prove the ability to automatically obtain data from the FIWARE Orion Context Broker into the EPCIS based EPCAT system they use. This gateway is now part of the FIWARE catalogue and available through GitHub[19]

In the proof of concept, the data models Building and Pig from UC5.1 where used. The UC5.1 data model Building contained sensor information like temperature, humidity and luminosity. This information was used to populate an event with sensor values and the UC5.1 data model Pig can be used to populate a Growth event or a Feed Intake event respectively Water Intake event. As we can see in Table 1 the Pig does roughly compare to the Agrifood.Animal. Additionally, the sensor observations of the Building do compare to those of the candidate model Pen in FIWARE Smart Data Models. This proves that FIWARE Smart Data Models can be used as source for events in an event-based system based on GS1's EPCIS standard.

2.3.4. Mapping with UC5.4 data models

UC5.4 originally developed the Animal data model in collaboration with FIWARE. They also used the AgriFarm and AgriParcel from the available FIWARE Smart Data Models. Since the FIWARE Smart Data Models were slightly changed because of new insights and probably also the UC5.4 models changed during development they now differ slightly from each other. The existing smart data model was generalized but the adoption is uncertain, because it has a low priority compared with other changes.

2.3.5. Mapping with UC5.5 data models

At the time of interview, the effort of data harmonization by developing data models using NGSI schema was still under development. Even though, they recognized that bin monitoring is quite similar to the FIWARE Smart Data Models for smart cities waste monitoring. They used these models as inspiration

---

18 https://public.3.basecamp.com/p/2de58uLTzXWwfG9zhQrg2pop
19 https://github.com/yalewkidane/FIWARE_EPCIS_Mediation_Gateway

for the development of their NGSI schema data models Silo, Silo Model. They also used the Device and DeviceModel and Farm from the FIWARE Smart Data device model and agrifood model as an inspiration. As of today, it is not known whether this use case indeed harmonized their models with the FIWARE smart data models, but it seems unlikely because of the low priority of this task.

2.3.6. Mapping with UC5.6 data models

HOP Ubiquitous, the technical company involved in UC5.6 is a gold member of the FIWARE Foundation and participate in the technical committee meetings that happen every week. They therefore try to adhere to the FIWARE Smart Data Models. However, at the time of the interview they seemed to slightly differ from the FIWARE models. This could easily happen as both FIWARE as HOP Ubiquitous were in the process of developing and improving their models.

During the interview HOP Ubiquitous told us that PigHealthGWObserved and PigHealthObserved are being defined and not (yet) openly available as Smart Data models at FIWARE. As of today the process of defining new candidate FIWARE Smart Data Models is still pending.

2.4. Comparison of the Use Case Data Models with rmAgro

To analyse interoperability, we also mapped the meat trial use case models on rmAgro. These data models are described in detail in the appendices 4 to 4.6. In the following chapter, only their names and some of their attributes will be quoted.

The primary focus of rmAgro was initially not the animal husbandry domain. rmAgro is the continuation of rmCrop, which was developed from the perspective of arable farming with emphasis on precision agriculture. In cooperation with stakeholders during different projects it was later extended to the horticulture and flower production chains and recently also to animal husbandry. rmAgro is an extension of rmCrop that now strives to cover the whole of primary agricultural production. The last project contributing to the animal husbandry domain was DATA-FAIR, in the trial 'Carbon footprint pig production'.

In summary rmAgro is a reference model in which knowledge from a lot of projects in the agricultural domain is captured. It is a logical data model and when new knowledge is acquired the model is changed accordingly to this knowledge without regard of installed bases of backward compatibility. This reference model strives to account for each detail and tries to avoid as much as redundancy as possible, resulting in a very normalised and detailed data model. That is why it is a reference model, a model to support projects in covering all situations for an intended practical solution and to take ideas from, to make a quick and robust head start with a solution. In the IoT reference architecture (Figure 1) rmAgro could be placed as part of Harmonized Information Models.

In practice all those details described by rmAgro are rarely needed and more importantly copying the level of normalisation done in the reference model in a physical model would result in a model with too many entities and too many joins which will negatively affect the performance of the system and it is unnecessary complex. Therefore, in practice the reference model will almost never be implemented as such. However, to enhance interoperability between use cases, the reference model could be used to compare use case model terms to create a kind of a common vocabulary. Currently a pilot is started to create first a sub-model out of rmAgro for a particular use case, which holds only the classes, attributes and relations relevant for that use case and secondly a corresponding RDF[20] output. The RDF output could possibly be used to link terms from use case data models to rmAgro terms in a linked data solution and thus creating a common rmAgro vocabulary.

Additionally, since rmAgro strives to be a reference model for agricultural domain as a whole, it also contains connections between sub models like e.g. animal housing and sensors. Those connections are probably less clear in disconnected sub models like e.g. OGC's sensorML[21] and the Semantic Sensor Network Ontology[22] it contributed to. The reference model might therefore help to identify important details in linking different sub models. In rmAgro, sensor observations are seen as part of operations and tasks and the sensors themselves are, particularly in arable farming, an integrated component of farm machinery.

Since rmAgro is a huge canonical data model it might be challenging to find the details of interest. To ease the comparison with the meat trial use case models we took sub models from rmAgro to cover the following data model subjects:

- rmAgro Animal (including animal housing and feed allocation)
- rmAgro Sensors (including all kind of IoT measurements)

The sub models must be regarded as subject related views on rmAgro and are still part of rmAgro as a whole. We compared the use case data models on both subjects with the following questions in mind:

- What knowledge incorporated in rmAgro could be of use for the use cases, especially while the use cases are upscaling?
- What knowledge from the use cases could be incorporated in rmAgro, in order to extend the knowledge in rmAgro and as such be of use for other projects?

20 https://www.w3.org/TR/rdf11-concepts/
21 http://www.sensorml.com/index.html
22 https://www.w3.org/TR/vocab-ssn/

2.4.1. Mapping Use Case Data Models on rmAgro Animal

2.4.1.1. The modelling of the animal domain in rmAgro

In rmAgro, the sub-domain package called drmAnimal contains the classes and the associations specific to that domain. As it has a lot in common with arable farming, in addition to this package, a lot of classes are used from other packages.

The **AnimalHoldingSite** entity is part of an **Organization** located at a distinct geographic location, where animals are housed for production purposes (see Figure 6). It stores legal and administrative information. An **AnimalHoldingSite** can have a **FarmYard** and/or **Containers** and a Containers can be of a **ContainerType** Silo. A **FarmYard** may also held one or more **Buildings.**

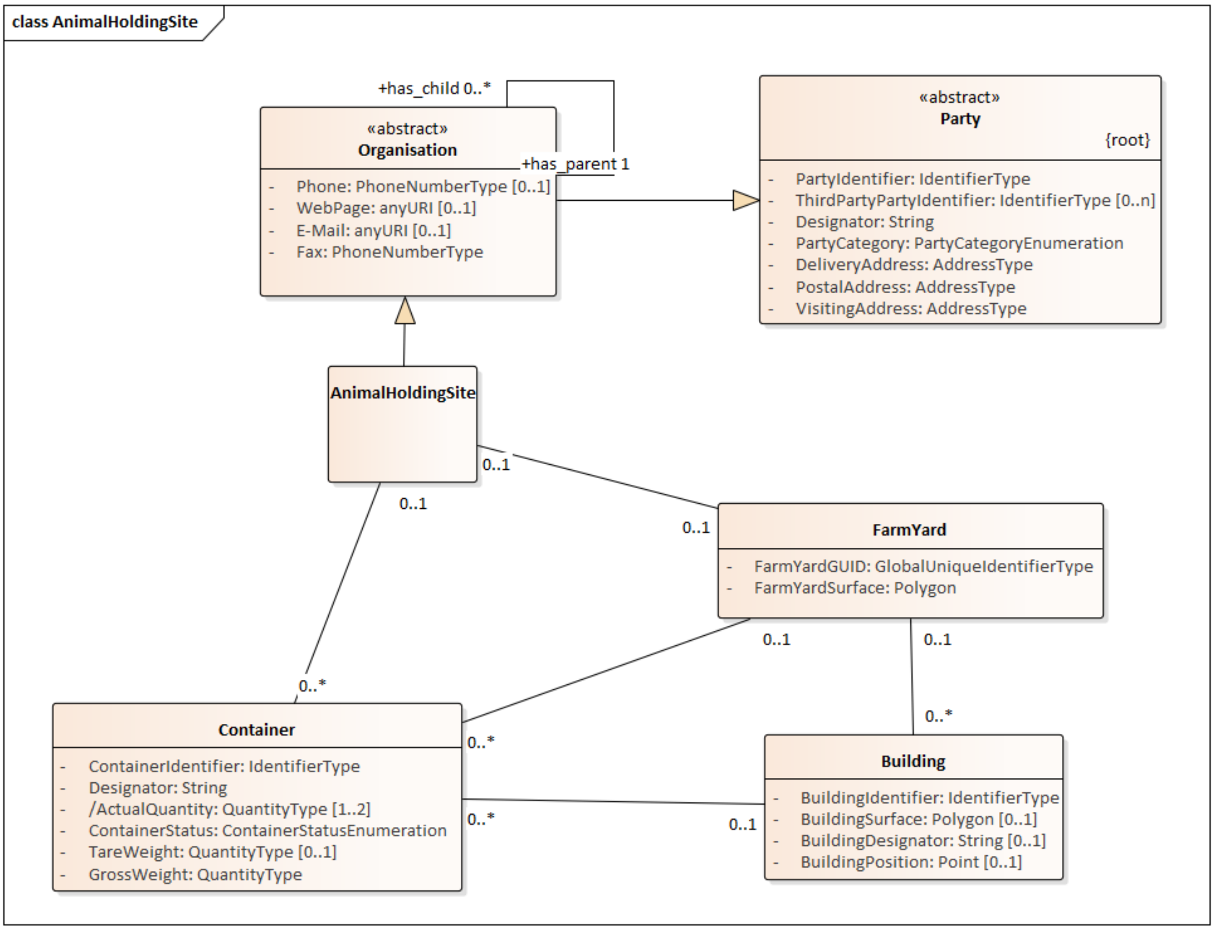


Figure 6: AnimalHoldingSite

The Animal entity stores basic information regarding the animals. In the context of precision livestock farming it is associated to a physical identifier, e.g. an ear tag.

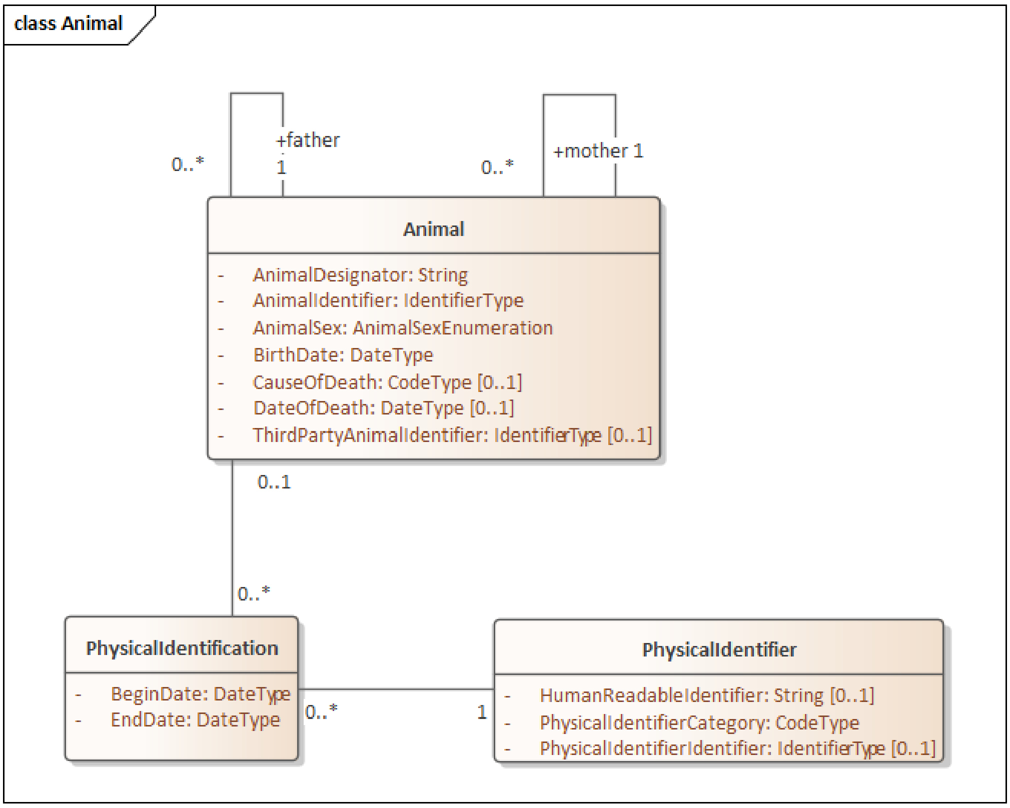


Figure 7: Animal modelling

The animal housing organisation described how and where the animals are reared (Figure 8). This information can be stored to relate measurements to a certain location. The complexity of the housing organisation (sub-divisions in compartments, pens etc.) depends on the livestock domain and the farm. The space that a sensor in an **AnimalHouse** can cover is expressed in **VerticalLayer**s of a **Zone**.

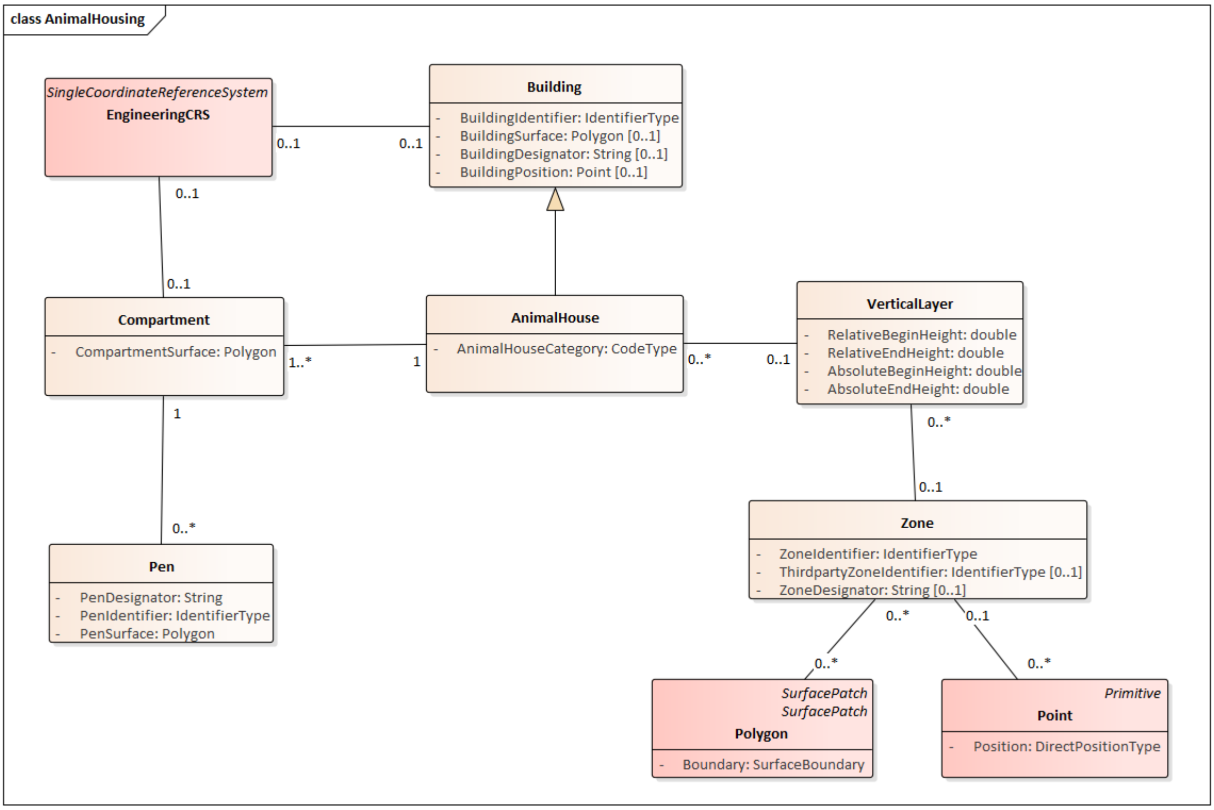


Figure 8: Animal housing modelling

The animal grouping and allocation is represented in Figure 9. **Animals** are individually or collectively (**AnimalGroup**) allocated to a place for a certain period of time. **AnimalGroupParticipation** records the period of time an **Animal** is part of group is an **AnimalGroup.** Transport inside the farm is covered by **AnimalRegrouping**, which specifies how a new group is formed by splitting or merging old groups, or transferring an animal.

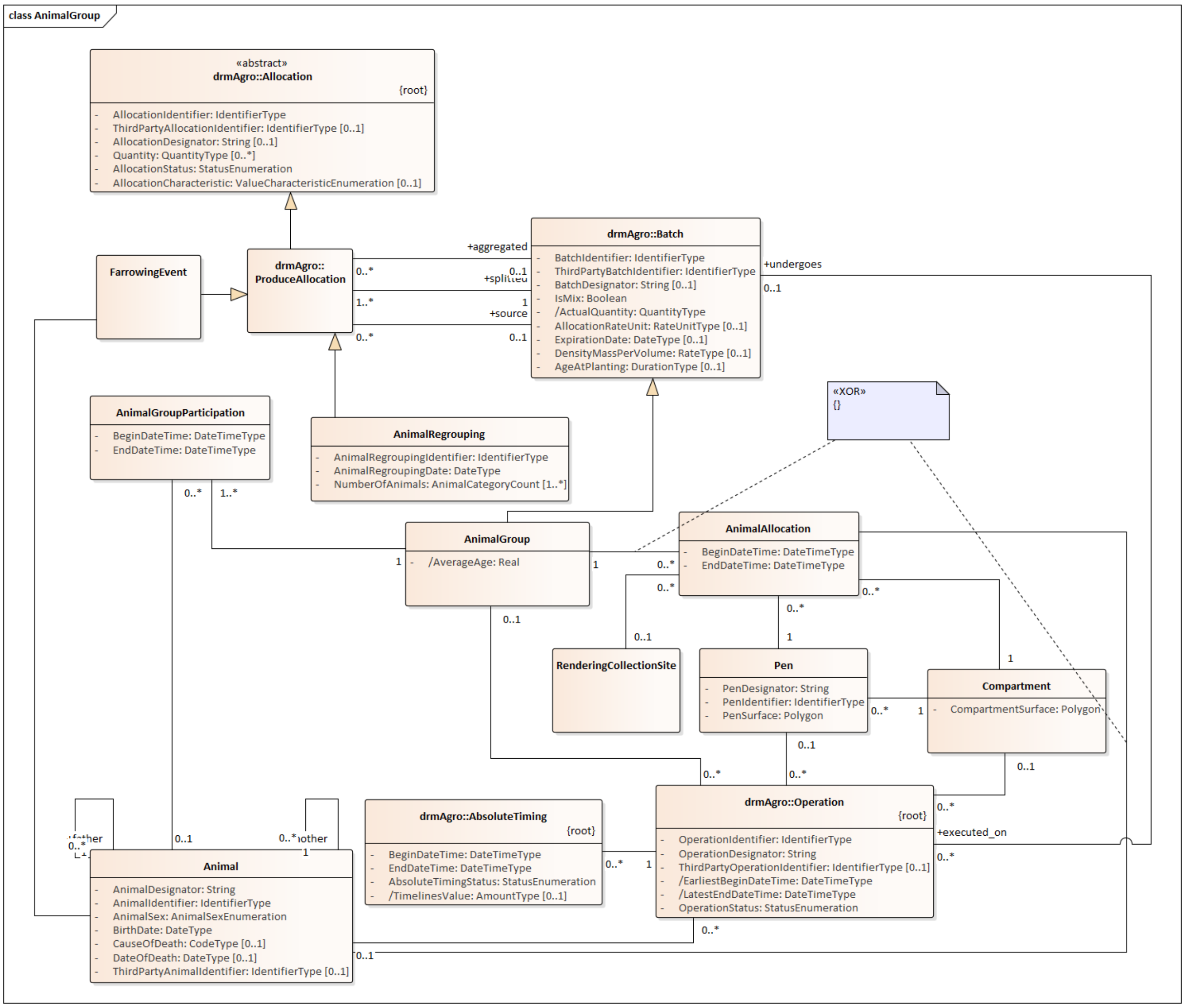


Figure 9: Animal grouping and allocation modelling

As for the animal allocation, the feed allocation is well described in rmAgro (see Figure 10). The feed is a product that is allocated in batches to an individual animal or to a part of the house (e.g. Pen).

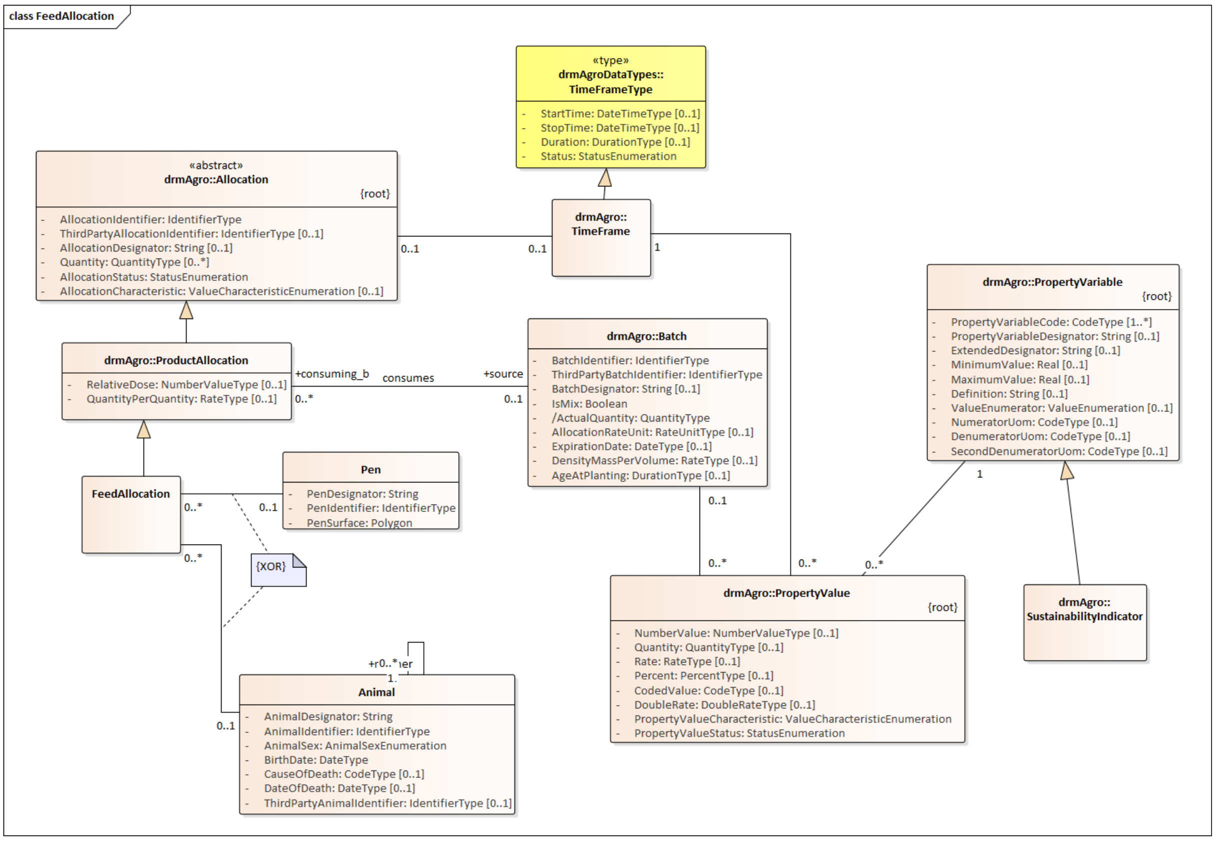


Figure 10: Feed allocation modelling

The inventory and storage of a product is addressed in rmAgro with a general point of view, not particularly centred on the animal and the feed storage (see Figure 11 below). **Silo** is a **ContainerCategory** of **Container** that has, among others, an identifier, a gross weight and a position. A **Batch** (specific identified number or quantity of Product) is stored in a **Container**. An **Inventory** can have several **StockLevel's** with a DateTime as attribute. This makes it possible to calculate expected CalculatedQuantities based on planned use of products. Calculations are based on **Allocations** and **ProductAllocations**. There is also a CountedQuantity when an inventory is made. Furthermore an **Inventory** can have one or more **InvertoryItem's** for which can have a minimum level of stock, i.e. a certain predetermined minimum quantity of a product inventory which should always be available in stock in the normal course of business.

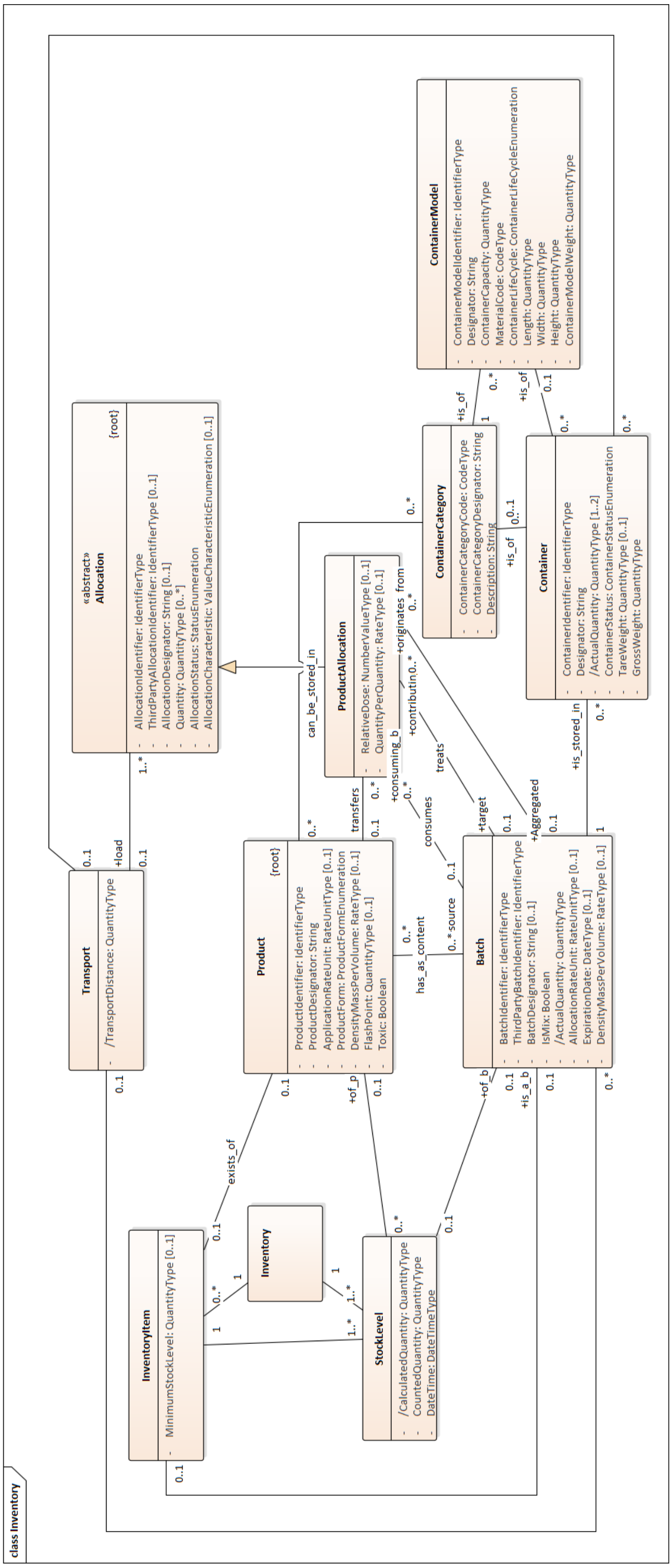


Figure 11: Inventory and product storage.

2.4.1.2. *Comparison of rmAgro animal domain to the meat trial data models*

For the Meat use cases, the animal is at the centre. Therefore, all of them deal, to some extent, with modelling the animal and the housing respectively location. The measurements related to an animal, a group of animal or a housing structure will be addressed in the chapter 2.4.2.

The information regarding the farm is quite limited in the use cases. They often need a unique identification to distinguish one farm from the other.

Table 2: Mapping regarding the farm topic.

Use Case	Use Case Data Models	rmAgro comparable classes
5.1	Farm (Id, name, owner)	**Animal Holding Site, Organisation** and **Party** (Figure 6)
5.2	Not a specific data model for farm but the farm name (idFarm) is used as an attribute in the various data models	
5.3	A farm id or a more granular location is stored in the Where part of each event (see Appendix 4.3, UC5.3 Event details)	
5.4	AgriFarm (used from the Smart Data Model[23])	
5.5	Farm (inspired by the Smart Data Model AgriFarm)	

In the meat use cases, the animals are either individually or collectively monitored. Broiler chickens in UC5.2 are collectively monitored. Pigs and beefs in the other use cases are usually individually or per group monitored using an ear tag or a neck collar[24]. Some indicators are therefore recorded for a group of animals. The table below shows the mapping between the use cases and rmAgro for the animal and housing topics.

Table 3: Mapping regarding the animal and housing topic.

Use Case	Use Case Data Models	rmAgro comparable classes
5.1	Pig	Pig is a subclass of Animal. See Table 5: Mapping of use case sensor measurements to rmAgroTable 5 and Figure 14 the relation with measurements and sensors
5.1	Building	**Building**. See Table 5: Mapping of use case sensor measurements to rmAgroTable 5 and Figure 14 the relation with measurements and sensors

[23] https://github.com/smart-data-models/dataModel.Agrifood/tree/master/AgriFarm

[24] Pigs have in many cases only an identifier linked to an AnimalHolding, and not always a (revolving!) sequence number, such that it is impossible to follow the individual animal during its life.

Use Case	Use Case Data Models	rmAgro comparable classes
5.1	Compartment	**Compartment**. If the compartment doesn't have a physical separator (see the definition in Table 8, in Appendix 4), the **Vertical Layer** and **Zone** class (see 2.4.2) could be used. See Table 5: Mapping of use case sensor measurements to rmAgroTable 5 and Figure 14 the relation with measurements and sensors
5.1	Pen	**Pen**. See Table 5: Mapping of use case sensor measurements to rmAgroTable 5 and Figure 14 the relation with measurements and sensors
5.1	SlaughteredPig	In **Animal** the attributes CauseOfDeath and DateOfDeath allow to register the slaughter. However, the slaughter process itself is not modelled in rmAgro as it is not part of the primary sector.
5.2	FarmPopulationObservation	**AnimalGroup** is a **Batch** of animals. **AnimalGroupParticipation** registers which animal was part of the **AnimalGroup** at a certain time period. Changes in the **AnimalGroupParticipation** can be explained by the animal attributes DateOfDeath/CauseOfDeath or change in **AnimalAllocation** (see Figure 9)
5.2	idHouse is an attribute in the various data models	**Building**
5.3	An animal id (SGTIN) or a group of animal id (LGTIN) is stored in the What part of each event dealing with pigs (see Appendix 4.3, UC5.3 Event details)	**Animal, AnimalGroup** Both Animal land AnimalGroup have an Identifier of the datatype IdentifierType. IdentifierType allows to use any identification scheme among which SGTIN and LGTIN. When the farmer decides to use another scheme, it is possible to use one or more ThirdPartyIdentifier's.
5.3	A barn and/or a pen id (SGLN) are stored in the Where part of each event (see Appendix 4.3, UC5.3 Event details)	**Building, Pen**
5.4	Animal (see Table 11 in Appendix 4.4)	**Animal**. See Table 5 and Figure 14 the relation with measurements like the weight and sensors. Some characteristics (e.g. the reproductive and phenological conditions) could be covered in rmAgro in coding tables, e.g. AnimalVariables

The table below shows the mapping between the use cases and rmAgro regarding the feed supply chain.

Table 4: Mapping regarding the feed and food topic.

Use Case	Use Case Data Models	rmAgro comparable classes
5.4	The attribute fedWith in the Animal Smart Data Model	**Feed** is a subclass of **Product** that has a category, units, a density etc. (see Figure 10)
5.5	Silo Model	**Silo** is a **ContainerCategory** of **Container**. The **ContainerModel** of a **Container** stores characteristics unlikely to change (height, capacity, material etc.) but not as much as Silo Model (see Figure 11).
	Silo (Bin is also used as a more generic term), Recipes	**Container, Batch, Inventory. Silo** is a **ContainerCategory** of **Container** which stores among others, an identifier, a gross and a tare weight and a position. The association between **Container** and **Batch** allows to relate information on the Product(s) to the storage. As explained in Figure 11**,** the **StockLevel** is either calculated based on the planned use of products or counted when an inventory is made. UC5.5 has a different approach by calculating the level based on what the camera sees inside the silo. In rmAgro a new link is added between **Container** and **SensorSytem** enabling support for modelling of sensors determining the actual inventory, see Figure 17. With the type of information store in **Silo**, alerts on the capacity could be set up like in UC5.5, **InventoryItem** can have a "minimum" stock level, but this will be a dynamic attribute as it will also depend on delivery times of the concerned product.

Regarding the location of an animal also transportation of animals is of importance. In rmAgro transport inside the farm is covered by AnimalRegrouping (see Figure 9). "External" transport in rmAgro is covered by TransportOperation, which has a **TransportOrigin** and a **TransportDestination**. The TransportOperation is realized by one or more Transports, which holds one or more **Batches**, which has as subclass **AnimalGroup**. **TransportDestination** and **TransportOrigin** are at this moment identified by their coordinate and a link to either an **ActivityField**, a Storage or a Container. **AnimalHolding** and Processor is now added (see Figure 12). Transport details were very limited in the data models we obtained from UC5.1, 5.2, 5.3 and 5.4, therefore a comparison table for this topic is not included in this report.

2.4.1.3. *Summary and Recommendations*

The animal husbandry domain was not the primary focus of rmAgro and is still not the most complete one. From the mapping as presented above, we learned rmAgro can incorporate the following knowledge from the use case models:

- Phenological and reproductive conditions as specified by UC5.4 could be covered in coding tables, for example AnimalVariables. The content of such a coding table has to be decided in collaboration with experts in the animal husbandry domain.
- The transport and the slaughter process is an important part of the meat supply chain. This topic is only brief addressed in the data models we obtained from UC5.1, 5.2, 5.3 and 5.4. In rmAgro the transport operation in the arable domain is already well modelled (see Figure 12). If more data model details were available concerning this topic in meat use cases, rmAgro could be enriched with these details and so help future initiatives in the animal husbandry domain.
- To support current and future feed supply chain cases, the container characteristics might need some extension. For instance, UC5.5 distinguishes the cylinder from the cone shape of the silo.
- The UC5.5 approach regarding the measurement of the silo's level could be modelled in rmAgro, in addition to the other methods. The use of the already existing classes **DataSet**, **Algorithm** and **DataProcess** is a possibility to explore (see the blue boxed area of the Figure 13 and the new link between **Container** and **SensorSystem** in Figure 17).

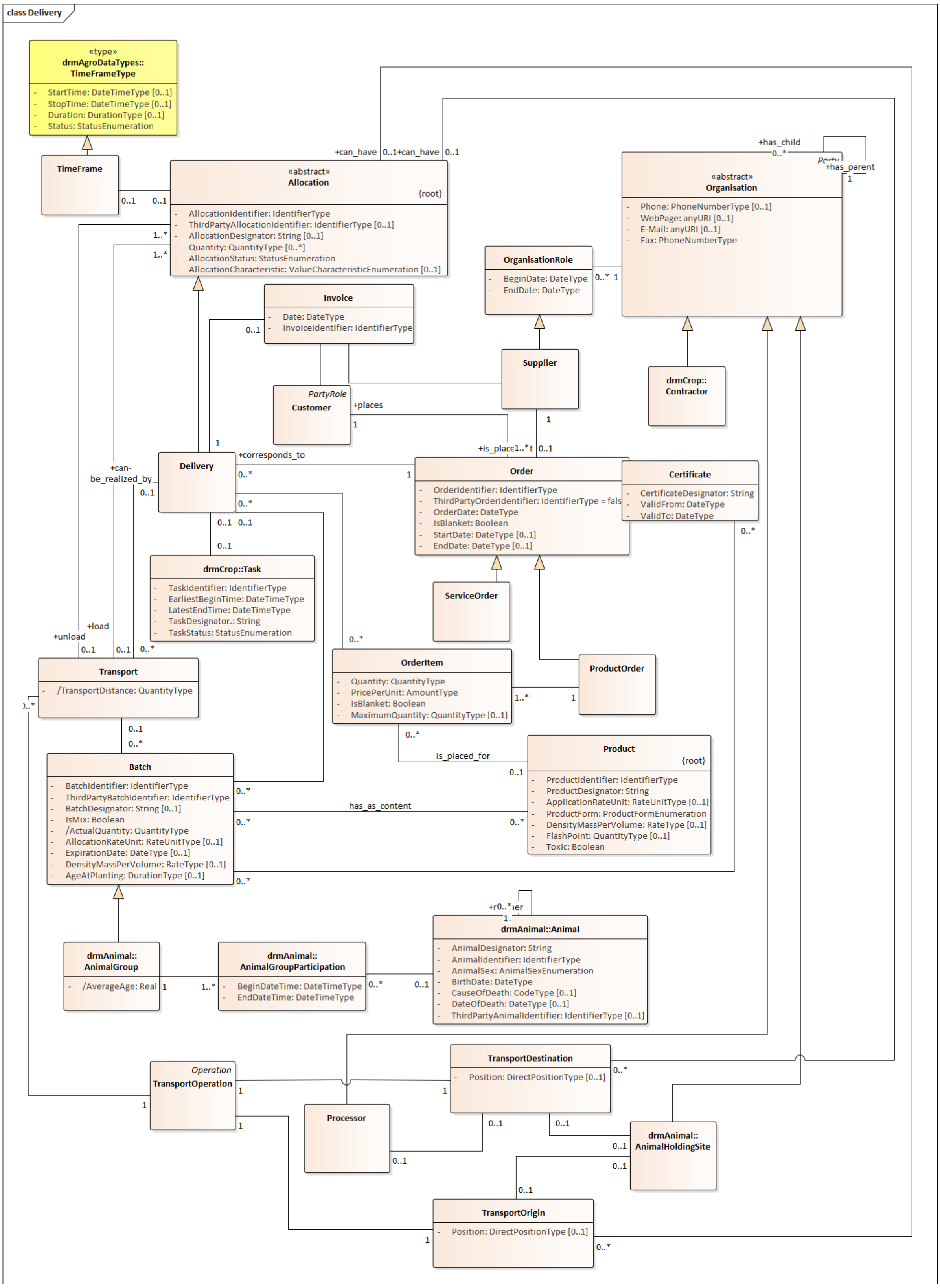


Figure 12: Delivery diagram in the drmAgro package of the Domain Model Agro (ftp://pragmaas.com/rmCrop/rmAgro_SNAPSHOT/).

When upscaling, the use cases could benefit from the following knowledge in rmAgro

- When use cases are upscaling a need to register more details of a farm and its surrounding could arise. The classes **Organisation** and **Party**, of which **AnimalHoldingSite** is a subclass, could be used as inspiration to elaborate on all kind of details in relationships with different roles of Persons. Note that rmAgro also recognise a **Farm** which is also a subclass of **Organisation** (like **AnimalHoldingSite**). **Farm** is used in the arable domain and the equivalent of **AnimalHoldingSite** in animal husbandry. It is already very well elaborated in rmAgro.
- The more detailed and normalised model of rmAgro enables easier track of all the changes occurring in time, e.g. the replacement of an ear tag of an animal. Figure 7 shows the **PhysicalIdentification** class between Animal and **PhysicalIdentifier** allowing historical capturing of associations between **PhysicalIdentifier** (e.g. ear tag) and an **Animal**. Another example is depicted in Figure 9 where **AnimalAllocation** and **AnimalGroupAllocation** are defined as separate classes. That way all historical allocations of **Animal** to animal housing like **Pen** and **Compartment** and of **Animal** to **AnimalGroup** can be captured. If the use case has an overview of all the changes and animal details to keep track of that needs to be registered, they could have a look at rmAgro to get a head start of the modelling.

2.4.2. Mapping use case data models on rmAgro Sensors

2.4.2.1. The Modelling of Sensors in rmAgro

In agriculture, sensors are used to observe status, behaviour and/or conditions in crop cultivation and animal husbandry. They are mostly used in precision agriculture. In this paragraph the modelling of sensors in rmAgro is explained.

Basically, a sensor can be mounted on equipment including drones, placed in a building, installed on a field, or attached to an animal or a human. Those sensors can vary from very simple to being part of complex systems, like a (mobile) observation platform or an implement (for example sprayer boom). When a sensor is part of a sensor system that system can have one or more sections on which one or more sensors are mounted. As a result of this multifaceted sensor system it might be very complex to determine the observation boundaries of a sensor.

Figure 13 shows the most relevant elements of Sensors in rmAgro[25]. As can be seen from the figure (black boxed area), rmAgro recognizes a **Sensor** as being part of a **SensorSystem** and as being a **Component** it might be part of **Equipment**. The Sensor in rmAgro is defined as a device that can measure the value of physical quantities. As said before, the determination of the observation boundaries might be complex and furthermore the equipment might need calibration which also can involve a complex process. The reference model rmAgro support a detailed registration of this process, as depicted by the classes in the blue boxed area of the figure. Class **ParameterFit** represent a procedure to for instance calibrate a scale or to determine the parameters of a semi-variogram. It uses a certain **Algorithm** in a **DataProcess** to determine **ParameterValue**'s. The **DataProcess** may need one or more **DataSets**, e.g. in case of the calibration of a scale a series of measured weights and a series of real weights are needed. Elements of the blue boxed area are general entities. They are

[25] The open issue list on rmAgro contains a task to compare it to OGC's sensorML (https://www.ogc.org/standards/sensorml). When applicable rmAgro might adapt to some insights of this model.

commonly used in rmAgro in several sub domains to specify all kind of properties, e.g. properties of a **Batch,** a **Container** or a **ProductSpecification.**

DataSet, PropertyVariable and **PropertyValue** with its association with **TimeFrame** can be used to capture all kind of measurements of a **Sensor** as illustrated in Figure 14. As can be seen from Figure 15 and Figure 16 the Sensor and Equipment has already associations with other subdomains (i.e. the Animal and the Task and Operation subdomains). From Figure 17 it can be seen that a link to the Inventory sub domain can be very easy established.

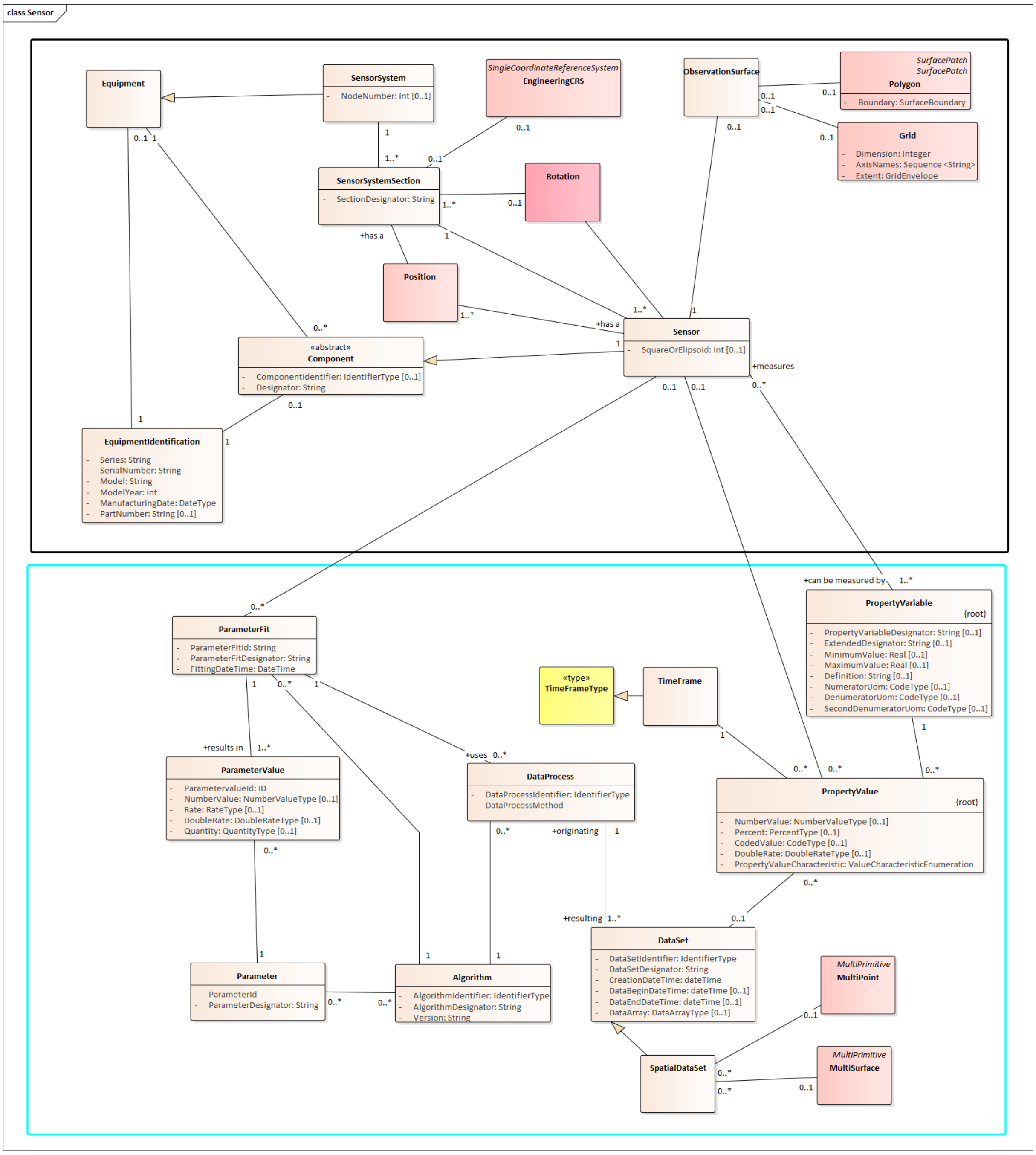


Figure 13: Modelling of equipment and sensors in the rmAgro reference model.

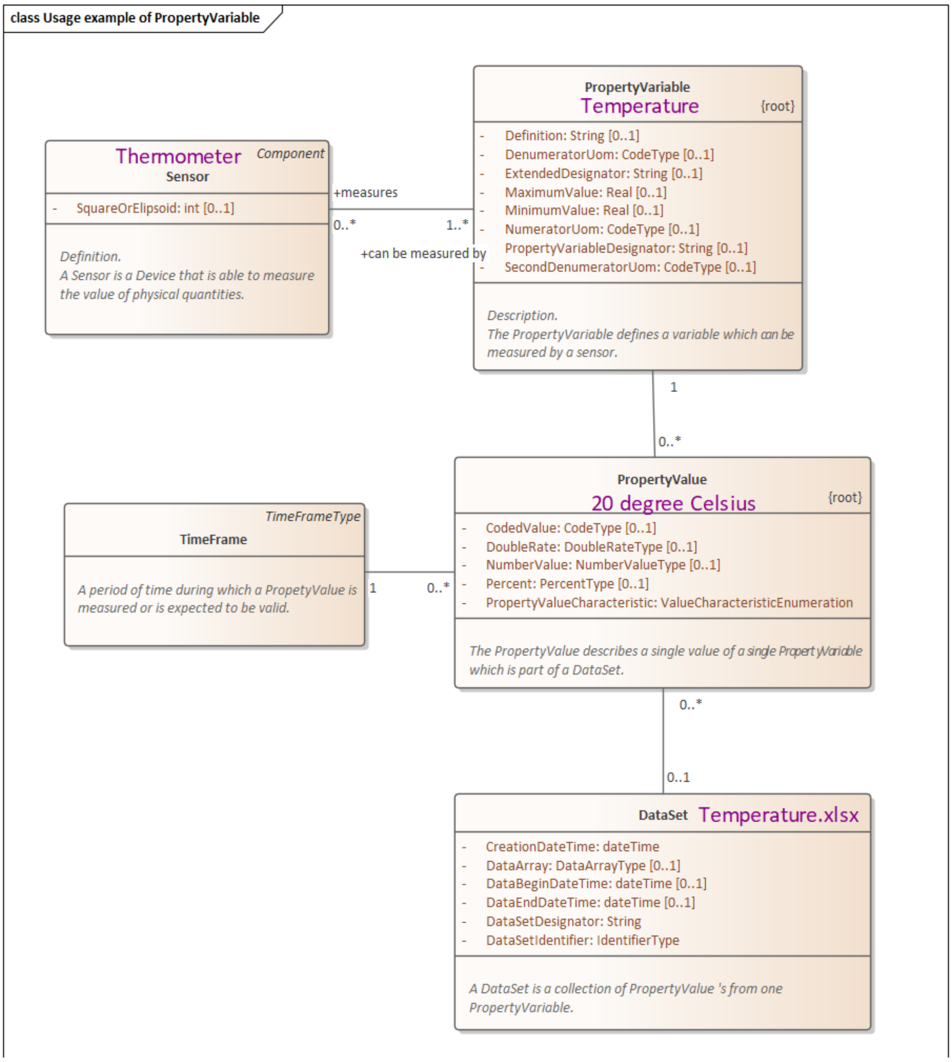


Figure 14: Example of the usage of PropertyVariable and PropertyValue.

In Figure 15 the relation between the Sensor and Observation domain (black and blue boxed) and the Animal domain (green boxed) is depicted. To relate the Sensor domain to the Animal domain the **ObservationSurface** of the Sensor domain is related to a **Compartment** in an **AnimalHouse**. If the ear tag also contains Sensors, e.g. to register the activity of an animal, the ear tag is a **SensorSystem** related to the **Animal** (i.e. yet another link between the two domains). Also when a sensor is mounted on a leg of an animal or any other way to bind on animals, like for example a bolus, it is a sensor system, which in most of those cases for animal husbandry will have one section with one or more sensors. Details of the Animal domain in rmAgro are described in paragraph 2.4.1.1.

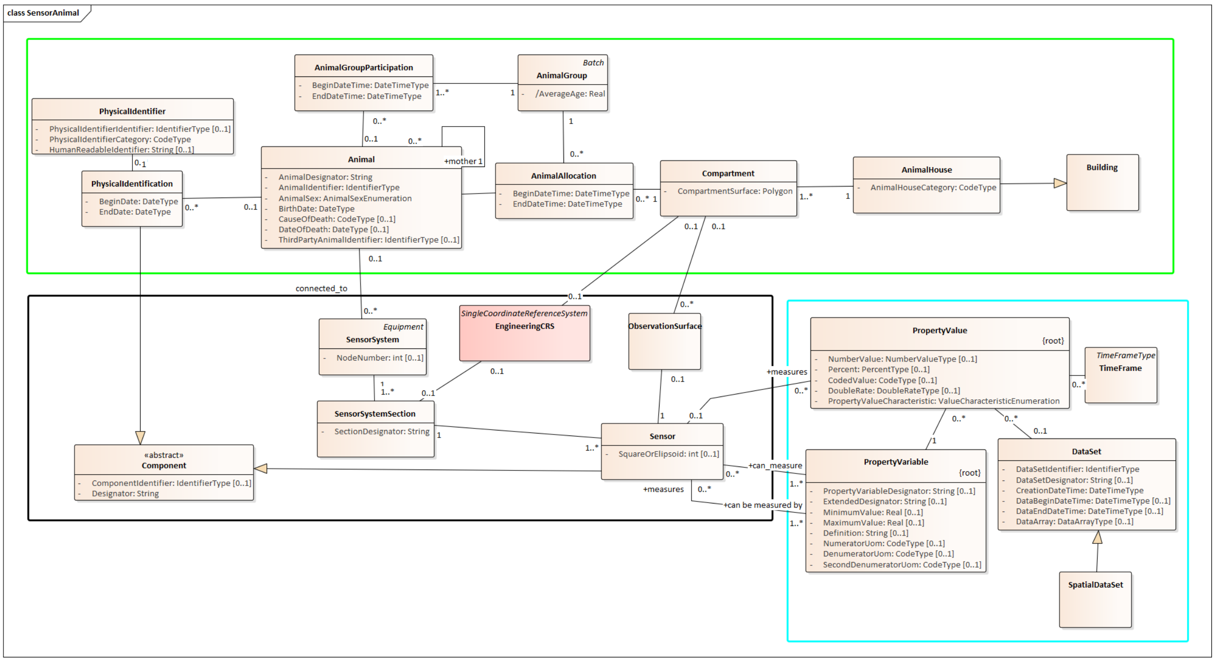


Figure 15: Relations between the Sensor and Observation domain and the Animal Domain.

In Figure 16 the relations between the Sensor and Observation domain (black and blue boxed), the Task and Operation domain (yellow boxed) and the Animal domain (green boxed) in rmAgro are depicted. **Tasks** and **Operation** (yellow boxed area) can be performed concerning an **AnimalGroup** (green boxed area) since a generalisation of **AnimalGroup** is **Batch**. An **Operation** is part of a **Task** and the **Task** describes the **WorkerAllocation** and also the **EquipmentAllocation**. Example: A worker transporting chickens (i.e. **Task**) could wear a bracelet (i.e. **Equipment/SensorSytem** with a **Sensor**) to measure the amount and severity of the movement of the chickens during truck loading (**Operation**) as a possible contributor to the stress level of the chickens (**AnimalGroup**). The measured values by the **Sensor** can be registered as **PropertyValue** of a certain **PropertyVariable**. The **PropertyVariable** dictates the unit the measurement is measured in. **PropertyVariable** is also used to specify what the sensor is capable of.

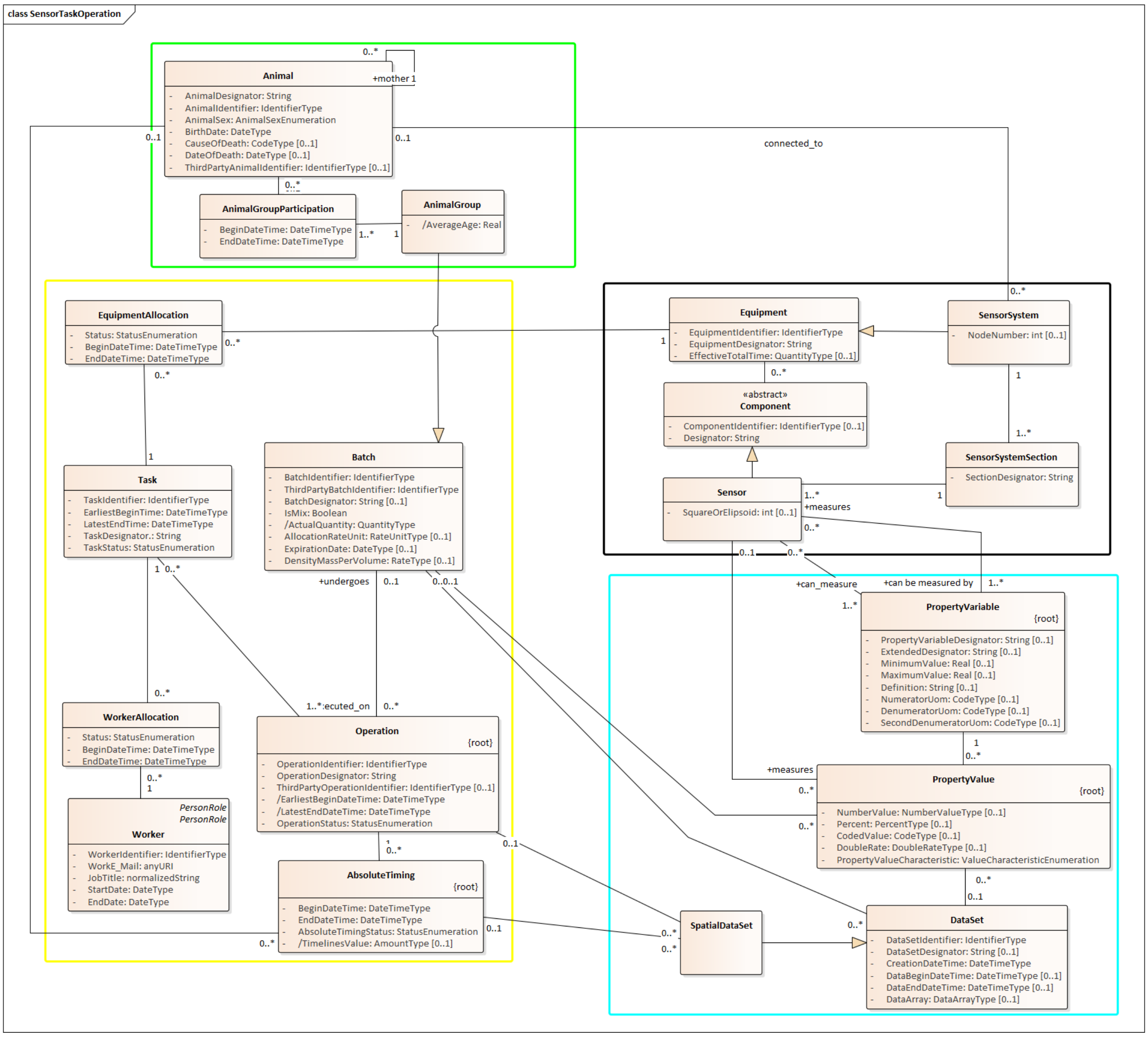


Figure 16: Relations between the Sensor and Observation domain and the Task and Operation domain in rmAgro.

2.4.2.2. *Comparison of rmAgro sensor data model to the meat trial data models*

All meat trial use cases use sensor observations in IoT solutions, although UC5.3 does not obtain the observations directly from an IoT device, but through an intermediate data storage which is fed by other IT solutions. To compare the meat trial data models with the sensor part of the rmAgro model we list the measurements observed by the use cases, the location where they were observed and the animals that were subject to these measurements.

For simplicity in the comparison, we don't consider the complexity of sensor systems and the positions of sensors within the sensor system. We also assume the sensor has an observation surface which is known and wouldn't have to be determined anymore. In general, the use cases provided us only with an overview of the data models they used, so this kind of details are not even known in this comparison. For our purpose, i.e. detecting similarities between data models of different use case, we don't need this level of detail as it would complicate the comparison without adding more insights. However, it is good to know a reference model like rmAgro does support this complexity.

Table 5: Mapping of use case sensor measurements to rmAgro.

Use Case	Measurement Level	Measurements	Use Case Data Models	RmAgro comparable classes
5.1 Pig Farm Management	Animal Housing	Temperature Humidity Luminosity CO2	Building Compartment Pen See also Table 7 and Table 8 in the Appendices	**PropertyValue**'s and **PropertyVariable**'s of a Sensor for a certain **ObservationSurface** in a **Compartment/Pen** of an **AnimalHousing**. To better understand the usage of **PropertyVariable** and **PropertyValue**'s, see the example Figure 14
5.1 Pig Farm Management	Animal/Animal Group	WaterConsumption FoodConsumption AvgWeight AvgGrow weightStDev	Pen Compartment Pig See also Table 7 and Table 8 in the Appendices	**PropertyVariable**'s and **PropertyValue**'s of a **Sensor** for a certain **VerticalLayer** of a **Zone** in a **Compartment/Pen** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation .** An **AnimalAllocation** can have registered an **AnimalGroup** or individual **Animals.** To better understand the usage of PropertyVariable and PropertyValue, see the example Figure 10

Use Case	Measurement Level	Measurements	Use Case Data Models	RmAgro comparable classes
5.2 Poultry Chain Management	Outside Weather at the location	Precipitation relativeHumidity skyDescription temperature windDirection windSpeed	WeatherPrediction-Observation AirQualityObserved (contains some outside measurements) SensorAirQuality-Observed See also Table 10 in the Appendices	**PropertyVariable** with **PropertyValue**. The predictions and outside weather measurements can be captured in mentioned classes and can be related to a **SensorPosition**. No direct association between weather **PropertieValue**'s and **Animal/AnimalAllocation** is needed, because they can be associated by mapping geocoordinates and timestamp (**SpatialDataSet**). To better understand the usage of PropertyVariable and PropertyValue, see the example Figure 14
5.2 Poultry Chain Management	Animal Housing	Airquality temperature CO2 NH3	AirQualityObserved CurvesIOfBreeding See also Table 10 in the Appendices	**PropertyValue**'s and **PropertyVariable** of a **Sensor** for a certain **VerticalLayer** of a **Zone** in a **Compartment/Pen** of an **AnimalHousing**
5.2 Poultry Chain Management	Animal Group/Animal	Gas consumption Feed consumption Water consumption Weight	FarmPopulation-Observed (e.g number of animals) FarmConsumptionObserved PoultryWeight-Observed See also Table 10 in the Appendices	**PropertyValue**'s and **PropertyVariable** of a **Sensor** for a certain **VerticalLayer** of a **Zone** in a **Compartment/Pen** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation ,** see also Figure 9. The food consumption in UC5.2 is indirectly measured by measuring the feed silo weight. In rmAgro, **Silo** has the attribute *TareWeigh* that could be used (see Figure 11)
5.3 Meat Transparency and Traceability	Animal Housing	Temperature CO2% Ammonia% Luminosity Noise *Airconditioning status*	Environment Event (Why part), see also Figure 24 in the Appendices	**PropertyValue**'s **and PropertyVariable** of a **Sensor** for a certain for a certain **VerticalLayer** of a **Zone** in a **Compartment/Pen** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation ,** see also Figure 9.**I.**

Use Case	Measurement Level	Measurements	Use Case Data Models	RmAgro comparable classes
5.3 Meat Transparency and Traceability	Animal	Feed intake Water intake and flow rate Animal weight	Growth Event (Why part), see Figure 21	**PropertyValue**'s of a **Sensor** for a certain **ObservationSurface** in a **Compartment** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation,** see also Figure 9.
5.4 Decision-making Optimisation in Beef Supply Chain	Animal	Feed intake Animal weight and growth	Animal model, see Table 11	**PropertyValue**'s of a **Sensor** for a certain **ObservationSurface** in a **Compartment** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation,** see also Figure 9.
5.5 Feed Supply Chain Management	Silo	Weight Content Silo Humidity Volumetric Content Silo Temperature	Measures for feed consumption, see Figure 29	Silo is a **ContainerCategory** of **Container** in the Inventory sub model of rmAgro (see also Figure 11) which now has an association with **SensorSystem** (see also Figure 17). The measurements are now **PropertyValue**'s of a **Sensor** in the **SensorSystem** of the **Container**.
5.5 Feed Supply Chain Management	Equipment	Battery Voltage Accelerometer Solar Voltage	Measures for device lifespan, see Figure 29	**PropertyValue**'s of a **Sensor** in the **SensorSystem.** A Sensor is a **Component** of **Equipment**.
5.6 Interoperable Pig Health Tracking	Animal Housing	CO_2 Ammonia Temperature Humidity Dust particles	AirQualityObserved, see Figure 37	**PropertyValue**'s of a **Sensor** for a certain VerticalLayer **of a** Zone in a **Compartment/Pen** of an **AnimalHousing** which has during a specific time period an **AnimalAllocation**, see also **Figure 9.**
5.6 Interoperable Pig Health Tracking	Animal	Heart rate Activity Temperature	PigHealthGWObserved PigHealthObserved, see Figure 38 and Figure 39	**PropertyValue**'s of a **Sensor** in a certain **SensorSystemSection** of a **SensorSystem** related to an **Animal.**

2.4.2.3. Summary and Recommendations

In Table 5 a comparison of the use case models with the rmAgro reference model is presented. As can be expected (see introduction to paragraph 2.3.2) the use cases don't detail their data model as much as is done in rmAgro. We also see that all use cases, being IoT solutions, make use of measurements, which can be mapped on rmAgo's sensor and equipment sub domain. Therefore, rmAgro might be used to act as a common vocabulary to enhance interoperability between the use cases. This also might be true for the Animal and Animal Housing part in rmAgro although rmAgro should incorporate some knowledge from the use cases, see paragraph 2.4.1.3.

We also saw that not always a direct link is needed in a data model to connect sub domains, because associations also can be made based on location/position (geo coordinates) and timestamp like modelled in rmAgro class **SpatialDataSet**. This is the case for the weather predictions in for instance UC5.2.

From the mapping in the previous section we learned rmAgro can incorporate the following knowledge from the use case models:

- rmAgro should add an association between the Inventory sub domain and the Sensor sub domain. UC5.5 uses sensors to measure the amount of feed in a Silo. rmAgro not yet included IoT solutions in the Inventory sub domain, but an association could be added very easily as depicted in Figure 17. As of November 2020 this association is indeed included in rmAgro.
- As a result of the knowledge obtained from the meat trial use cases also a link between Animal and SensorSystem is added to the rmAgro model.
- In rmAgro Operation is used instead of the widely used term Observation. It could be considered to add Observation as a sub class to Operation to ease recognition by users of rmAgro.

When upscaling, the use cases could benefit from the following knowledge in rmAgro

- By adopting rmAgro's separate sensor and / or sensor system solution, the use cases can for example also support sensor replacement or movements, or even maintenance. The aforementioned events may result in differences in measurement accuracy and are therefore worth recording.
- By adopting the distinction between PropertyValues and PropertyVariables the use cases can prescribe the units of the captured measurements by defining them in PropertyVariable. In the current data models it is often not very clear what unit is used. This can easily lead to misinterpretations while using this data for analysis.
- For food and water intake as well as the average weight and growth and its standard deviation some calculation needs to be done and sometimes more than one observation is needed. rmAgro provides a model which also accounts for details of these determinations. When in future use cases have a need to register determination details as well they could benefit from the way rmAgro addresses this topic.
- Reference model rmAgro also links several sub domains which can be of use for the meat trial use cases:
    - Link between animal/animal housing and sensors, see Figure 15
    - Link between animal/animal group, Sensor domain and the Task and Operation domain, see Figure 16

- Link between the inventory sub domain (which include Silo) and the Sensor sub domain, see Figure 17

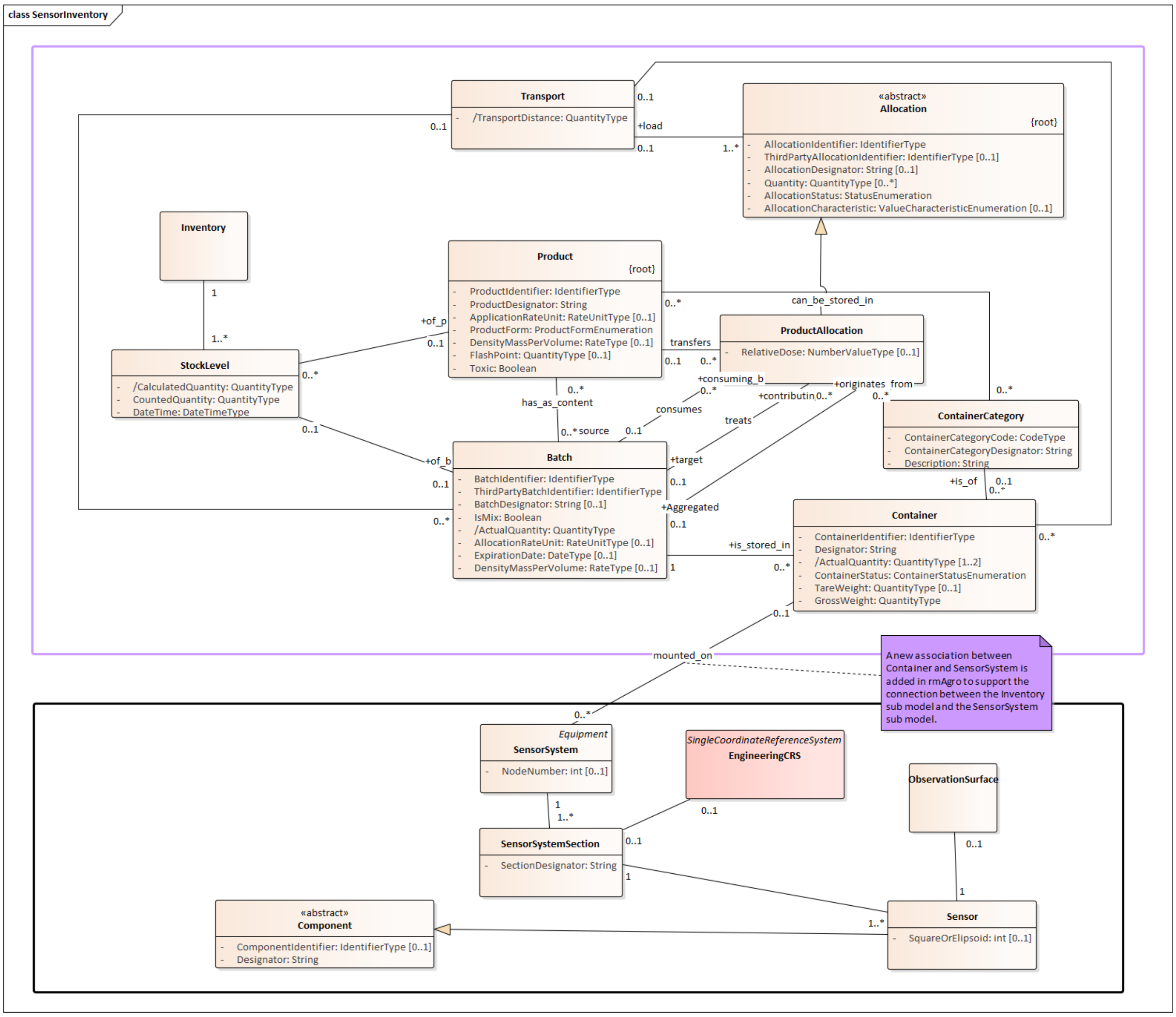


Figure 17: New link between the Inventory sub model and the Equipment and Sensor sub model.

3. Conclusions and Recommendations

In chapter 2 an analysis was made of the data models of the meat trial use cases in order to look for synergies, common challenges and ways to improve interoperability.

Conclusions:

- A standardized glossary and data dictionary, like initiated for UC5.1 and UC5.2, proved to be a tremendous help in understanding the data. However, these glossaries and data dictionaries were not set up at the beginning of the use cases, which made it very difficult to create one and get content of a good quality later on.
- Different types of data modelling were used, depending on their specific purpose and their chosen technical solutions. We can basically divide them in two main categories: event modelling and entity-based modelling. Depending on the kind of solution the use case has to provide, event modelling or entity-based modelling might be the better choice. Entity based models are well suited to capture all details of a certain domain of interest. Event models enable to capture the history of a certain object. For the animal and meat supply chain this can be locations, chain of custody, medications, transportations, slaughtering and processing after slaughtering. Data as part of an event model also could be traced back to data definitions in an entity-based model, as shown in the proof of concept ("OLIOT-Gateway").
- To monitor animals in the meat chain the correct identification of an animal is crucial. The registration of devices like ear tags, neck collars and/or bracelets to identify animals is necessary to relate sensor measurements to animals. Sometimes these devices need to be replaced or are reused when an animal dies or is slaughtered. In general, a farmer has numerous animals and those animals are reared far from the farmer's registration system. Therefore, the correct registration of an animal with its physical identifier device is a real challenge.
- There is a need to use predefined lists of values to register certain characteristics. In order to improve interoperability, it is desirable that use cases can refer to standard lists.
- Domain specific sub models like FIWARE Smart Data models and domain specific sub models in rmAgro could provide new initiatives with a head start of their data model to address their use case needs. A canonical model like rmAgro may also help to harmonize the integration of sub data models, as we saw in sensor modelling and animal husbandry.
- A key value of interoperability is the benefit of easy data sharing. Use of a common vocabulary by all solutions would facilitate this. rmAgro might be used to act as a common vocabulary to enhance interoperability between the use cases. All use cases make use of measurements and need to register Animals or AnimalGroups in some kind of Animal Housing. This maps very well on rmAgro's sensor and equipment part and the Animal and Animal Housing part.
- Use cases implement their specific domain related terminology in order to be easy understandable by their stakeholders. This seems to be contradictory to the requirement of interoperability. Linking use case data to reference data in standardized models like rmAgro and FIWARE Smart Data Models by means of semantic web technology might be a solution. Same-as relations as specified in OWL and SKOS can be used to overcome the usage of different

terms in different perspectives and solutions. Currently a pilot is started to create RDF[26] output out of rmAgro. The RDF output could possibly be used to link terms from use case data models to rmAgro terms in a linked data solution and thus creating a common rmAgro vocabulary. FIWARE Smart Models also support NGSI-LD format, a linked data format which makes it possible to link terms to comparable rmAgro terms using aforementioned same-as relations.

Recommendations:

- Ask new initiatives to create a glossary and data dictionary as part of their work and provide them with a desired format and guidelines to do so. Re-use what is already in place by other initiatives. This will improve the ability to plot use case data models on reference models in order to create a common vocabulary, hence improving interoperability.
- Ask use cases to evaluate and share their solution with regard to registering identifying devices with animals, so new initiatives can learn from it and may come up with a good general solution to this problem, especially when large volumes of animals are involved. A robust solution to this problem is a prerequisite for achieving traceability and transparency of meat. Regarding this topic also GS1's (S)GTIN standard might be investigated. The standard doesn't have allocation rules for livestock yet.
- Define an international agreed set of values and international recognized code lists for use in the animal domain. If (new) use cases can refer to these standards interoperability will be improved.
- It would help future initiatives to have a pre-selected choice of reference data models and expert data models to choose from to give them a head start in creating a data model. They could be published in the data marketplace[27] which is described in deliverable D3.6. Each of these pre-selected data models should have
    - A clear description of their scope and purpose and clear and easily accessible usage guidelines;
    - Easy and quick access to the parts of interest for new initiatives.
    - A version control system in place and information about changes and compatibility levels between versions of the (sub) model. After all, the pre-selected data models will be subject to continuous enhancements and improvements. In usages it should be known to which version of the model is referred.
- In order to enhance and improve pre-selected (reference) data models it would be helpful to incorporate knowledge from new initiatives into these data models. It is therefore desirable that collaboration of these new initiatives with administrators of such pre-selected (reference) data models will be made simple.
- The use of a common vocabulary by all solutions would facilitate interoperability, but solutions often have to adhere to stakeholders' terminology. Use case vocabulary (i.e. stakeholders' terminology) might be matched with common concept vocabularies provided by reference data models and expert data models using linked data and semantic web technology. In order to achieve such a linked data solution, it might be helpful if a reference data model could offer a

[26] https://www.w3.org/TR/rdf11-concepts/
[27] https://market.ioflab.opplafy.eu/

service to create RDF files from its (sub) models. This would enable easy linking to proprietary use case data models by means of semantic web technology.

4. Appendices

4.1. Glossary and Data dictionaries defined by use case 5.1

4.1.1. UC5.1 Domain Terms Glossary

Table 6: UC5.1 Domain Terms Glossary

Term	Definition	Synonyms	Homonyms	Preferred Term (Y/N)	Relationships to other elements	Range of values	Validation rules
Boar Taint	An unpleasant odour that can occur in the entire male carcasses of pigs. It is measured by human sniffers at the slaughter line of the slaughterhouse.			Y	is property of pig	It is evaluated with a score between 0 (normal smell) and 4 (strong boar taint) it can also be represented as 0 (no boar taint), 1 (boar taint) and 9 (not measured)	Can only have a value when the pig is male and when it is slaughtered It can only have the value 9 (not measured) or 0 (no boar taint) for female pigs
Building	A building where animals are housed.	Barn Shed		Y	(1) Belongs to Farm (2) Can (optional) be divided in Compartments (3) Can (optional) contain Pens (4) Can (optional) contain Pigs		
Carcass weight	The weight of a pig carcass at the slaughterhouse after being slaughtered and excess parts removed.			Y		numericals	positive values only
Company	A company owning pig farms			Y	Owns Farms		
Compartment	Artificial area in a building or department that is measured by certain sensors. A compartment is not necessarily a physical separator. It can be a department or a grouping of several pens within a department that are being measured by the same sensor.	Zone			(1) is part of a building (2) can contain Pens (optional) (3) can be part of another compartment (optional) (4) Can (optional) contain Pigs		
Crate	Individual housing area of breeding sows that may be used in some stages of the reproduction cycle. Gestation crate: for insemination and first stages of pregnancy. Farrowing crate: right before and after giving birth until the piglets are weaned.				related to a building or compartment, houses a sow (and optionally piglets)		

Term	Definition	Synonyms	Homonyms	Preferred Term (Y/N)	Relationships to other elements	Range of values	Validation rules
Department	Separated area in a pig barn building. Departments are physically separated from other departments by full walls and a door, making bio secure rooms. Pigs in a department are thus physically separated from pigs in other departments. The department can contain a number of pens or crates.	Housing			(1) is part of a building (2) can contain Pens or crates (optional)		
Farm	A physical address where pig farming occurs. The farm could have a name.				(1) Has Buildings (2) Can be owned/managed by a Company (or Companies) - is this relationship mandatory? Yes		
Fasting time	The amount of time before the slaughter in which the pig isn't fed. Fasting pigs before the slaughter reduce mortality during transport to the abattoir. It also reduces also the muscle PH drop, and therefore leads to a higher quality of meat. This amount of time is measured once per Slaughter Batch				Relates to a group of pigs to be transferred to the slaughterhouse (Slaughter Batch)	time, from (tranport + waiting time) to several days	positive values only
Feed Consumption	Amount of food a pig or a group of pigs have eaten. It is usually in kg, but this can be kilogram dry feed, liquid feed or meal, depending on what kind of feed is delivered to the pigs and what reference is wanted by the farmer.	Feed intake, Feed delivered, Feed dosed		Y	Relates to a Pig or a group of pigs (pen, department, compartment, feed tray) over a certain period of time	numericals	positive values only
Feed Type	Type of feed fed to the pigs. On farm-level or building level this can be the type of feed that is delivered (dry feed, liquid feed, meal), this is fixed in time, as specific feeding systems can only handle one type of feed. On compartment and feed tray level, this refers to the phase and composition of the feed that is delivered to the pigs. This is a function of the age of the pigs and can involve a merging phase in which two feed types are given at the same time. When available, this is registered for every feeding. Otherwise a standard table with feed type compositions in function of the days of the feeding curve is used.				1) Is a property of a farm or building 2) is property of a feeding occurance		
Genetics					is property of Pig or group of pigs		
Growth	The growth in weight of a pig or mean growth of a group of pigs in a certain amount of time				is related to Pig or group of pigs over a certain period of time	numericals	positive and negatives
Health treatment	A health treatment of the pig or group of pigs at the farm, dose and used product are registered				is related to Pig or group of pigs on a specific day		
Pen	Fenced area in a building or department or outside housing a group of animals. Animals in a pen can move and interact freely. Pens are often not completely separated from each other (half walls, iron bars, fences,…), making it possible that animals from neighbouring pens can see/touch/smell each other, but they are physically limited to their own pen and cannot move from one pen to another.				1) is related to a Building or a Compartment or Farm (in case of outside farming) 2) Can Contain (optional) Pigs		

Term	Definition	Synonyms	Homonyms	Preferred Term (Y/N)	Relationships to other elements	Range of values	Validation rules
Pig	General term to identify a pig. More specific terms can be used when necessary: - piglet: young pig, usually from birth to start of the fattening period - fattening pig: A pig raised for the meat industry - gilt: female pig that has not yet given birth (can be a starting breeding sow or a female fattening pig) - (breeding) sow: A female pig used for breeding - boar: uncastrated male pig, can be used for breeding or can be a uncastrated male fattening pig - barrow: castrated male pig, usually for fattening		The term Pig can have different meanings in different contexts. For this use case, we only have pigs for the meat sector (fattening pigs and piglets, no distinction is made)		(1) Is housed in a Building and/or a Compartment and/or a Pen (2) Belongs (optional) to a Slaughter Batch		
Pig Weight	The weight of a pig at a certain date and time.				is related to Pig at a certain time	depends on age of pig	positive values only
Sensor	Equipment measuring a particular property of the housing (Pen, Compartment, Building) where pigs are raised for the meat industry are held. What details are registered about a sensor? E.g. identification number, name, location, sensor type, brand, manufacturer, operation time, software/hardware version, maintenance etc.				is related to a Building or a Compartment or a Pen or a Pig		
Slaughter Batch	A group of pigs from a certain farm that were sent to the slaughterhouse together. As slaughter batches are a mix of pigs from several departments it is impossible to link slaughterhouse results back to farm batch characteristics, unless the previous location of the pigs in the barn is linked to the identification numbers of the pigs that are send to the slaughter (which is laborious and requires a certain kind of organisation on farm, not all farmers do this)				consists of Pigs		
Slaughterhouse	The place a pig is slaughtered and processed into meat	Abattoir			is place of slaughter of a Slaughter Batch		
Slaughterhouse WaitingTime	The time the pigs are held at the slaughterhouse, before they are slaughtered. This time is measured once per SlaughterBatch.				attribute of a Slaughter Batch	time, usually not more than 24h	positive values only
TransportTime	The amount of time the transportation of the pigs from the farm to the slaughterhouse took. This amount of time is measured once per Slaughter Batch				attribute of a Slaughter Batch	time, usually less than 8 hours	positive values only
Water Consumption	The amount of water that was consumed by a pig or a group of pigs	Water Intake			Relates to a Pig or group of pigs over a certain period of time	depends on group size	positive values only
WaterFlow	Raw data from a waterflow sensor connected to a water pipe. It gives the momentary waterflow rate. Data processing is needed to calculate water consumption from these values.				Is related to Pen or Compartment of Building at a certain timestamp	depends on maximal flow rate of the water pipe	positive values only

4.1.2. UC5.1 Database in IOT Platform

Table 7: Data dictionary UC5.1 Database in IOT Platform

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
Farm	farmid	unique identifier of a farm	String		How do you uniquely identify a farm? A UUID4 is generated and added to the word "Farm-" to create a unique identifier for the farm	Y
	name	name of the farm	String			Y
Building	BuildingID	unique identifier of a building	String		How do you uniquely identify a building? A UUID4 is generated and added to the word "Building-" to create a unique identifier for the building	Y
	name	name of the building	String			Y
	lastUpdateTimestamp	Date and time at which the measurements in the building were taken What unit is used? it is a UNIX timestamp, thus it is in seconds.	int			Y
	farmId	unique identifier of a farm	String		It must be an existing farmId	N
	companyId	unique identifier of a company	String		It must be an existing companyId	N
	currentTemperature	The last received temperature value (to distinguish it from an historical measures). What is the unit used for this measure? It is the one used by the system used by the farm. Usually, it is in Celsius degree	Double (2 decimal)			N
	currentHumidity	The last received humidity (to distinguish it from an historical measures). It is a quantity representing the amount of water vapour in the atmosphere in the building What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage.	Double (2 decimal)			N
	currentLuminosity	The last received luminosity (to distinguish it from an historical measures). It is the brightness of a light source of a certain wavelength in the building. What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	currentCO2	The last received CO2 concentration (to distinguish it from an historical measures). What is the unit used? It is the one used by the system used by the farm.	Double (2 decimal)			N
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Compartment	compartmentid	unique identifier of a compartment	String		How do you uniquely identify a compartment? A UUID4 is generated and added to the word "Compartment-" to create a unique identifier for the compartment	Y

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	lastUpdateTimestamp	Date and time at which the measurements in the compartment were taken What unit is used? it is a UNIX timestamp, thus it is in seconds.	int			Y
	farmid	unique identifier of a farm the compartment belongs to	String		It must be an existing farmId	N
	buildingid	unique identifier of a building the compartment is located in	String		It must be an existing buildingId	N
	companyId	unique identifier of a company	String		It must be an existing companyId	N
	parentCompartmentid	unique identifier of the compartment where this compartment is a part of. It is used only when a compartment contains other compartments	String		It must be an existing CompartmentId	N
	name	name of the compartment.	String			N
	currentTemperature	The last received temperature value (to distinguish it from historical measures). What is the unit used for this measure? It is the one used by the system used by the farm. Usually, it is in Celsius degree	Double (2 decimal)			N
	currentHumidity	The last received humidity (as to distinguish it from historical measures). It is a quantity representing the amount of water vapour in the atmosphere in the building What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage.	Double (2 decimal)			N
	currentLuminosity	The last received luminosity (as to distinguish it from historical measures). It is the brightness of a light source of a certain wavelength in the building. What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	currentCO2	The last received CO2 concentration (as to distinguish it from historical measures). What is the unit used? It is the one used by the system used by the farm.	Double (2 decimal)			N
	currentNumAnimals	The last received number of pigs in the compartment (as to distinguish it from historical measures). The compartment is more or less a "zone" of the building. When it is possible a wall or a panel is used to divide the monitored area.	Double (2 decimal)			N
	currentAvgWeight	The last received avg weight of the pigs in this compartment (as to distinguish it from historical measures). In which unit is this measured? It is the one used by the system used by the farm. Usually, it is in Kg	Double (2 decimal)			N
	currentAvgGrowth	The last received avg growth in weight of the pigs in this compartment (as to distinguish it from historical measures).	Double (2 decimal)			N
	currentweightStDev	The last received standard deviation associated to the average weight of the pigs/piglets contained in the compartment (as to distinguish it from historical measures).	Double (2 decimal)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	currentWaterConsumption	The last received the total amount of water that came out from the tap or taps in the compartment (as to distinguish it from historical measures). How is this measured? Through flowmeters and specific structure to let only one pig at a time drink in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured.	Double (2 decimal)			N
	currentFeedConsumption	The last received total amount of food that has been eaten from the feeding station(s) in the compartment (as to distinguish it from historical measures). How is this measured? Through feed intakes and specific structure to let only one pig at a time eat in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured Is this the total amount of food that is measured between two compartment measures? It depends on the system already installed in the farm. What unit is used? The one used by the system already installed in the farm. No translation is made.	Double (2 decimal)			N
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Pen	penId	Unique identifier of the Pen	String		How do you uniquely identify a pen? A UUID4 is generated and added to the word "Pen-" to create a unique identifier for the pen	Y
	lastUpdateTimeStamp	Date and time at which the measurements in the Pen were taken Which unit is used? Seconds	int			Y
	farmId	Unique identifier of the Farm the Pen is located in. This attribute is only added in the description, not in the example message. Is it part of this entity? Yes. Even though it is a replication we preferred to reinsert this field	String		It must be an existing farmId	N
	buildingId	Unique identifier of the Building the Pen is located in	String		It must be an existing buildingId	N
	compartmentId	Unique identifier of the Compartment the Pen is located in. This attribute is only added in the example message, not in the description. Is it part of this entity? Yes. Even though it is a replication we preferred to reinsert this field	String		It must be an existing CompartmentId	N
	companyId	unique identifier of a company	String		It must be an existing companyId	N
	currentTemperature	Last received temperature of the Pen (as to distinguish it from historical measures). What is the unit used for this measure? It is the one used by the system used by the farm. Usually, it is in Celsius degree	Double (2 decimal)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	currentHumidity	Last received Quantity representing the amount of water vapour in the atmosphere in the pen (as to distinguish it from historical measures). What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage	Double (2 decimal)			N
	currentLuminosity	Last received brightness of a light source of a certain wavelength at the pen (as to distinguish it from historical measures). What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	currentWaterComsumption	Last received total amount of water that came out from the tap or taps in the pen (as to distinguish it from historical measures). How is this measured? Through flowmeters and specific structure to let only one pig at a time drink in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured. Is this the total amount of water that is measured between two pen measures? It depend on the farm. What unit is used? It depends on the farm systems.	Double (2 decimal)			N
	currentFeedConsumption / currentFoodFlow	Last received total amount of food that has been eaten from the feeding station(s) in the pen (as to distinguish it from historical measures). How is this measured? Through feed intakes and specific structure to let only one pig at a time to eat in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured. Is this the total amount of food that is measured between two pen measures? It depends on the farm What unit is used? It depends on the farm What name must be used for this attribute (description and example message differ with regard to this name)? currentFoodFlow is used to facilitate the work of the dashboard provider. The preferred name is "currentFeedConsumption"	Double (2 decimal)			N
	currentCO2	Last received CO2 concentration in the pen (as to distinguish it from historical measures). What unit is this measured in? It depends on the farm. Usually LUX	Double (2 decimal)			N
	currentNumAnimals / numPigs	Last received number of pigs/piglets contained in the Pen (as to distinguish it from historical measures). We assume a Pen is fenced, is that correct? Yes. numPigs is used to facilitate the work of the dashboard provider. The preferred name is "currentNumAnimals"	Double (2 decimal)			N
	currentAvgWeight	Last received average weight of the pigs/piglets in this Pen (as to distinguish it from historical measures). In which unit is this measured? It is the one used by the system used by the farm. Usually, it is in Kg	Double (2 decimal)			N
	currentAvgGrowth	Last received average growth in weight of the pigs/piglets in this pen (as to distinguish it from historical measures).	Double (2 decimal)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	currentweightStDev	Last received standard deviation associated to the average weight of the pigs/piglets contained in the Pen (as to distinguish it from historical measures).	Double (2 decimal)			N
	Sex	The sex of the pigs contained in the pen	String	Z/B or M/F		N
	deadAnimalsSinceDateOfArrival	Number of dead animals since the date of arrival	Double (2 decimal)			N
	arrivalTimestamp	Date and Time at which the pigs were inserted in the pen	Double (2 decimal)			
	additionalInfo	List of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Pig	PigID	Unique identifier of the Pig	String		How do you uniquely identify a pig? A UUID4 is generated and added to the word "Pig-" to create a unique identifier for the pig	Y
	penID	Unique identifier of the Pen the Pig is contained in	String			Y
	startTimestampMonitoring / startTimestampAcquisition	The timestamp of the moment the housing of the Pig in this Pen started.	int			N
	endTimestampMonitoring / endTimestampAcquisition	The timestamp of the moment the housing of the Pig in this Pen ended.	int			N
	farmId	Unique identifier of the Farm the Pig is housed in.	String		It must be an existing farmId	N
	buildingid	Unique identifier of the Building the Pig is housed in.	String		It must be an existing buildingId	N
	serialNumber	The serial number assigned to the pig by the farm	String			N
	lastUpdateTimestamp	Date and time at which the last taken measure (reported in current…) Which unit is used? seconds	int			Y
	currentWeight	The weight of the Pig at the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)			N
	currentTotalConsumedWater	The amount of water that was consumed between the moment the pig started to drink and the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)		Is e.g a minimum and/or maximum time duration applicable? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	currentTotalConsumedFood	The amount of food that was consumed between the moment the pig started to eat and the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)		Is e.g a minimum and/or maximum time duration applicable? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	currentTotalTimeConsumedWater	The amount of time the pig spends on drinking from the beginning to the end of a drink. In the example messages also a startTimestamp and an endTimestamp are stated. Should these attributes also be included for this entity? The startTimestamp and endTimestamp are not mandatory. Thus, this attribute is useful when they are not specified	Double (2 decimal)		Are there some rules applicable on what is counted as one drink? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	currentTotalTimeConsumedFood	The amount of time the pig spends on eating from the beginning to the end of a food portion/ food moment. In the example messages also a startTimestamp and an endTimestamp are stated. Should these attributes also be included for this entity? The startTimestamp and endTimestamp are not mandatory. Thus, this attribute is useful when they are not specified	Double (2 decimal)		Are there some rules applicable on what is counted as one food portion/ food moment? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	additionalInfo	List of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
PigStatus	pigStatusId	Unique identifier of the a measure related to a Pig	String		How do you uniquely identify a slaughtered pig? Is it still the same identification as the PigID when the pig was alive? A UUID4 is generated and added to the word "PigStatus-" to create a unique identifier for the slaughtered pig. No, it is not the same ID as Pig. There is only a link among them.	Y
	timestamp	The day and time of the status	int			Y
	Weight	The weight of the Pig at the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)			N
Consumption	consumptionId	Unique identifier of the Consumption	String		How do you uniquely identify a slaughterhouse? A UUID4 is generated to create a unique identifier for the consumption	Y
	Type	Type of consumption	String	water/food		N
	Amount	The revealed measure	Double (2 decimal)			N
	duration	In case we are storing a measure that cover an interval, this field contains the duration (it could be calculated from startTimestamp and endTimestamp when they are available)				
	startTimestamp	In case we are storing a measure that cover an interval, this field contains the startTimestamp				
	endTimestamp	In case we are storing a measure that cover an interval, this field contains the endTimestamp				
	Timestamp	Date and time at which the Pig measurements was taken. Which unit is used? seconds	Object (JSON)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
Slaughterhouse	slaughterhouseid	Unique identifier of the Slaughterhouse	String		How do you uniquely identify a slaughterhouse? A UUID4 is generated and added to the word "Slaughterhouse-" to create a unique identifier for the slaughterhouse	Y
	address	Postal addres of the Slaughterhouse? In our case the address is not completely useful, thus it is treated as a general information and we inserted it as a String. Each slaughterhouse can indicate it in its preferred format. Can it also be a Geo location? Not in our case	String			N
	name	Name of the Slaughterhouse	String			N
	additionalInfo	List of all raw values sent by the sensor / platform with all possible extra features that are not included in the main structure.If additional measurements are measured by sensors in the slaughterhouse, to which entity are these measurements related? Is that the pig?	Object (JSON)			N
Company	companyid	Unique identifier of the Company owning pig farms	String		How do you uniquely identify a company? A UUID4 is generated and added to the word "Company-" to create a unique identifier for the company Is no address required to uniquely identify a company? In our case it is not necessary	Y
	name	Name of the company	String			Y
Measure	measureId	Unique identifier of the Measure	String		How do you uniquely identify a slaughterhouse? A UUID4 is generated to create a unique identifier for the measure	Y
	Type	Type of consumption	String	temperature/humidity /luminosity/waterFlow /foodFlow/CO2		N
	Value	The revealed measure	Double (2 decimal)			N
	Timestamp	Date and time at which the Pig measurements was taken. Which unit is used? seconds	Object (JSON)			N

4.1.3. UC5.1 Orion Context Broker in IOT Platform

Table 8: Data dictionary UC5.1 Orion context broker data in IOT Platform

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
Farm	farmid	unique identifier of a farm	String		How do you uniquely identify a farm? A UUID4 is generated and added to the word "Farm-" to create a unique identifier for the farm	Y
	address	In use case 5.1 the address is not completely useful, thus it is treated as a general information and we inserted it as a String. Each farm can indicate it in its preferred format. It is not specifoed as a Geo location	String			N
	name	name of the farm	String			Y
	companyId/ownerCompany	It is the CompanyId that uniquely identifies the Company (i.e., Company-<UUID4>) What name must be used for this attribute (description and example message differ with regard to this name)? They have the same meaning. The name was changed for some farms to simplify the work of the dashboard provider that was readapting an existing software. Preferred name: companyId	String		How do you uniquely identify a company? For the Company entity a UUID4 is generated and added to the word "Company-" to create a unique identifier for the building. Thus here, we report the one that is related to the farm	N
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure and are necessary to look up the farm and related company in the database. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Building	BuildingID	unique identifier of a building	String		How do you uniquely identify a buildiing? A UUID4 is generated and added to the word "Building-" to create a unique identifier for the building	Y
	name	name of the building	String			Y
	lastUpdate	Date and time at which the measurements in the building were taken What unit is used? it is a UNIX timestamp, thus it is in seconds What name must be used for this attribute (description and example message differ with regard to this name)? In the Orion datamodel it is called lastUpdate	int			Y
	farmId	unique identifier of a farm	String		It must be an existing farmId	N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
	temperature	Temperature of the building The unit used for this measure is the one used by the system used by the farm. Usually, it is in Celsius degree.	Double (2 decimal)			N
	humidity	A quantity representing the amount of water vapour in the atmosphere in the building What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage	Double (2 decimal)			N
	luminosity	The brightness of a light source of a certain wavelength in the building What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	co2	The CO2 concentration in ? - what is the unit used? It is the one used by the system used by the farm.	Double (2 decimal)			N
	additionalInfo	List of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure and are necessary to look up the building and the related farm in the database. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Compartment	compartmentid	unique identifier of a compartment	String		How do you uniquely identify a compartment? A UUID4 is generated and added to the word "Compartment-" to create a unique identifier for the compartment	Y
	lastUpdate	Date and time at which the measurements in the compartment were taken What unit is used? it is a UNIX timestamp, thus it is in seconds	int			Y
	farmid	unique identifier of a farm the compartment belongs to	String		It must be an existing farmId	N
	buildingid	unique identifier of a building the compartment is located in	String		It must be an existing buildingId	N
	parentCompartmentid	unique identifier of the compartment where this compartment is a part of. It is used only when a compartment contains other compartments	String		It must be an existing CompartmentId	N
	name	name of the compartment.	String			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
		The compartment example message also contains a name. It is not in the description. Could a compartment have a name?				
	temperature	Temperature of the compartment. What is the unit used for this measure? It is the one used by the system used by the farm. Usually, it is in Celsius degree	Double (2 decimal)			N
	humidity	Quantity representing the amount of water vapour in the atmosphere in the compartment. What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage	Double (2 decimal)			N
	luminosity	The brightness of a light source of a certain wavelength at the compartment What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	co2	The CO2 concentration in the compartment What unit is this measured in? It is the one used by the system used by the farm.	Double (2 decimal)			N
	numAnimals	Number of pigs in the compartment Is the compartment divided by physical walls? The compartment is more or less a "zone" of the building. And, yes, when it is possible a wall or a panel is used to divide the monitored area.	Double (2 decimal)			N
	avgWeight	The avg weight of the pigs in this compartment In which unit is this measured? In kg? It is the one used by the system used by the farm. Usually, it is in Kg	Double (2 decimal)			N
	avgGrowth	The avg growth in weight of the pigs in this compartment	Double (2 decimal)			N
	weightStDev	The standard deviation associated to the average weight of the pigs/piglets contained in the compartment	Double (2 decimal)			N
	waterconsumption	The total amount of water that came out from the tap or taps in the compartment. It is measured through flowmeters and specific structure to let only one pig at a time to drink in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured.	Double (2 decimal)			N
	feedconsumption/outputFeed	The total amount of food that has been eaten from the feeding station(s) in the compartment	Double (2 decimal)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
		It is measured through feed intakes and specific structure to let only one pig at a time to eat in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured. Is this the total amoumt of food that is measured between two compartment measures? It depends on the system already installed in the farm. What unit is used? The one used by the system already installed in the farm. No translation is made				
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure and are necessary to look up the compartment an the related building and farm in the database. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Pen	penId	Unique identifier of the Pen	String		How do you uniquely identify a pen? A UUID4 is generated and added to the word "Pen-" to create a unique identifier for the pen	Y
	lastUpdate	Date and time at which the measurements in the Pen were taken	int			Y
	farmId	Unique identifier of the Farm the Pen is located in.	String		It must be an existing farmId	N
	buildingId	Unique identifier of the Building the Pen is located in	String		It must be an existing buildingId	N
	compartmentId	Unique identifier of the Compartment the Pen is located in.	String		It must be an existing CompartmentId	N
	temperature	Temperature of the Pen. What is the unit used for this measure? It is the one used by the system used by the farm. Usually, it is in Celsius degree	Double (2 decimal)			N
	humidity	Quantity representing the amount of water vapour in the atmosphere in the pen. What is the unit used? It is the one used by the system used by the farm. Usually, it is a percentage	Double (2 decimal)			N
	luminosity	The brightness of a light source of a certain wavelength at the pen. What unit is it measured in? It is the one used by the system used by the farm. Usually, it is in LUX	Double (2 decimal)			N
	waterFlow	The total amount of water that came out from the tap or taps in the pen	Double (2 decimal)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
		How is this measured? Through flowmeters and specific structure to let only one pig at a time drink in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured Is this the total amoumt of water that is measured between two pen measures? It depend on the farm What unit is used? It depends on the farm systems				
	foodFlow	The total amount of food that has been eaten from the feeding station(s) in the pen. How is this measured? Through feed intakes and specific structure to let only one pig at a time eat in ILVO farm, while a total amount for all the pigs in the pen/compartment is measured Is this the total amoumt of food that is measured between two pen measures? It depends on the farm What unit is used? It depends on the farm	Double (2 decimal)			N
	co2	The CO2 concentration in the pen	Double (2 decimal)			N
	numAnimals	Number of pigs/piglets contained in the Pen We assume a Pen is fenced, is that correct? Yes	Double (2 decimal)			N
	avgWeight	The average weight of the pigs/piglets in this Pen. In which unit is this measured? In kg? It is the one used by the system used by the farm. Usually, it is in Kg	Double (2 decimal)			N
	avgGrowth	The average growth in weight of the pigs/piglets in this pen	Double (2 decimal)			N
	weightStDev	The standard deviation associated to the average weight of the pigs/piglets contained in the Pen.	Double (2 decimal)			N
	Sex	The sex of the pigs contained in the pen	String	Z/B or M/F		
	deadAnimalsSinceDateOfArrival	number of dead animals since the date of arrival	Double (2 decimal)			
	arrivalTimestamp	Date and Time at which the pigs were inserted in the pen	Double (2 decimal)			
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure and are necessary to look up the pen and the related compartment, building and farm in the database. It is a JSON structure similar to this: { "temperature": 32, "humidity":42}	Object (JSON)			N
Pig	PigID	Unique identifier of the Pig	String		How do you uniquely identify a pig? A UUID4 is generated and added to the	Y

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
					word "Pig-" to create a unique identifier for the pig	
	penID	Unique identifier of the Pen the Pig is contained in	String			Y
	startTimestampAcquisition / startTimestampMonitoring	The timestamp of the moment the housing of the Pig in this Pen started. This atttribute is only added in the example message, not in the description. Is it correct to assume that this attribute is part of this entity? Yes! It is part of the entity. Furthermore, in the Orion data model the right name is startTimestampAcquisition. However, to maintain a retro-compatibility with the model we were using at the beginning of the UC, we insert the same value in both fields	int			N
	endTimestampAcquisition / endTimestampMonitoring	The timestamp of the moment the housing of the Pig in this Pen ended. This attribute is only added in the example message, not in the description. Is it correct to assume that this attribute is part of this entity? Yes! It is part of the entity. Furthermore, in the Orion data model the right name is startTimestampAcquisition. However, to mantain a retro-compatibility with the model we were using at the beginning of the UC, we insert the same value in both fields	int			N
	farmId	Unique identifier of the Farm the Pig is housed in. It is not mandatory, but it is an attribute we use to facilitate the work of the dashboard provider	String		It must be an existing farmId	N
	buildingid	Unique identifier of the Building the Pig is housed in. It is not mandatory, but it is an attribute we use to facilitate the work of the dashboard provider	String		It must be an existing buildingId	N
	serialNumber	The serial number assigned to the pig by the farm	String			N
	lastUpdate	Date and time at which the Pig measurements were taken. Which unit is used? seconds	int			Y
	weight	The weight of the Pig at the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)			N
	totalConsumedWater	The amount of water that was consumed between the moment the pig started to drink and the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)		Is e.g a minimum and/or maximum time duration applicable? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to	N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
					reveal anomalies and if we limit this value we could lose something	
	totalConsumedFood	The amount of food that was consumed between the moment the pig started to eat and the time of the measurement. What unit is used? It depends on the system already installed in the farm.	Double (2 decimal)		Is e.g a minimum and/or maximum time duration applicable? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	totalTimeConsumedWater	The amount of time the pig spends on drinking from the beginning to the end of a drink. In the example messages also a startTimestamp and an endTimestamp are stated. Should these attributes also be included for this entity? The startTimestamp and endTimestamp are not mandatory. Thus, this attribute is useful when they are not specified	Double (2 decimal)		Are there some rules applicable on what is counted as one drink? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	totalTimeConsumedFood	The amount of time the pig spends on eating from the beginning to the end of a food portion/ food moment. In the example messages also a startTimestamp and an endTimestamp are stated. Should these attributes also be included for this entity? The startTimestamp and endTimestamp are not mandatory. Thus, this attribute is useful when they are not specified	Double (2 decimal)		Are there some rules applicable on what is counted as one food portion/ food moment? It depends on the farms. Furthermore, we cannot limit the values sent through Orion or saved in the database. It is also important to reveal anomalies and if we limit this value we could lose something	N
	additionalInfo	list of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure and are necessary to look up the pig and the related pen, building and farm in the database. It is a JSON structure similar to this: { "ILVOLFtag":"984000100625798" "ILVOPigId":"8", "ILVOPenId":"1", "ILVOPeriod":"3", "ILVOHFtagLeft":"E00401005BA42946", "ILVONRstation":"1798", "ILVOHFtagRight":"E00401005BA438AD", "ILVOSanite"l:"75503", "Name": "", "First":"21/08/201900:01:54", "Last":"21/08/2019 00:01:54",	Object (JSON)			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
		"Cycle": 1, "Port":1, "Location":"Feeder 4" }				
SlaughteredPig	lastSlaughteredPigid	Unique identifier of the latest Slaughterd Pig	String		How do you uniquely identify a slaughtered pig? Is it still the same identification as the PigID when the pig was alive? A UUID4 is generated and added to the word "SlaugtheredPig-" to create a unique identifier for the slaughtered pig. No, it is not the same ID as Pig. There is only a link among them.	Y
	lastUpdated/lastUpdateTimestamp	The day and time the latest pig was slaughtered	int			Y
	serialNumber	The serial number of the latest Slaughtered Pig. This is the same serial number as used in the Pig entity? It depends on the farms. We are not aware of this information	String			N
	slaughterhouseid	Unique identifier of the latest Slaughterhouse the Pig was slaughtered in	String			Y
	additionalInfo	List of all the raw values sent by the sensor/platform with all the possible extra properties that are not included in the main structure Which additional properties are included for a Slaughtered Pig? At the current moment we are not treating any slaughterhouse information, so we are not still aware of these values How does this differ from the additionalInfo attribute in the Pig entity? We could, for example, slaughter pigs there were not monitored in the farm	Object (JSON)			N
Slaughterhouse	slaughterhouseid	Unique identifier of the Slaughterhouse	String		How do you uniquely identify a slaughterhouse? A UUID4 is generated and added to the word "Slaughterhouse-" to create a unique identifier for the slaughterhouse	Y
	address	Postal addres of the Slaughterhouse? In our case the address is not completely useful, thus it is treated as a general information and we inserted it as a String.	String			N

Data Model	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)
		Each slaughterhouse can indicate it in its preferred format. Can it also be a Geo location? Not in our case				
	name	Name of the Slaughterhouse	String			N
	additionalInfo	List of all raw values sent by the sensor / platform with all possible extra features that are not included in the main structure and are necessary to look up the Slaughterhouse in the database.	Object (JSON)			N
Company	companyid	Unique identifier of the Company owning pig farms	String		How do you uniquely identify a company? A UUID4 is generated and added to the word "Company-" to create a unique identifier for the company Is no address required to uniquely identify a company? In our case it is not necessary	Y
	name	Name of the company	String			Y

4.2. Glossary and data dictionary defined by use case 5.2

4.2.1. UC5.2 Domain Terms Glossary

Table 9: UC5.2 Domain Terms Glossary

Term	**Definition**	**Synonyms**	**Homonyms**	**Preferred Term (Y/N)**	**Relationships to other elements**	**Range of values**	**Validation rules**
Animal scale	An instrument for weighing the animals			Y			
Area	A part of the house. A house can be split in different areas (1, 2, 3 etc.) where the parameters (temperature, ventilation etc.) are handled separately. The areas can be registered in the Integral Farm Controller by the farmers	Zone			(1) is part of a House	It depends of the size of the house and the sensors equipment	Positive values only
Breeding day	The day number since the flock arrived at the farm. breedingDay is equal to one the day of their arrival. breedingDay is equal to ten when it is the tenth day at the farm			Y	Related to Flock	From 1 to X days where X is the last day at the farm that depends of the poultry production system. In the use case it is between 1 and about 80 days	Positive values only
Breeding period	The breeding of the broilers is split in different periods. For each period the most suited parameters values are estimated			Y	Related to Flock	It depends of the poultry production system. For example, it can be from day 1 to day 3	Positive values only
Broiler	A meat-type chicken that is usually less than 12 weeks of age and has been bred specifically for meat production. For the use case, no age grouping is done.	Fryer: a young meat-type chicken	Chicken: broader term than broiler that can be used for non meat chickens		(1) is reared in a House (2) belongs to a Flock (3) is owned (optional) by an Integrator (4) is reared by a Farmer		
Farm	A company located at a distinct geographic location with agricultural production activities			Y	(1) contains one or several Houses		

Term	Definition	Synonyms	Homonyms	Preferred Term (Y/N)	Relationships to other elements	Range of values	Validation rules
Flock	A flock is a group of broilers living together			Y	(1) contains Broilers (2) is reared in a House		Positive values only
House	A building in which poultry are sheltered or reared	Shed? Building? Barn?		Is House the preferred term?	(1) is part of a Farm (2) contains (optionnal) Areas (3) contains one Flocks or is empty (cleaning between two Flocks)		
Inlet	An opening in the house through which fresh air enters that plays a role to the house ventilation			Y	(1) is part of a House (2) usually there are several inlets in a House?	It depends of the house size?	
Integrator	An integrator owns the broilers. The farmers are not always the broilers' owners. A company called integrator can provide the broilers for the farmers to reared them. In the use case, all the broilers are owned by SADA	Company?		Y	(1) owns Broilers (2) has business relationship with farmers	(1) A farmer can have a business relationship with more than one integrator at the same time (2) an integrator can have a business relationship with more than one farmer. It is true fo rthe use case	
Ship							
Silo	A storage structure used to store the feed			Y	(1) is part of a Farm	A farm can have more than one silo	
Silo scale	An instrument for weighing the silos. It is used to estimate the feed consumption	Load cell			(1) is part of a Silo		
Static pressure	The air pressure difference between the inside and outside. It causes the air flow that produces the air exchange required as part of a mechanically ventilated poultry house				(1) is a parameter of a House		
Transverse window					(1) is part (optional?) of a House	A House can have several transverse windows	
Tunnel window					(1) is part (optional?) of a House	A House can have several tunnel windows	

4.2.2. UC5.2 Orion Context Broker in IOT Platform

Table 10: Data dictionary UC5.2 Orion context broker data in IOT Platform

Entity	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)	Implemented (Y/N)
General entity attributes	breedingDay	The day number since the flock arrived at the farm. breedingDay is equal to one the day of their arrival. breedingDay is equal to ten when it is the tenth day at the farm	Integer	It depends of the poultry production system. In the use case it is between 1 and about 80 days	Positive values only	Y It is a key attribute in the data models. The analytics work is based on it	
	dateObserved	The date of the observation	String			Y	
	idFarm	The identification of a farm by its name	String		a Farm has only a unique name in the use case	Y	
	idFlock	The unique identification of a flock. The identification contains the information of when and where the flock is reared	Integer		The idflock indicates the initial day of a flock The id of a flock is not unique by itself but becomes unique combined with the identification of Farm, Integrator and House	Y	
	idHouse	The unique identification of a house by its number assigned by the farmer	String		a House has only a unique number inside a farm	Y	
	idIntegrator	The unique identification of an integrator by its name	String		an Integrator has only a unique name in the use case		
WeatherPredictionObservation entity the weather forecast of the location covered by the farm provided by AEMET. It defines the parameters related to wind,							
	precipitation	The probability of precipitation in percentage	Float	From 0 to 100%			
	relativeHumidity	The ratio of the partial pressure of water vapor to the equilibrium vapor pressure of water at a given temperature, expressed in percentage	Float	From 0 to 100%			
	skyDescription	How open is the sky percentage of cloud cover? (clear, overcast, clouds etc.?)	String				
	temperature	The temperature in degree Celcius	Float				

Entity	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)	Implemented (Y/N)
precipitation, relative Humidity etc.	windDirection	The wind direction expressed as an abbreviation (NE, NO etc.)?	String				
	windSpeed	The speed of the wind in kilometer per hour	Float				
	getWindSpeed	The wind direction as an abbreviation (NE, NO etc.)?	Float				
FarmPopulationObservation entity defines information related to the ship, farm and the flock	deaths	The number of dead broilers since the arrival of the Flock?	Integer	From 1 to the number of broilers in the flock	Positive values only		
	numberOfAnimals	The number of broilers in the flock	Float				
	sentToSlaughterhouse	The number of broilers (of a Flock?) sent to the slaughterhouse	Integer	From 1 to the number of broilers in the flock	Positive values only		
AirQualityObserved entity defines information related to environmental variables measured by sensors in a fixed position higher than the broilers level in the farm ship. It is managed by the Exafan Farm Management Application.	calculatedAirInletSpeed	The incoming and outgoing air inlet speed in meter per second	Float		Positive values only		
	calculatedStaticPressure	The static pressure in Pascal	Float		Positive or negative values?		
	co2	The CO2 concentration in parts-per-million (ppm)	Integer		Positive values only		
	currentWorkRegime		String				
	desiredTemperature-ForVentilation		Float				
	ext.RH	The relative humidity outside the house in %	Float	From 0 to 100%			
	ext.anemometer	The external wind speed in meter per second and direction measured by a sensor placed outside the house	Float				
	ext.temperature	The external temperature in Celcius measured by a sensor placed outside the house	Float		Positive or negative values		
	ext.vane	The wind direction outside the house indicated by a metal object on the roof of a building that turns easily in the wind	String				
	heating1ONOFF	This attribute indicates if the heating system is ON or OFF in the area 1.	String	ON -OFF - NULL?			
	lightChannelAnalog1		Integer				
	lightChannelDigital1		String				
	m2OfTransverseWindows	The surface of transverse windows in meter square	Float		Positive values only		
	m2OfTunnelWindows	The surface of tunnel windows in meter square	Float		Positive values only		
	m3HourMinimalVentilationCalculated	The minimal ventilation needed in the house per cubic meter per hour	Float				
	m3HourVentilationCalculated	The current ventilation in the house per cubic meter per hour	Float				
	nh3	The NO3 concentration in parts-per-million (ppm)	Integer		Positive values only		

Entity	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)	Implemented (Y/N)
	operationTimeHeating1	This attribute indicates how much time the heating system has been working for a breeding day for the area 1	Float	From 0 to 24h?			
	operationTimeInLongitudinal		Float				
	operationTimeInMinima		Float				
	operationTimeInMixed		Float				
	operationTimeInTunnel		Float				
	operationTimeRefrigeration1	This attribute indicates how much time the cooling system has been working for a breeding day for the area 1	Float	From 0 to 24h?			
	refrigeration1ONOFF	This attribute indicates if the cooling system is ON or OFF in the area 1.	String	ON -OFF - NULL?			
	relativeHumidity	The relative humidity in %	Integer	From 0 to 100%			
	staticpressure	The static pressure in Pascal	Integer		Positive or negative values?		
	temperatureOFFHeating1	The temperature above which the heating system switches off in the area 1	Float				
	temperatureOFFRefrigeration1	The temperature below which the cooling system switches off in the area 1	Float				
	temperatureONHeating1	The temperature below which the heating system switches on in the area 1	Float				
	temperatureONRefrigeration1	The temperature above which the cooling system switches on in the area 1	Float				
	temperatureZoneLeftCenter	The temperature in degree Celcius of the zone (see Area definition) left center of the house	Float				
FarmConsumptionObserved entity defines the consumption (electricity, gas, water and food) in the farm ship	electricPowerConsumption	The electric power consumed by the farm (e.g. to light the houses)	Float			**FarmConsumptionObserved entity** the consumption in the farm ship	
	gasConsumption	The gaz consumed by the farm (e.g. to heat the houses)	Float				
	silo1	The unique identification of a silo by its number assigned by the farmer	Integer		a Silo has only a unique number inside a farm		
	silo1Load	The weight of the silo 1	Integer		Positive values only		
	waterConsumption	The average water consumption of the broilers	Integer				
PoultryWeightObserved entity defines the weight of the broilers	scaleNumber	The unique identification of an animal scale by its number	Integer		an animal scale has only a unique number inside a farm		
	weight	The estimated weight of a broiler. It is impossible to be sure that the measure weight was for one animal as the broilers are free to walk to the scale and to be with others on the scale	Float				

Entity	Term	Definition	Format	Range of values	Validation rules	Mandatory (Y/N)	Implemented (Y/N)
CurvesOfBreeding entity Pre-defined Breeding Curves for the farm. It defines the predefined values by range of days for temperature, relative humidity, m3hourPerMeatKg etc.	breedingDay01	The breeding period 1	Integer			**CurvesOfBreeding entity** the pre-defined breeding curves for the farm	
	externalTemperatureToChange		Float				
	hvac		Integer				
	inflectionPoints		Integer				
	m3hourPerMeatKg	The cubic meters of fresh air per kg body weight and hour that the ventilation system should provide	Float				
	m3hourPerMeatKg01	The cubic meters of fresh air per kg body weight and hour that the ventilation system should provide for the period 1	Float				
	maxVentilation	The maximum ventilation that should be provided	Float				
	maxWindowOpen		Integer				
	minWindowOpen		Integer				
	relativeHumidity01	The relative humidity in % needed for the period 1	Integer	From 0 to 100%			
	temperature	The current/measured temperature in degree Celcius	Float				
	temperature01	The temperature in degree Celcius needed for the period 1	Float				
	tunelAirSpeed	The speed of the air in the tunel	Float				
	weigth	The ideal weight of the broilers in kg?	Integer		Positive values only		
	windInSpeed	The speed of the wind in meter per second	Float				
	windSpeedMin	The minimum speed of the wind in meter per second	Integer				
SensorAirQualityObserved entity Environmental Data from the Tekniker Environmental devices. This data model defines the values of the sensor, temperature, relative humidity, nh3, co2 etc.	TimeInstant	The date and time of the observation	ISO8601				
	luminosity	The luminosity in lux	Float				
	solarRadiation	The solar radiation in watts per square meter	Float				N Sensors at animal level don't send this information

4.3. Event model UC5.3

4.3.1. UC5.3 Event Flow Model and Listing of Events

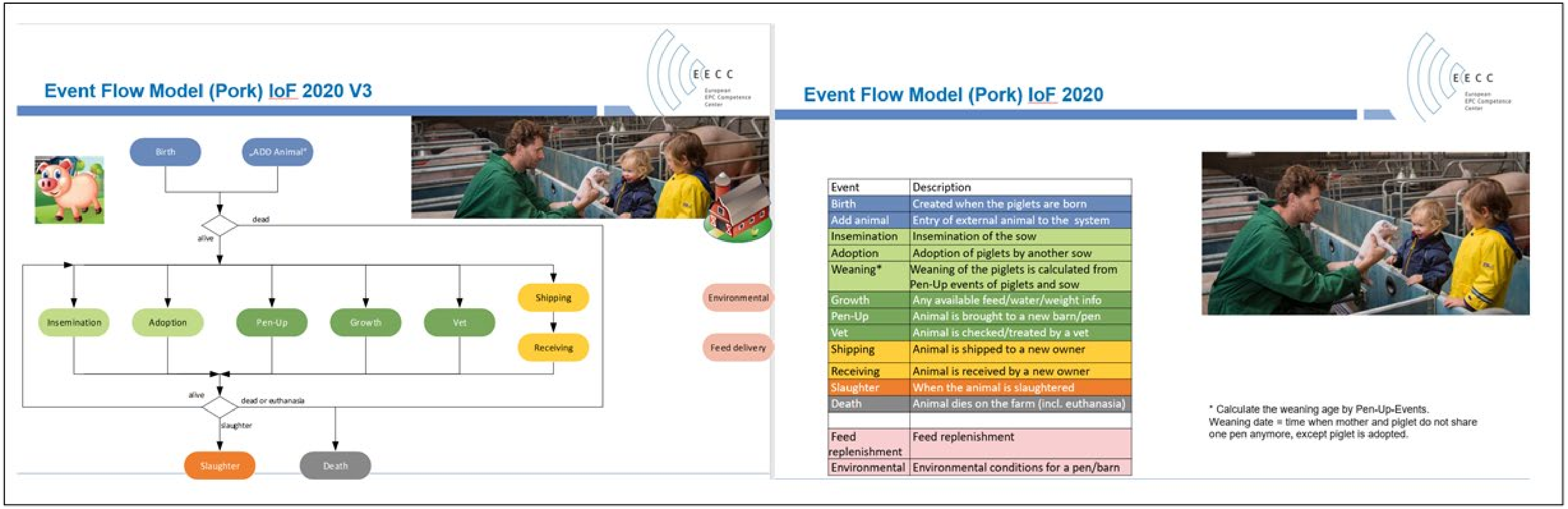


Figure 18: Event Flow Model UC5.3 and list of events.

UC5.3 Event details

UC5.3 already made use of a generic data model, the Core Business Vocabulary, a companion to the EPCIS standard. The CBV standard contains definitions of data values that can be used to fill the data structures defined in the EPCIS standard to describe the observed process steps. The use of a standardized vocabulary is critical to interoperability and enables the exchange of data by providing different organizations with a common language to describe processes. If the defined vocabulary is not sufficient, the standard provides methods to use industry-specific or proprietary vocabulary elements. As far as the UC5.3 event model is concerned, almost all requirements are met by the CBV. Only values like drug treatments and the dose in Vet events needed the introduction of specific vocabulary.

The architecture defined in the EPCIS standard provides capture and accessing applications to serve the capture and query interfaces. Capture applications prepare the data to be captured, that the defined events can be written to the EPCIS repository. Ideally the data is already delivered in EPCIS/CBV format. On the other end, the AccessingApps collect the EPCIS event flow and prepare the data for defined request. In UC 5.3 this pre-aggregated information is presented on the Auditor’s dashboard.

Detailed information on EPCIS and CBV can be found following this link: https://www.gs1.org/standards/epcis

Beyond this event model, UC5.3 has contributed to UC4.5 on Digital Ecosystem Utilization (CYSLOP) - Post- farm processes. With the advice of UC5.3, UC4.5 developed its own EPCIS event model for the table olives supply chain, using an EPCIS-compliant solution based on EECC's EPCAT for the data store. Beyond this event model, UC5.3 has contributed to UC4.5 on Digital Ecosystem Utilization (CYSLOP) - Post- farm processes. With the advice of UC5.3, UC4.5 developed its own EPCIS event model for the table olives supply chain, using an EPCIS-compliant solution based on EECC's EPCAT for the data store.

For each UC5.3 event listed in Figure 18 in this appendix per event an overview of the data is included.

Figure 19: UC5.3 Event data overview of “Birth Event” and “Add Animal” Event

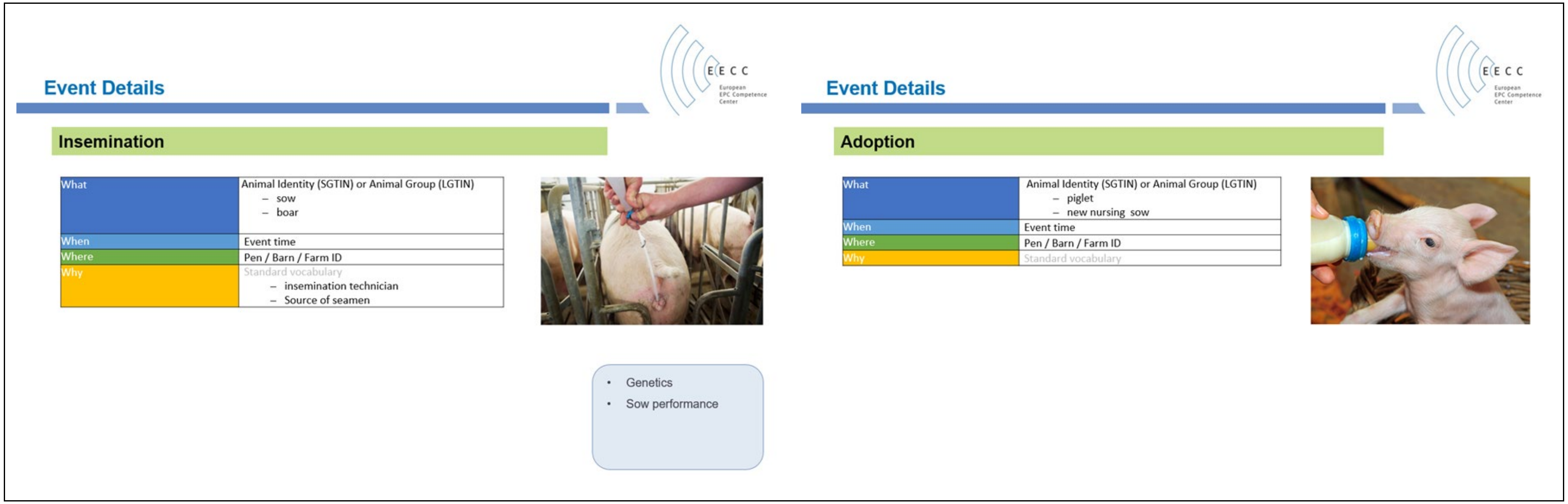


Figure 20: UC5.3 Event data overview of "Insemination" and "Adoption" Event

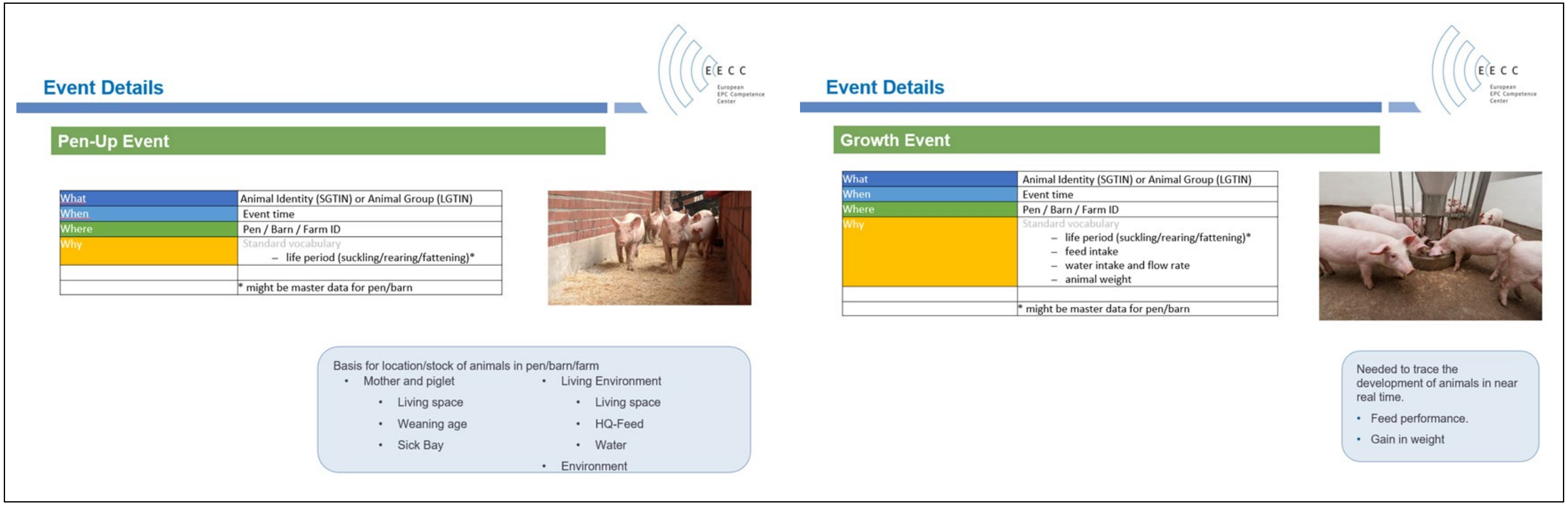


Figure 21: UC5.3 Event data overview of "Pen-Up Event" and "Growth Event"

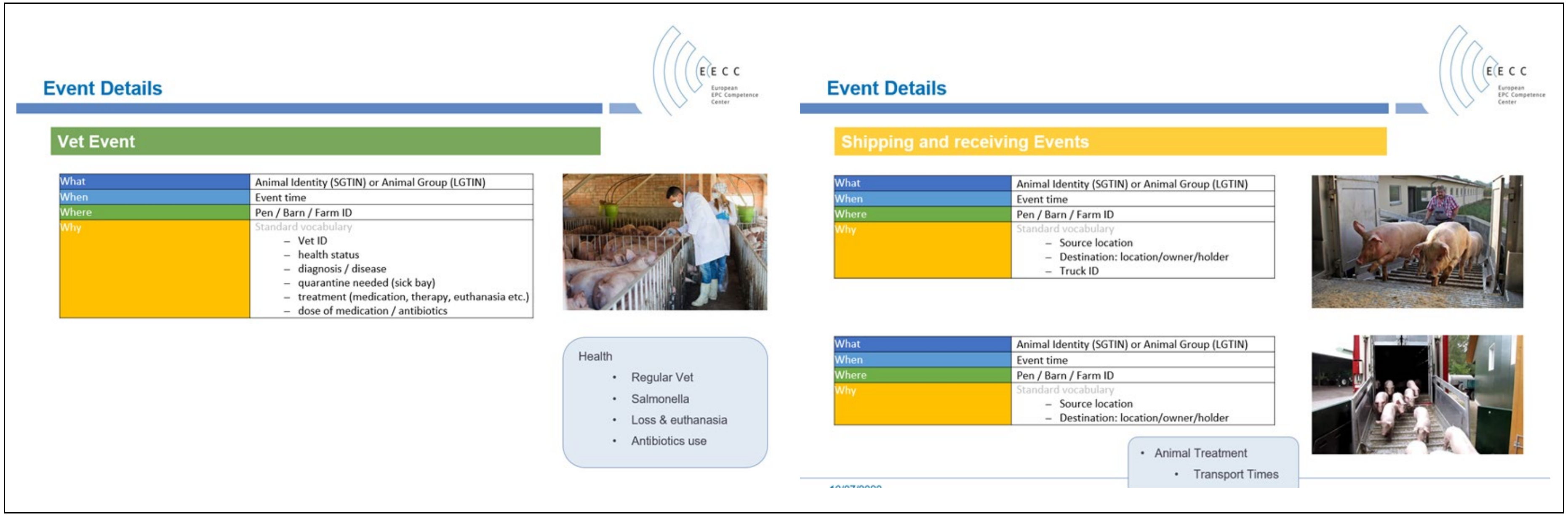


Figure 22: UC5.3 Event data overview of “Vet Event” and “Shipping and receiving Events”

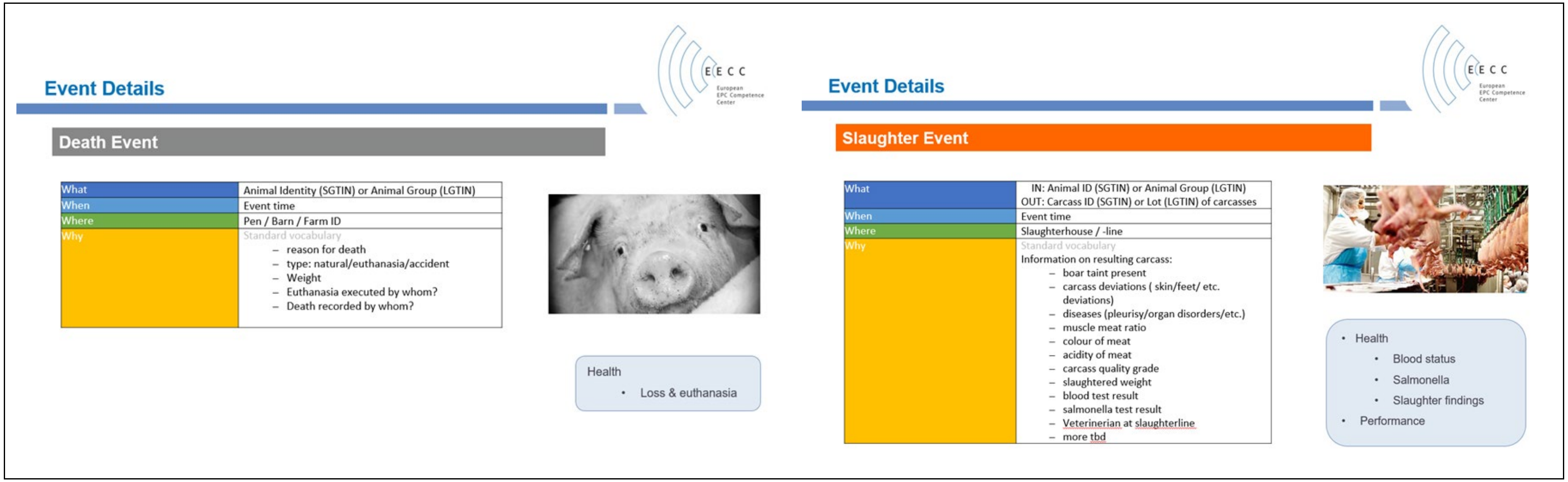


Figure 23: UC5.3 Event data overview of "Death Event" and "Slaughter Event"

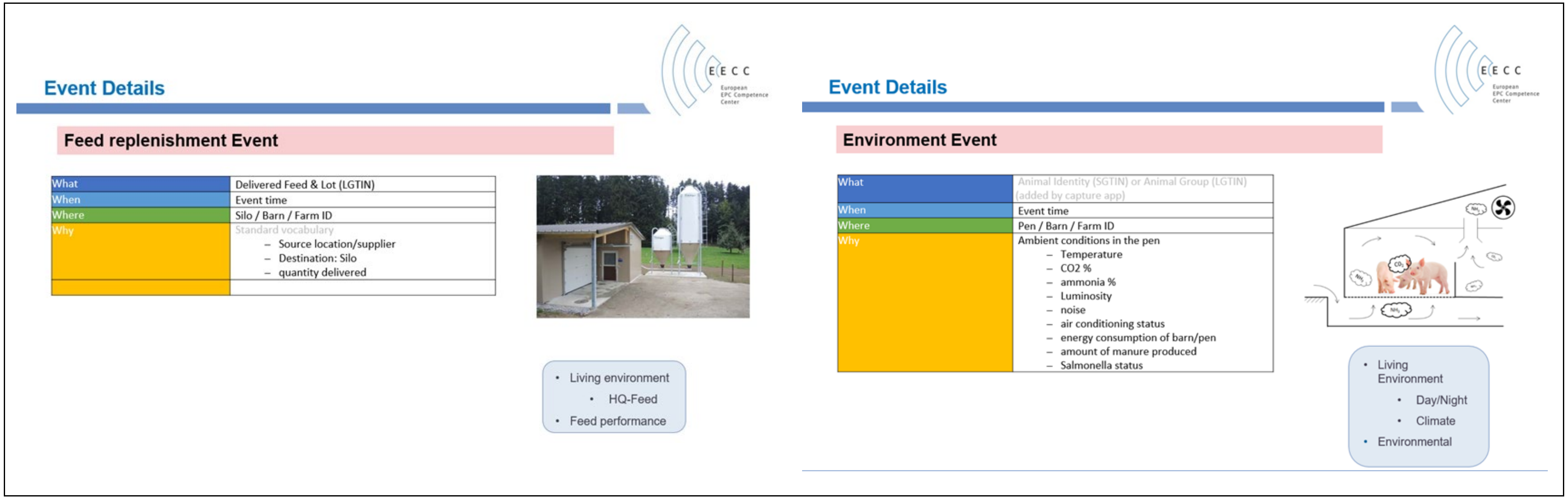


Figure 24: UC5.3 Event data overview of "Feed replenishment Event" and "Environment Event"

4.3.2. UC5.3 Relevant Master Data for GLN ID's as defined so far

Relevant Master Data (to be extended…)

European EPC Competence Center

Locations	farmer	– contact details	
		– GLN	
	barn	– SGLN	
		– adress/barn	
		– owner of barn	
		– capacity/barn	
	pen	– SGLN	
		– area/pen	
		– capacity/pen	
		– barn SGLN	
		– feed storage SGLN	
		– lightning type	
		– floor type	
		– area/sqm	
		– feeding system	
		– Farming concept	
		– type of Farming	
	truck	– type	
		– licence plate	
		– capacity	
		– specs	
	feed silo	– SGLN	
		– capacity	
		– barn SGLN(s)	

Figure 25: Relevant Master Data for GLN ID's as defined so far for UC5.3

4.4. Data modelling in UC5.4

4.4.1. UC 5.4 Implementation of the IoF2020 Reference Architecture

In Figure 26 the UC5.4 ShareBeef implementation of the IoF2020 reference architecture is depicted.

- The connectivity layer aims to connect, through several long range low power communication standards such as Sigfox, 2G/3G/4G, NBIoT, Lora and even Satellite Low Power Global Area Network, the IoT battery-powered devices used with the platform.
- The IoT service layer will handle the device management and the integration, through MQTT and similar protocols, with the mediation layer. In this layer, the information coming from external open data sources and the information coming from the IoT devices is adapted, transformed and connected, through a FIWARE IoT Agent, with the information management layer.
- The information management layer acts as a data centre to enable publication, analysis, storage, blockchain integration and subscription of all information generated by ShareBeef components. The core is based on FIWARE technologies that handle NGSI-10 based data.
- The application layer includes all the services and apps provided to the data consumers, like farmers, consumers, authorities, open data, etc. It includes a decision support system that generates information from the data, dashboards and apps to visualize the information and services to expose the data through APIs to ensure the interoperability with other platforms and services.
- Supervising all the processes and data exchange the security layer is located. It handles all the security and privacy aspects of the platform.
- The information models harmonization layer ensures that all the information manage by the mediation and information layers is harmonized and follow the standards.

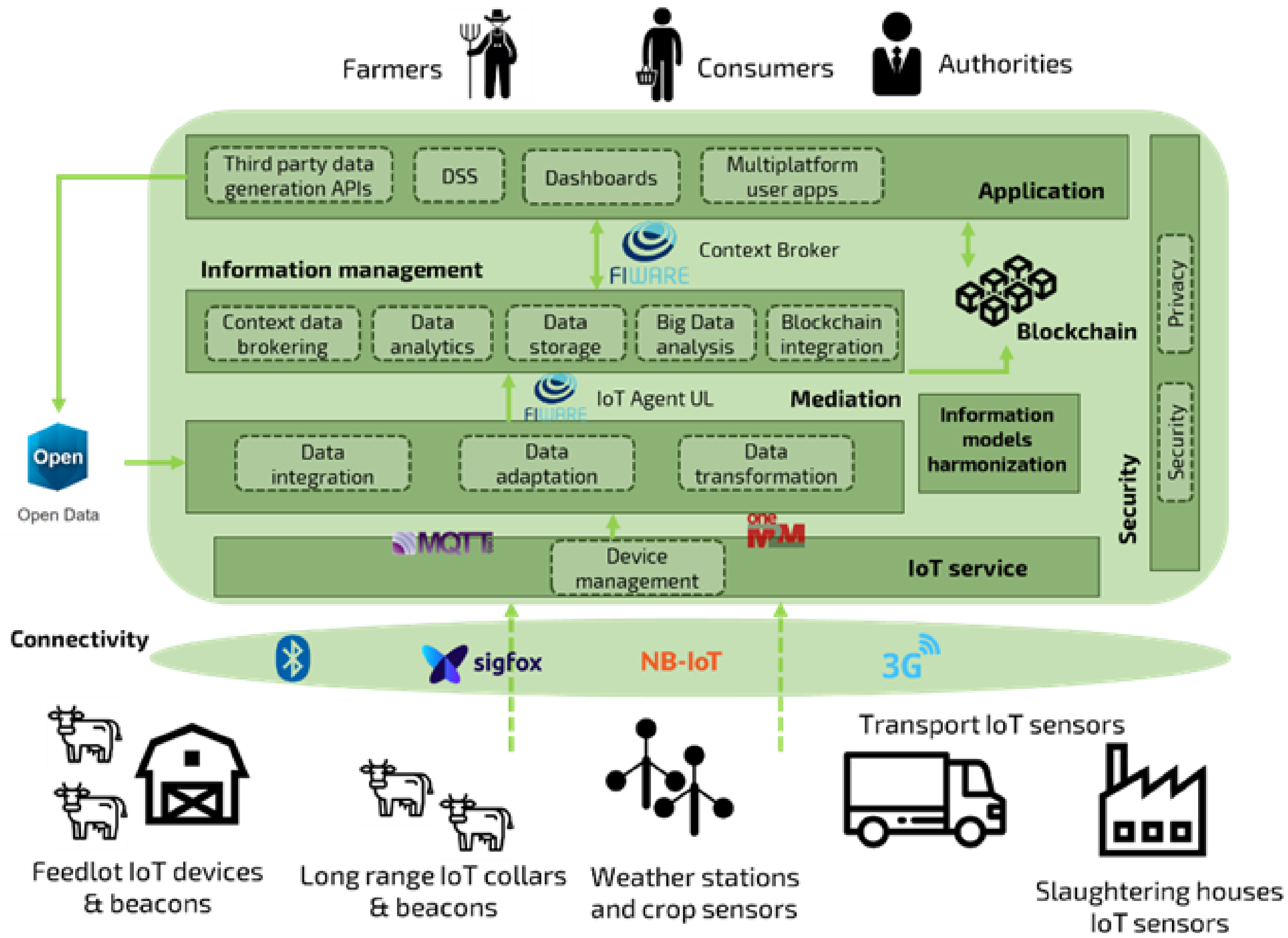


Figure 26: UC5.4 ShareBeef implementation of the IoF2020 reference architecture.

4.4.2. UC 5.4 Beef Chain

Figure 27 shows the scope of the use case 5.4 Shared Beef. The beef chain shows different stakeholders in the beef chain as well as some of their interaction.

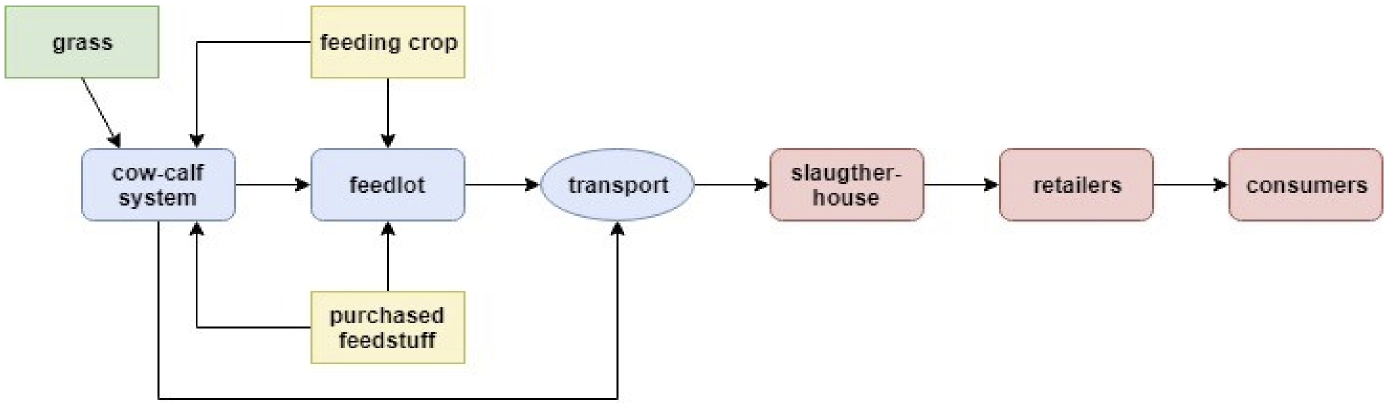


Figure 27: Beef chain considered in UC5.4 Shared beef, source: [*https://github.com/smart-data-models/dataModel.Agrifood/blob/66f5b46597e3562e925f965dfa76952795fa2db6/Animal/doc/spec.md*](https://github.com/smart-data-models/dataModel.Agrifood/blob/66f5b46597e3562e925f965dfa76952795fa2db6/Animal/doc/spec.md)

In the use case SensoWave is in charge of monitoring the animals. They collaborated with Agricolus who was responsible of the crop monitoring and handled weather stations and field sensors, e.g. for decision making on crops like irrigation and fertilization. The use case also used satellite data to correlate data of the cow-calf system with e.g. pasture quantity. In the meat trial focus is mainly on the Animal (the purple coloured part of the diagram) and we only received data model details about the Animal data model.

4.4.3. UC 5.4 Animal Model Definition

In Figure 28 the data model as provided in the animal model of the FIWARE Smart Models[28] is depicted. It focusses mainly on the animal, but the use case also considers for example the feed intake, i.e. the crops to feed the animals and the growth conditions (weather conditions, humidity, etc.) of those crops. Of the latter no details were provided.

In Table 11 the properties of the Animal data model as specified in the FIWARE Smart Models are listed. The model may slightly change in time, but since the model is maintained in GitHub it will be possible to view the change history.

[28]: https://github.com/smart-data-models/dataModel.Agrifood/tree/66f5b46597e3562e925f965dfa76952795fa2db6/Animal

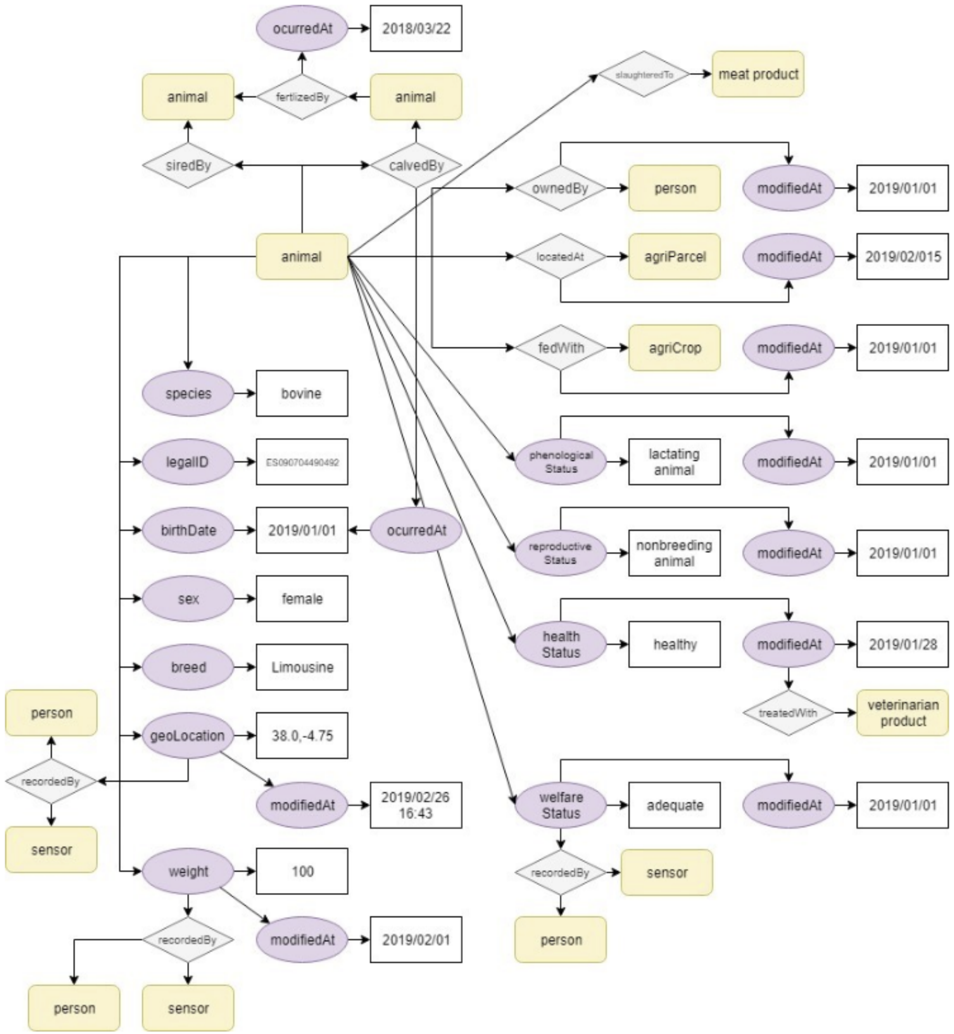


Figure 28: Data model UC5.4 - focus on the animal, relations with Location (agriParcel) and feed (agriCrop). Source: https://github.com/smart-data-models/dataModel.Agrifood/blob/66f5b46597e3562e925f965dfa76952795fa2db6/Animal/doc/spec.md

Table 11: Animal data model as specified in FIWARE Smart Data models. Source: https://github.com/smart-data-models/dataModel.Agrifood/blob/master/Animal/doc/spec.md

Attribute	Description	Format	Range of values	Mandatory
id	unique identifier			Y
type	Entity type. For the Animal data model it must be equal to "Animal"			Y
species	Species to which the animal belongs	Text	Allowed values: (dairy cattle, beef cattle, sheep, goat, horse, pig)	Y
relatedSource	ID of the animal in external applications	List of StructuredValue	application: ID of the particular AgriApp entity Type: URI applicationEntityId: ID of the animal in the external application Type: Text	N
legalID	Legal ID of the animal	Text		Y
birthdate	Animal's birthdate	DateTime		Y
sex	Sex of the animal	Text	Allowed values: (female, male)	Y
breed	Breed of the animal	Text		N
calvedBy	Mother of the animal		Relationship to other Animal	N
siredBy	Father of the animal		Relationship to other Animal	N
location	Location of the animal represented by a GeoJSON geometry with optional timestamp for the observed value	Geo.json		N
weight	The weight of the animal in kg with optional timestamp for the observed value	Number		N
ownedBy	The owner of the animal		Relationship to Party/Person	N
locatedAt	AgriParcel relationship with optional timestamp for the observed value		Relationship to AgriParcel	N
phenologicalCondition	Phenological condition of the animal with optional timestamp for the observed value	Text		N
reproductiveCondition	Reproductive condition of the animal with optional timestamp for the observed value	Text		N
healthCondition	Health condition of the animal with optional timestamp for the observed value	Text		N
fedWith	Food used for the animal with optional timestamp for the observed value		Relationship to feed	N
welfareCondition	Indicator of the animal welfare with optional timestamp for the observed value	Text		N

4.5. Data modeling in UC5.5

UC 5.5 Domain model

This use case aims to optimise the integral feedstock management system in order to reduce costs and enhance labour efficiency. The farmer, the feed manufacturer and the nutrition expert are interested in monitoring feed stock, and consumption rate in order to achieve timely restocking of feed silos in combination of efficient truck route planning by the feed manufacturer. Figure 29 below, shows the domain model made by the use case.

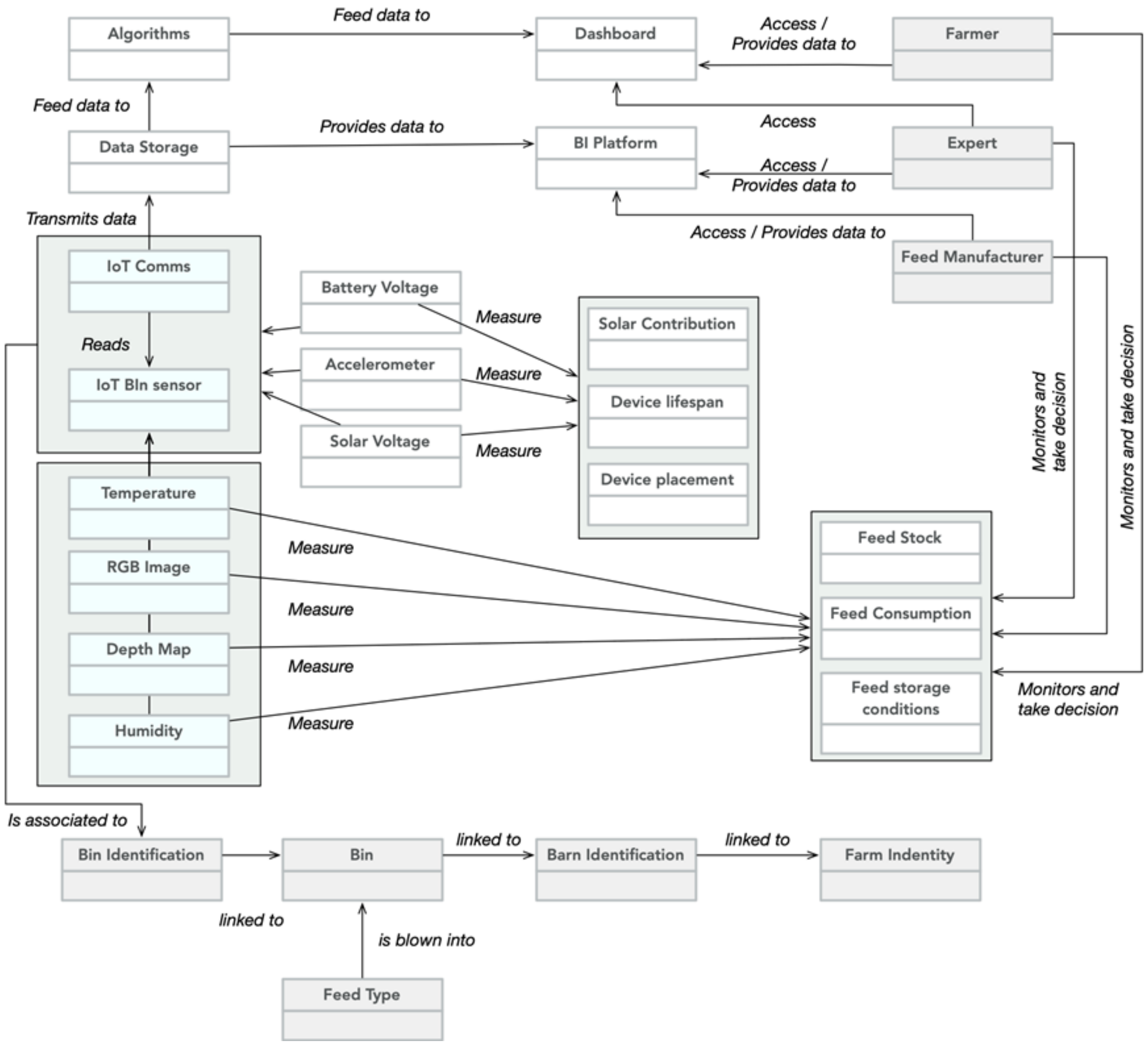


Figure 29: Domain model of UC 5.5 Feed Supply Chain Management. Source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

4.5.1. UC 5.5 Data model and API

UC5.5 provides an API to get access to the data the use case gathered. The API is available at the URL https://apis.insylo.io/swagger/index.html.

The basic data model is depicted in Figure 30 below. The main subjects supported by this API are:

- Setting areas and granting access to bin data
- Getting historical and current stock values from bin or area
- Setting material properties by informing recipe/diet (material density)

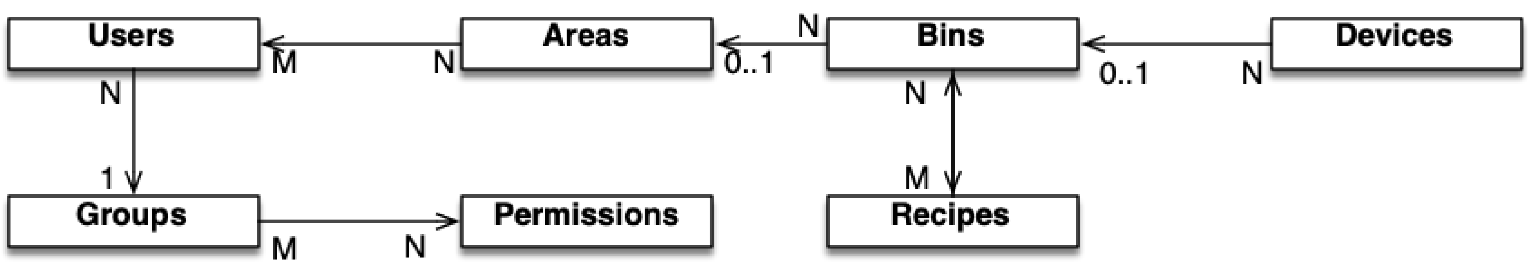


Figure 30: The basic data model the UC5.5 API exposes, source: UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

Details of the Devices, Areas, Bins and Recipes are shown in Figure 31.

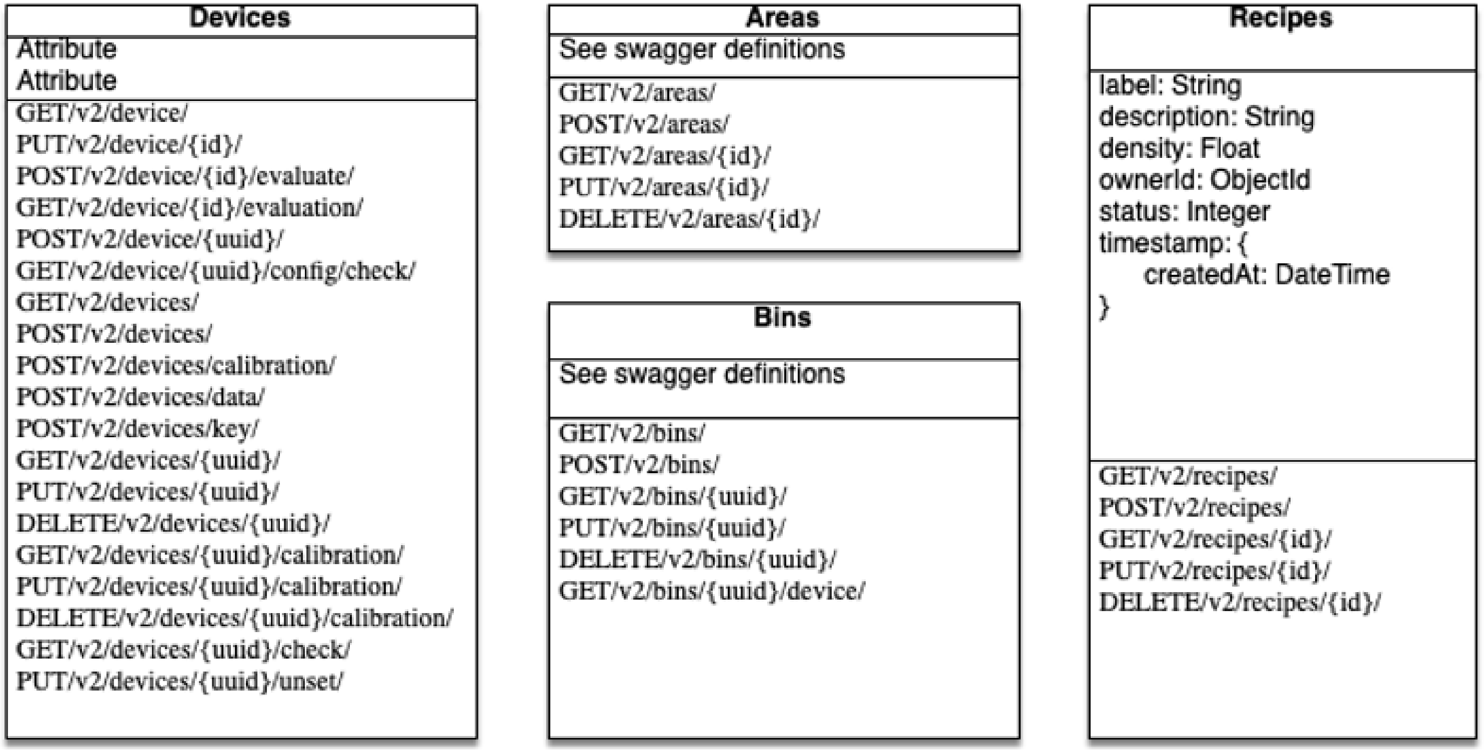


Figure 31: API endpoints for Devices, Areas, Bins and Recipes

At the time of interview with the use case they were in the middle of harmonizing the data model with NGSI Schema and they used the FIWARE Waste Monitoring for Smart Cities as reference for their Silo monitoring since it is quite similar. The preliminary harmonised data models for Silo monitoring are displayed in the next figures.

SILO


```
{
  "id": "feedsilo:GroupId:s12",
  "type": "FeedSilo",
  "refFeedSiloModel": "feedsilomodel:manufacturer:m15",
  "refFeedSiloFarm":  "feedsilofarm:ClientName:001",
  "serialNumber": "UBK1233B842-12JE-4B89-1E0F-1B72CB2AE3AB",
  "location": {
        "type": "Point",
        "coordinates": [  2.1449558484494, 42.85759498839384 ]
  },
  "weightingCell" : 4582,
  "fillingLevel" : 0.4,
  "feedDensity" : 640,
  "userAlertFull": 90,
  "userAlertHalf": 50,
  "userAlertWarning": 20,
  "userAlertEmpty": 10,
  "userAlertCustom1": 0,
  "userAlertCustom2": 100,
  "dateLastRefilling": "2017-11-18T15:05:59.408Z",
  "dateEstimatedNextActuation": "2017-12-10",
  "status": "ok",
  "feedType": ["granular"],
 "feedRef":"RecipeId"
  "refDevice": ["UBK-INSYLO:ABC:0000001"]
}
```


Figure 32: Preliminary NGSI data model Silo, source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

```
SILO MODEL
{
  "id": "feedsilomodel:manufacturer:15T",
  "type": "FeedSiloModel",
  "name": "Fibber-silo-2012-15TM-4-LEGS",
  "cylinderWidth": 0.50,
  "cylinderHeight": 0.50,
  "cylinderAngle": 45,
  "coneWidth": 0.50,
  "coneHeight": 0.50,
  "coneAngle": 60,
  "distanceHole":43,
  "aimingAngle":5,
  "distanceTruncatedCone":30,
  "maxDepth": 4.45,
  "cargoVolume": 23,
  "cargoWeight": 15000,
  "brandName": "BrandName",
  "modelName": "15T",
  "compliantWith": ["UNE-ENV-1991-4:1995"],
  "madeOf": "fibber",
  "features": ["metallic pilars", "tap"],
  "category": ["feed-silo"]
}
```

Figure 33: Preliminary NGSI data model Silo Model, source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

```
FARM
{
  "id": "feedsilofarm:ClientName:001",
  "type": "FeedSiloFarm",
  "location": {
      a.   "type": "Point",
      b.   "coordinates": [  2.1449558484494, 42.85759498839384 ]
  },
  "address": {
           "streetAddress" : "N-232 km12",
           "addressLocality": "Vic",
           "addressCountry": "ES"
  },
  "features": ["pig-producer"],
  "name": "Can Marsal",
  "description": "Client Name - Pig producer",
  "containers": ["silofeed:FarmName:s1", "silofeed:FarmName:s2"]
}
```

Figure 34: Preliminary NGSI data model Farm, source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

```
DEVICE
{
  "id": "UBK-INSYLO:ABC:0000001",
  "type": "Device",
  "category": ["sensor"],
  "controlledProperty": ["humidity", "temperature",
                    "remainingVolume", "remainingWeight", "batteryLevel",
                    "solarPower", "solarUtilization", "sensorRoll",
                    "sensorPitch", "sensorYaw",
                    "deltaRoll", "deltaPitch", "deltaYaw"],
  "ipAddress": "84.18.13.XX",
  "mcc": "214",
  "mnc": "07",
  "batteryLevel": 0.75,
  "serialNumber": "UBK1233B842-12JE-4B89-1E0F-1B72CB2AE3AB",
  "refDeviceModel": "UBIKWA-INSYLO-feedsensor-02",
  "value": "v=0.21;w=134.4;t=21.2;h=56.2;sr=23;sp=34;sy=43;dr=13;dp=4;dy=13;
            b=34;s=34;su=83",
  "deviceState": "ok",
  "dateFirstUsed": "2017-01-08T10:34:00Z",
  "dateCreated":"2017-01-08T10:34:00Z"
}
```


Figure 35: Preliminary NGSI data model Device, source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx


```
DEVICE MODEL
{
  "id": "UBIKWA-INSYLO-feedsensor-02",
  "type": "DeviceModel",
  "name": "INSYLO Feed Sensor for Silos v2",
  "brandName": "INSYLO",
  "modelName": "INSYLO HQ",
  "manufacturerName": "UBIKWA SYSTEMS SLU.",
  "supportedProtocol":["ul20","mqtt","http","coap","gprs"],
  "category": ["sensor"],
  "function": ["sensing"],
  "controlledProperty": ["humidity", "temperature",
                    "remainingVolume", "remainingWeight", "batteryLevel",
                    "solarPower", "solarUtilization", "sensorRoll",
                    "sensorPitch", "sensorYaw",
                    "deltaRoll", "deltaPitch", "deltaYaw"]
}
```


Figure 36: Preliminary NGSI data model Device Model, source UC55 DDF-IOFEED-20191227-DRS-1-1 – System Architecture and Data Model.docx

4.6. Data modeling in UC5.6

4.6.1. UC5.6 Measurements

This use case aims to optimize wellbeing and health care costs of pigs with regard to pig farming by intensive monitoring trough IoT sensors. They use two devices, namely the PPG sensor and the 3D accelerometer embedded into an ear tag and the Smart Spot Gateway.

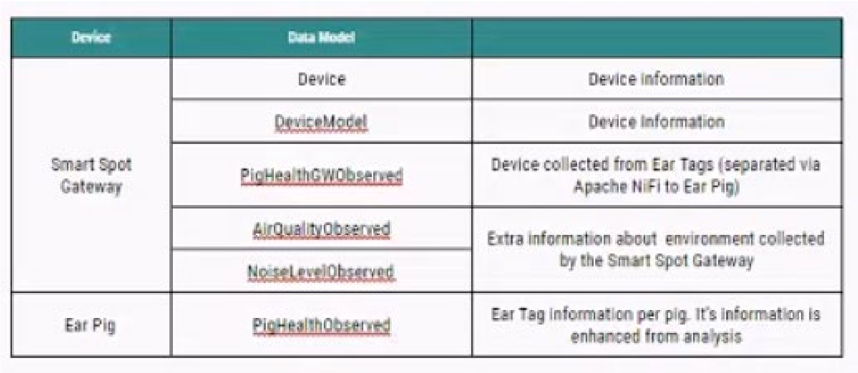
The ear tag sensors collect the following data:

- Pig activity
- Pig heart rate
- Temperature

And the Smart Spot Gateway collects additional environmental data like (see AirQualityObserved and NoiseLevelObserved in Figure 37):

- Air quality measurements: ammonia, carbon dioxide and dust particles
- Weather: temperature and humidity
- Noise level

The ear tag information is also captured by the Smart Spot gateway (see PigHealthOGWbserved in Figure 37). The Smart Spot gateway only contains the latest measurements, so the collected information is transformed in order to store historical information (see PigHealthObserved in Figure 37). This is done using an Apache NIFI component which is also capable of deduplicated data. Sometimes one building has several gateways close to each other, and then information might have to be deduplicated.



Device	Data Model	
Smart Spot Gateway	Device	Device information
	DeviceModel	Device information
	PigHealthGWObserved	Device collected from Ear Tags (separated via Apache NiFi to Ear Pig)
	AirQualityObserved	Extra information about environment collected by the Smart Spot Gateway
	NoiseLevelObserved	
Ear Pig	PigHealthObserved	Ear Tag information per pig. It's information is enhanced from analysis

Figure 37: Data models and devices in UC5.6, source: use case interview March 18th, 2020

4.6.2. UC5.6 Data models

The use case tries to use FIWARE Smart Data models. They use the following models (also listed in Figure 37):

- Device
- DeviceModel
- PigHealthGWObserved
- AirQualityObserved
- NoiseLevelObserved
- PigHealthObserved

The two data models for the ear tag information, PigHealthGWObserved and PigHealthObserved are in the process of being defined (see and Figure 38 and Figure 39) and not (yet) openly available as Smart Data models at FIWARE[29].

Device and Device model are used to register details of an IoT Device. For this already FIWARE Smart Models exist, but the UC5.6 models might differ slightly from these models because despite their efforts to harmonize, both FIWARE as well as UC5.6 continued to develop their models.

[29] https://github.com/smart-data-models

Also for AirQualityObserved and NoiseLevelObserved exist already FIWARE Smart Models (as part of the Environmental data model). At this moment we don't know if the UC5.6 models differ from the published FIWARE Smart Models, but we know UC5.6 strives to comply to these models.


```
[
  {
    "id": "urn:ngsi:PigHealthGWObserved:HOPFitPigDemo",
    "type": "PigHealthGWObserved",
    "dataProvider": {
      "type": "Text",
      "value": "www.hopu.eu",
      "metadata": {}
    },
    "observation": {
      "type": "StructuredValue",
      "value": {
        "pigear_id": "c6c8322aad43",
        "pigear_act": 9,
        "pigear_name": "FITPIG_0017",
        "pigear_hr": 91,
        "pigear_hrqi": 8,
        "pigear_bat": 86,
        "pigear_tmp": 39,
        "pigear_rssi": -98,
        "pigear_hrts": 7
      },
      "metadata": {
        "dateObserved": {
          "type": "DateTime",
          "value": "2020-02-24T11:56:32.00Z"
        },
        "description": {
          "type": "Text",
          "value": "FitPig observation value"
        }
      }
    },
    "refDevice": {
      "type": "Relationship",
      "value": "urn:ngsi:Device:HOPFitPigDemo",
      "metadata": {}
    },
    "source": {
      "type": "Text",
      "value": "www.hopu.eu",
      "metadata": {}
    }
  }
]
```


Figure 38: Preliminary model PigHealthGWObserved


```
[
  {
    "id": "urn:ngsi:PigHealthObserved:c6c8322aad43",
    "type": "PigHealthObserved",
    "pigear_act": {
      "value": 9,
      "metadata": {
        "dateObserved": {
          "type": "DateTime",
          "value": "2020-02-24T11:56:32.00Z"
        }
      }
    },
    "pigear_name": {
      "value": "FITPIG_0017",
      "metadata": {
        "dateObserved": {
          "type": "DateTime",
          "value": "2020-02-24T11:56:32.00Z"
        }
      }
    },
    "pigear_hr": {
      "value": 91,
      "metadata": {
        "dateObserved": {
          "type": "DateTime",
          "value": "2020-02-24T11:56:32.00Z"
        }
      }
    },
    "pigear_hrqi": {
      "value": 8,
      "metadata": {
        "dateObserved": {
          "type": "DateTime",
          "value": "2020-02-24T11:56:32.00Z"
        }
      }
    },
    "pigear_bat": {
```


Figure 39: Preliminary model PigHealthObserved