Ref. Ares (2018)4843939 - 20/09/2018
Ref. Ares(2019)5016973 - 31/07/2019

x

Intelligent Fish feeding through Integration of ENabling technologies and Circular principle

Grant Agreement (GA) No: 818036

D2.2 IoT open architecture and new elements specifications

Version :1.0

Date: 01/07/2019

Document type:	**Report**
Dissemination level:	**Public**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 818036

Project data

Project Title:	Intelligent Fish feeding through Integration of ENabling technologies and Circular principle
Project Grant Agreement (GA) No:	818036
Project Acronym:	iFishIENCi
Duration:	48 months, 1 November 2018 – 31 October 2022
Type of action:	Innovation Action

Deliverable Administration and Summary

Status:	Draft	**Due:**	31/04/19	**Date:**	
Author (s)	F. Le Gall (EGM)				
Reviewer	S. G. Prescott (ABT), C. Lai (Sustainable innovation)				
WP	2	**Relative No. in WP**	1, 3		
Comments					

Document change history

Version	Date	Author	Description
0.1	12/03/09	F. Le Gall (EGM)	Creation. Sections 1 & 4
0.2	17/05/09	M. Laffont (BIOC) C. Jensen (OXY) E. Ravagnan (NORCE) F. Le Gall, A. Abid (EGM)	Section 5 Section 4 Section 6 Sections 6, 8
1.0	01/07/20	F. Le Gall	Overall review and cleaning

Disclaimer

The information in this document is as provided and no guarantee or warranty is given that the information is fit for any particular purpose.

This document reflects the author's views and the Community is not liable for the use that may be made of the information it contains.

This document may contain copyright content of certain **iFishIENCi** consortium parties and may not be reproduced or copied without permission. All **iFishIENCi** consortium parties have agreed to full publication of this document. For commercial use of any information contained in this document a license from the proprietor of that information may be required. Neither the **iFishIENCi** consortium as a whole, nor a certain party of the **iFishIENCi** consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk and does not accept any liability for loss or damage suffered by any person using this information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 818036

Table of content

1 Introduction

An ambition of the iFishIENCi project is to provide to the market a flexible iFishIENCi Biology Online Steering System (iBOSS) that significantly improves production control and management for all fish aquaculture systems. iBOSS intends to maximise feed utilisation through smart feeding, providing continuous monitoring of fish behaviour, health and welfare and reducing response times to aberrations. In addition, iFishIENCi targets circular principles and zero waste by qualifying new and sustainable organic value chains for feeds, and valorisation of by-products.

The iBOSS system is a central piece of the iFishIENCi project and also allows to connect information from the Fish-Talk-to-Me project innovation which integrates conventional and emerging technologies for enabling continuous control on fish behaviour, growth, physiology, welfare, health & environmental microbiome

The present deliverable "D2.2 IoT open architecture and new elements specifications" is the first technical contribution of the project and ambitions to provide the baseline for following technical development in the project. It this includes:

- A description of the foreseen innovations, in particular concerning sensing of water and fish related parameters (Fish-Talk-to-Me) (section 2)
- A list of platform requirements extracted from the pilot sites description and building upon the market knowledge of the consortium partners (section 3)
- A proposal of a plan for data modelling, allowing interoperable exchange of information among the different project components (human and IT) (section 4)
- Finally, a concrete technical plan is proposed with a proposal for interfaces protocol (section 5) architecture analysis (section 6), as well as a plan for integration of the components building upon the planned experiments in the development sites (section 7).

2 Innovations to be integrated

2.1 Introduction

From interviews with experimenters and foreseen testers, the iBOSS product is planned to monitor the majority of physicochemical and biological parameters in aquaculture tanks equipped with a RAS system or within open cages. It should come at a minimum additional cost by interoperating with equipment already deployed on sites. The values of interest are diverse (water quality, fish behaviour, management) and have to be collected for process optimisation but also for archival purposes. Whenever possible data will be collected automatically but for some parameters, values could be entered manually in the system.

As an example of planned scale deployment, the Vitafort experiment will take place in Nongteng in a small ponds systems. The farm contains fifteen 10*12meter ponds. During the experiment 4-6 ponds will be used and maximum 500 fish are examined per ponds.

2.2 Essential needs

Regarding the iBOSS product, essential needs are:

- **SENSORS TYPES:** Collect parameters autonomously with underwater sensors. Depending on the size of the tank, the sensors can be placed at different depths. They will be wired until the datalogger.

Figure 1: examples of commercial sensors

- **DATA TRANSMISSION:** get an instantaneous feedback of water parameters. Data must be transmitted in a very short time (about minute) in order to be able to perform actions on the power supply (automation of power processes).
- **SENSOR REQUIREMENTS:** to monitor parameters at high frequency with high accuracy. The sensors connected to the iBOSS system will have to be low energy consumers, with a high accuracy in the measurements made, with little drift.
- **IN SITU FISH SENSORS** (Intradermal fish monitoring): theses sensors can be placed directly in the body of fishes. Sensors transmitted data directly to receivers. This receiver will communicate wirelessly with the sensor.

Figure 2: Foreseen main characteristics for in situ fish sensors.

- **CONNECTIVITY:** the iBOSS system must be connected to the internet (over TCP/IP network) to provide enhanced cloud functionalities (dashboards, etc.)
- **ENERGY:** It is important to consider the electrical energy available on the farms to power the iBOSS system. Some of these farms being installed off-shore, it will be necessary to operate on solar or wind system.

In order to identify the most relevant parameters that will be retained in the final iBOSS product, a wide variety of parameters must be studied first. Only the most relevant parameters will be retained.

All the data that will be collected can be separated into 3 categories:

- Water parameters (includes the physico-chemical parameters of the water)
- Fish parameters (monitoring the fish behaviour and other parameters essential for the fish-talk-to-me product)
- Metadata (essential data from the global environment of the fish, and human actions)

The first set of parameters of interest have been selected regarding different criteria:

- Adviced by the FAO (Svobodová et al. 1993[1])
- Recommended by aquaculturists from the iFishIENCi consortium, providing deployment sites for the iBOSS test
- Availability of the product in the market and innovative nature of the sensors has been taken into consideration. The cost of the sensors is also taken into consideration in the final choice.
- Characteristics of the product corresponding to our requirements: Robustness, Reliability and accuracy.

By grouping these requirements, various parameters were first selected and presented in Table 1.

According to the selected parameters, all are not relevant because they have only very weak influences from RAS farms or are not the most studied for their effects in aquaculture. Some parameters may vary rapidly in the order of minutes or hours (e.g. dissolved oxygen) while others are relatively stable in the RAS system and there are much slower variations (e.g. calcium). To be able to estimate the dynamics of the parameters as precisely as possible, the parameters will be tracked on the highest frequency possible while those with small variations of amplitude will be followed less regularly (hourly or daily).

The number of parameters here is exhaustive and unlimited, the relevance of these parameters will be evaluated to keep only most relevant parameters in the final iBOSS product.

Below, the key for understanding below table.

1 Svobodová, Z.; Lloyd, R.; Máchová, J.; Vykusová, B. Water quality and fish health. EIFAC Technical Paper. No. 54. Rome, FAO. 1993. 59 p.

		Type of System			Technology		
Parameter		RAS	Pound	Open cages	Avaible for online and real time monitoring	Mean of communication	Compagnie
Water parameter	Ammonium	H	H	H	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Calcium	D	D	D	YES	LoRaWan	Bioceanor
	Carbon dioxide	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Chlorophyll a	H	H	H	YES	LoRaWan	Bioceanor
	Chlorides	D	D	D	YES	LoRaWan	Bioceanor
	Colored dissolved organic matter	C	H	H	YES	LoRaWan	Bioceanor
	Water level	C	C	C	To adapt	To adapt	Axis technologies ; Nexsens
	Disolved oxygen	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Fluorescein	H	H	H	YES	LoRaWan	Bioceanor
	fluorides	H	H	H	YES	LoRaWan	Bioceanor
	nitrate	C	C	C	NO	To adapt	?
	pH	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Phycoerythrin	H	H	H	YES	LoRaWan	Bioceanor
	Redox potential	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Rhodamin	H	H	H	YES	LoRaWan	Bioceanor
	Salinity	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Temperature	C	C	C	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	O3 in water	C	H/D	H/D	YES	LoRaWan, ModBUS	Oxyguard ; Bioceanor
	Turbidity	C	C	C	YES	LoRaWan	Bioceanor
	Water currents	C	C	C	YES	To adapt	Axis technologies ; Nexsens
Fish parameter	Activity	C	C	C	NO	Avaible for LoRaWan, to adapt	Thelma Biotel; Star Oddi
	Behaviour	C	C	C	NO	To adapt	Thelma Biotel; Star Oddi
	Distribution in water	H	H	H	NO	camera	HCMR
	Visual appearance	D	D	D	NO	Manually	HCMR
	Heart rate	C	C	C	NO	Avaible for LoRaWan, to adapt	Thelma Biotel; Star Oddi
	Depth	C	C	C	NO	Avaible for LoRaWan, to adapt	Thelma Biotel; Star Oddi
	Temperature	C	C	C	NO	Avaible for LoRaWan, to adapt	Thelma Biotel; Star Oddi
	Parasite loading	W	D	D	YES	To adapt	Aquabyte
	Opercular openning	C	C	C	NO	To adapt	Aquaexcel 2020 project
Metadata	Light	C	C	C	?	To adapt	?
	Feed consumption (feed falling)	H/D	H/D	H/D	NO	To adapt	Aquabyte?
	Water flow input and output	C	C		NO	To adapt	In-Situ
	Room temperature	C	C	C	YES	LoRaWan	Green citizen; auroras; decentlab
	Oxygen input	C	C	C	NO	To adapt	?
	Underwater noise	C	C	C	NO	To adapt	?
	Barometric pressure for room	C	C	C	YES	Wireless	Decentlab
	Rain		C	C	YES	LoRaWan	Green citizen; auroras
	Wind		C	C	YES	LoRaWan	Green citizen; auroras
	Humidity	C	C	C	YES	LoRaWan	Green citizen; auroras

Integration		Frequency
Must	C	Continuous
Should	H	Hourly
May	D	Daily
	W	Weekly

Table 1: Parameters from water, Fish and metadata to be measured by the iBOSS product, regarding the type of system (RAS, Pound,

2.3 Domain specific KPIs

Overall, the iBOSS system is expected to improve a number of business indicators thanks to optimised algorithms applied to the innovative processes being developed within the projects.

Pursued objectives can be classified into 2 categories:

- **Scientific objectives** addressed within the project experiments. These mostly relate to increase the understanding of the processes in play. At that stage, the iBOSS system is expected to store all collected data and to provide querying mechanisms to later analysed the measures. This is expected to develop analytical models that would be the basis of the IA models. As an example, this includes the testing of different methods of inducted spawning (agents); cryopreservation methods; the study of effectiveness of cryopreservation, the larvae rearing methods, the feed utilization, etc. In addition this objectives include the analysis of the environmental impact of the processes.
- **Business objectives**, which mostly turn into economical objectives. These KPIS include mortality level, total weight, feed loss, FCR etc. Turned into economic analysis, the cost of the used feed and the market price of the sold fish provides data to determine economic efficiency.

3 Platform requirements

3.1 Methodology

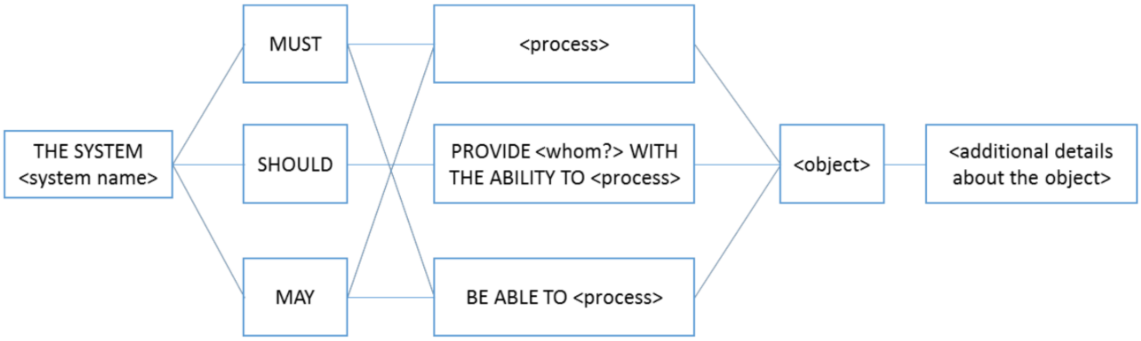
The present section aims at detailing the DAS and IADAS requirements in an unambiguous way and for that purpose suggests expressing them in natural language but making use of standardised approaches and vocabularies aimed at limiting ambiguities. Firstly, priorities will be given among requirements through the use of the following verbs, as defined in the IETF RFC2119[2]:

- MUST -This word, or the terms “REQUIRED” or “SHALL”, means that the definition is an absolute requirement of the specification.
- MUST NOT –This phrase, or the phrase “SHALL NOT”, means that the definition is an absolute prohibition of the specification.
- SHOULD –This word, or the adjective “RECOMMENDED”, means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course
- SHOULD NOT –This phrase, or the phrase “NOT RECOMMENDED” means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY –This word, or the adjective “OPTIONAL,” means that an item is truly discretionary.
- FUTURE –This word means that objectives are provided as guidance or expectation and may or may not be accurate.

Requirements will be listed in tables, each requirement having:

- A unique identifier, for traceability purpose
- A description, using the priority verb and a sentence built as follow:

2 IETF, “Key words for use in RFCs to Indicate Requirement Levels”, RFC2119



- A comment, generally to provide a quick justification for that requirements

3.2 Requirements elicitation

3.2.1 Functional requirements

3.2.1.1 Interfaces

Identifier	Description	Comments
F.I.1	The iBOSS MUST be able to collect data from other cloud services	Appropriate protocols must be implemented
F.I.2	The iBOSS MUST register measurements on water quality/chemistry	
F.I.3	The iBOSS MUST register parameters for fish welfare/behavior	
F.I.4	The iBOSS MUST register measurements on fish health	
F.I.5	The iBOSS MUST have a Dashboard for monitoring/controlling compliant with computer (Windows PC, Mac OS)	Preferably a browser solution for easier cross platform.
F.I.6	The iBOSS MUST take manual registrations	In situations where there is no online sensor for a particular parameter.

3.2.1.2 Local processing

Identifier	Description	Comments
F.LP.1	The iBOSS MUST maintain its local functionality even in case of internet disruption.	Especially if directly controlling the feeders
F.LP.2	The iBOSS Edge MUST include a local Feeder Control for direct control of the feeders	There are several different ways of controlling feeders. The most relevant must be determined and embedded.
F.LP.3	The iBOSS Edge MUST be able to update feeder algorithms from the iBOSS/iFishIENCi cloud	
F.LP.4	The iBOSS Edge MAY include sensors	This could be barometric pressure sensors.

3.2.1.3 Communications

Identifier	Description	Comments

F.C.1	The iBOSS Edge MUST be able to connect to the iBOSS Cloud directly.	It must implement Ethernet and TCP/IP
F.C.2	The iBOSS Edge MAY be able to control feeders over MODBUS	
F.C.3	The iBOSS Edgde SHOULD be able to control feeders with PWM	
F.C.4	The iBOSS Edge MAY be able to control feeders over 4-20mA	
F.C.5	The iBOSS Edge MUST implement a RS485 interface	
F.C.5	The iBOSS Edge MUST implement a RS232 interface	
F.C.6	The iBOSS Edge MUST be evolutive toward new protocols and interfaces	Possibly though additional protocols adapters

3.2.2 Physical requirements

3.2.2.1 Power consumption/Autonomy

Identifier	Description	Comments
PR.P.1	The iBOSS Edge SHOULD have embedded battery backup	For uninterrupted operation during power outages
PR.P.2	The iBOSS Edge MAY be powered through energy harvesting	Solar panel, etc.

3.2.2.2 Safety/reliability

Identifier	Description	Comments
PR.SR.1	The iBOSS Edge MUST comply with EU safety regulations	
PR.SR.1	The iBOSS Edge SHOULD withstand several years of continuous operation	Foreseen mínimum lifetime > 5 years

3.2.2.3 Casing

Identifier	Description	Comments
PR.C.1	The iBOSS Edge MAY include a user interface.	This could be just a small display or a touch screen depending on the type of work done by the gateway
PR.C.2	The iBOSS Edge MUST implement a visual feedback to operator if operation is not running correctly	Red alarm light or acoustic event for example.
PR.C.3	The iBOSS Edge housing SHOULD be at least IP54	
PR.C.4	All casings MUST be corrosion resistant	Plastic casings are prefered.

3.2.3 Software requirements

Identifier	Description	Comments
SR.1	Data to the iBOSS Edge MUST be validated before they are used	This can be, but not limited to, sanity checks, ..., etc.

SR.2	The iBOSS system MUST ensure the integrity of the data received by both Cloud and Edge.	Could be done with encryption.
SR.3	The iBOSS system MUST implement its own security mechanisms	Nor relying no local firewall, etc.
SR.4	The iBOSS MUST use best practice solutions for securing authorization of data sources, data authenticity and data integrity.	OAuth, SSL, AES, Hashing. As software tends to be vulnerable, care must be taken to implement this in a highly secure way. Best practice helps in selecting the correct solution.
SR.5	The iBOSS MUST NOT use any standards/algorithms that are known to have vulnerabilities	For example: MD5, FreeRTOS TCP/IP Stack up until v10.0.1,
SR.6	The iBOSS Edge SHOUD NOT accept any incoming connection.	By forcing the iBOSS Edge to establish all connection, some vulnerabilities can be avoided.

3.2.4 Lifecycle requirements

3.2.4.1 Certification & calibration

Identifier	Description	Comments
LR.CC.1	All hardware SHOULD conform to CE regulations	Not necessary for TRL7, but any end product must conform
LR.CC.2	The iBOSS system MAY identify needs for sensors calibration	

3.2.4.2 Installation

Identifier	Description	Comments
LR.I.1	On field deployment of the iBOSS system should be plug&play	
LR.I.2	The iBOSS Edge MUST support different network provisioning	Static IP, DHCP, wired/wireless
LR.I.3	A portable version of the iBOSS SHOULD be made available	For demo purpose. Consider integrating part of the fish-talk-to-me

3.2.4.3 Maintenance

Identifier	Description	Comments
LR.MA.1	A distant control of the iBOSS MUST be present for maintenance activities	
LR.MA.2	A local interface MUST be present for on-site maintenance	
LR.MA.3	The iBOSS Edge MAY support over the air firmware upgrade	

3.2.4.4 Monitoring & self-test

Identifier	Description	Comments
LR.MO.1	The iBOSS edge MUST verify the connection status on its different interfaces	Include sensors, feeders, iBOSS cloud
LR.MO.2	The iBOSS system MUST generate alerts upon anomaly detection	

4 Data modelling

The development of the iFishIENCi platform and model involves the collection, processing and sharing of data between components and between the platform and external systems. This section presents the background and technologies helping next in designing and developing the iFishIENCi data model.

4.1 Existing Vocabularies and Standards

When designing systems to work with a diverse range of hardware and contexts of use, in a wider environment with multiple stakeholders, ensuring data interoperability is a crucial step. The iFishIENCi platform needs to share data from one context to another while preserving its meaning.

In such cases, where data are coming from different sources, domains and formats, one of the proposed solutions is to use standardised (shared) vocabularies. A vocabulary in the context of digital data, is a set of terms identified and organised by domain-experts, intended to be used to describe data items so that they can be easily be shared. Vocabularies are generally standardised for ensuring a large use and interoperability in data exchange. Vocabularies (and schemas and ontologies, of which more below) tend to be constructed around particular topics –for example, people, or sensors, or aquaculture –and systems will often use multiple vocabularies to cover all of the topics represented in the data they store.

The iFishIENCi platform needs to represent and share data relating to multiple different kinds of entity, and so needs to use vocabularies relevant to each. We briefly summarise here the required topics and relevant existing vocabularies for each, indicating with which ones we have chosen to align the proposed Data Model. Such choices are not necessarily exclusive.

Aquaculture, Fisheries and Marine Environments

The Food and Agriculture Organization (FAO) of the United Nations maintains the AGROVOC[3] vocabulary, which contains terms for over 36000 concepts, covering all areas of the FAO's interest, in food, nutrition, agriculture, fisheries, forestry and the environment, and associated concepts such as location, time, activities, measures, and so on. It is aligned with other vocabularies and ontologies, meaning that there are existing translations between terms in those vocabularies and AGROVOC.

In iFishIENCi , the specific terminology associated with these system productions can be divided in the following major categories: production: open, semi-closed, recirculated systems; fish cultivation: farmed species (fish), feed and feed components, fish pathogens, fish welfare; environment: environmental status, interactions aquaculture-surrounding environment (both ways: water quality, contaminants, waste,), weather conditions.

3 http://aims.fao.org/fr/agrovoc

All these aspects are characterised by specific variables and parameters which must be measured to acquire a detailed and precise status of the system as well as to implement efficient actions. Aquaculture production systems require a mix of measurements that span from the classical manual sampling to detect diseases and sub-lethal effects in fish to precise measurements at the highest possible frequency obtained from online sensors to attest water quality. The biological responses must be linked to physical and chemical multi-sensors, to provide a holistic evaluation of the system status, fish welfare and growth; however, the combination of data so different in magnitude, precision and frequency is not a trivial task. Both the data repository and the model require a high degree of flexibility and adaptability.

To allow a sustainable growth of the aquaculture industry, preserving natural resources and stimulating greater circularity are fundamental steps, achieved through the valorisation of waste and sludge from production systems, approaching the important goal of zero-waste production. For this reason, all the production steps, products and waste must be carefully monitored.

In addition to the production systems and cycles, the aquaculture industry is also defined by specific perceptions from the consumers and general public. In particular, the consumer perceptions, linked to the quality of the product, use of chemicals in the production and the level of fish welfare dictate the demand of specific products, at the end affecting the industry revenues.

In iFishIENCi , the aquaculture production systems considered are:

1. Recirculating Aquaculture System (RAS) on land: the system is based on the re-use of water and use different water treatment techniques to maintain an optimal water quality, which must be continuously monitored. The controlled environment of RAS will be used to develop the technological innovations, which will be then tested in the other production systems.
2. open cages system: cultivation of fish in an enclosed section of the ocean, with complete exchange of water with the surrounding environment;
3. Semi-closed containment systems: in-sea floating containment open at the top to reduce the interactions between farmed fish and the external environment, with water pumped into the system from -20 to -50m to avoid pathogens and parasites located near the surface.

The systems developed within iFishIENCi must therefore be flexible and adaptable to the different production systems and their specific requirement, as well as the requirements of both marine and freshwater species

4.2 Semantic Web technologies

The Semantic Web[4] provides new approaches to manage information and processes on the Web using metadata to describe Web data. The Semantic Web brought with it new technologies such as new meta-formats. These meta-formats represent metadata in a format that can be processed by machines to infer additional information, to allow for data sharing and for interoperability amongst Web pages. Most of the Semantic Web meta-formats are built on XML. In order for data to be linked, the concept of a vocabulary comes to the fore, so that data published in one location and be linked to, read, and understood by external users (or applications) at another.

RDF[5] (Resource Description Framework) provides a standard syntax to create, exchange and use resources in the semantic Web. Resources are described in RDF in the form of triple statements which

4 https://www.w3.org/standards/semanticweb/

5 https://www.w3.org/RDF/

are "Subject", a "Predicate" and an "Object". The RDFS or RDF vocabularies, describe classes and properties using the RDF vocabulary description language. Within RDF, several serialisation methods were proposed. Json-LD[6] was proposed also as new method for encoding linked data using Json and thus to get closer to developers using Json.

Ontology[7] is defined as a formal, explicit specification of a shared conceptualisation. It provides a shared common vocabulary, including important concepts, properties and their definitions, and constraints, sometimes referred to as background assumptions regarding the intended meaning of the vocabulary, used in a domain that can be communicated between people and heterogeneous, distributed application systems.

The Simple Knowledge Organization System[8] (SKOS) is a common data model for sharing and linking knowledge organization systems via the semantic Web. SKOS is itself expressed in RDF –every valid SKOS document is a valid RDF document and is interoperable with the full Semantic Web technology stack. The SKOS vocabulary offers several classes for linking concepts and creating concept hierarchy such as `skos:broader` and `skos:narrower`, and for external alignment such as `skos:exactMatch`, `skos:broadMatch`. The use of SKOS-XL as a modelling language presents several interests:

- The thesaurus approach is easy to understand for domain experts
- SKOS allows to express linked data relations and to define hierarchy based on concepts and properties (broader/narrower, related)
- The SKOS-XL extension allows to define preferred/alternate labels, languages…
- This approach is used by the FAO which would ensure sustainability of the project contributions
- A web platform (vocbench) allows easy co-edition over the web.

4.3 Analysis of FAO thesaurus

The AGROVOC thesaurus if the FAO is defined as a SKOS-XL model. The AGROVOC vocabulary has been developed originally for the agricultural sector and, even if many concepts are common to several primary industries and sectors, it needs to be updated and adapted to include concepts linked specifically to fisheries and aquaculture. The specific requirements of aquaculture are not only the typical farmed species, from algae to carnivorous fish, but also the type of farming systems, from open cages at sea to closed systems with recirculated water. Each of the systems have its own specific set of characteristics and requires the appropriated vocabulary. The AGROVOC vocabulary will be integrated with other vocabulary, such as “Aquatic Sciences and Fisheries Thesaurus” from FAO, to develop further the aquatic section of the vocabulary. To allow these activities, contacts have been initiated with the FAO to expand their model.

4.4 iFisIENCi data model

As outlined in the DoA, iFishIENCi is going to build upon the FIWARE ecosystem. This ecosystem includes a web services architecture communicating through a service bus. The interfaces are based on the OMA NGSI specification which evolved within the FIRWARE project toward NGSI v2 and is now

6 https://json-ld.org/

7 https://www.w3.org/standards/semanticweb/ontology

8 https://www.w3.org/2004/02/skos/

being further extended to integrate the semantic dimensions within the ISG-CIM specification group of the ETSI. In this context the NGSI-LD[9] was proposed. The NGSI-LD metamodel extends the RDF/RDFS grounding model with new concepts including Entity, Relationship of Entities, Property and Value. The metamodel is also extended to the Cross-Domain level, which includes other concepts. The NGSI-LD is represented in Json-LD[10]. The NGSI-LD specification includes also an API[11] that supports querying data presented in the NGSI-LD data model. It offers Batch and temporal operations. The NGSI-LD metamodel, Cross domain ontology and API are detailed in Section 8.

The iFishIENCi project covers several domains that must respect the NGSI-LD specification. Domains covered by the iFishIENCi project are detailed below:

To cover all these domains, we organize the data according to their domains. We distinguish as four main domains detailed below:

- **Data Source**: Every entity that provides data is considered as a data source. For example, devices, sensors, cameras, gateways, satellites (providing images or weather data) are considered as data sources. Several standardised ontologies are defined and standardized, covering different layers of the Web of Thing architecture. Data sources in the iFishIENCi data model will be aligned with the SSN[12] (Semantic Sensor Network) and SOSA (Sensor, Observation, Sample, and Actuator) ontologies
- **Services/States**: In aquafarms several services are offered. In order to automate the process of their execution services, their states and their properties (such as Input, preconditions....) must be modelled. Example of Services: Feeding, Heating, Sampling....
- **Aqua-Thesaurus**: This concerns all technical and scientific aquaculture terms. In fact, when a data source returns a measure, it must refer to its technical reference modelled in the aqua-thesaurus. The iFishIENCi will be aligned with the AGROVOC vocabulary presented in section 4.3.
- **Individuals:** The individual entity will follow the Foaf[13] standardised ontology for modelling persons, individuals and organization present in aquafarms. It will be used also for specifying administrators that can activate a service or a data source.
- **General Purposes:** The core distinctive data features of the iFishIENCi project are generally covered by the above vocabularies –domains relating to aquaculture and the marine environment, sensors, observable parameters, and units. There are several other concepts which are useful across many domains and for which well-established standards exist, such as the unit ontology [14], time ontology[15]….

In each presented domain there exist several standardised ontologies. Thus, the iFishIENCi data model will extent the NGSI-LD cross domain ontology for the aquaculture which will allow to make use of the NGSI-LD API.

9 https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf

10 https://json-ld.org/

11https://forge.etsi.org/swagger/ui/?url=https://forge.etsi.org/gitlab/NGSI-LD/NGSI-LD/raw/master/spec/updated/full_api.json

12 https://www.w3.org/TR/vocab-ssn/

13 http://xmlns.com/foaf/spec/

14 http://qudt.org/

15 https://www.w3.org/TR/owl-time/

5 NGSI-LD interface specification

As the iFishIENCi data model will be aligned with the NGSI-LD specification, in this section we detail main concepts of NGSI-LD data model and API.

NGSI-LD is represented in JSON-LD and thus should have a grounding in RDF. It is mainly based on RDF standards to capture high-level relations between entities (i.e. IoT devices, group of devices or non-IoT information) and properties of entities, as shown below. The core concept in the NGSI-LD data model is the "Entity" which it can have properties and relationships to other entities. An entity is equivalent to an Owl class. The assumption is that the world consists of entities, which can be physical entities like a car or a building, but also more abstract entities like a company or the coverage area of a WLAN access points. Entity instances are identified by a unique URI and a type, e.g. a sensor with identifier `urn:ngsi-ld:Sensor:01` and of type Sensor. Different from rdf:Properties, NGSI-LD properties (and relationship) are also considered as owl classes also. Properties and relationships can be annotated by properties and relationships themselves, e.g. a timestamp, the provenance of the information or the quality of the information can be provided. The `hasObject` and `hasValue` in the NGSI-LD metamodel are defined to enable RDF reification, based on the blank node pattern, to leverage the property graph model.

The NGSI-LD cross domain ontology extends the NGSI-LD metamodel to cover several general contexts presented below[16]:

- **Mobility** defines the stationary, movable or mobile characteristics of an entity;
- **Location** differentiates and provide concepts to model the coordination based, set based or graph-based location;
- **Temporal specification** includes property and values for temporal property definitions;
- **Behavioural system** includes properties and values to describe system state, measurement and reliability;
- **System composition and grouping** provides a way to model system of systems in which small systems are composed together to form a complex system following specific patterns.

The NGSI-LD cross domain ontology is presented in Figure 1.

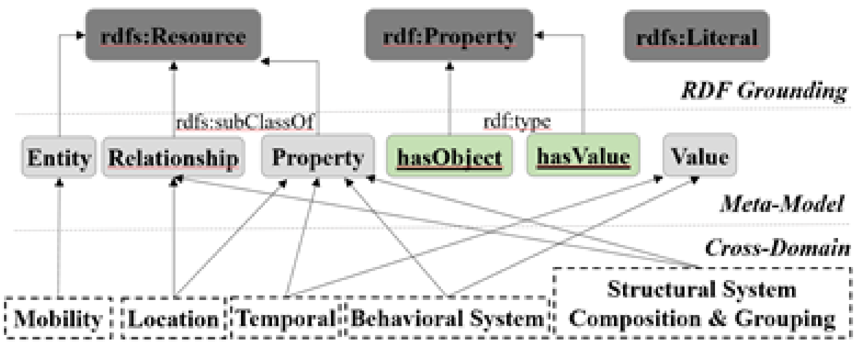


Figure 3: RDF standards to capture high-level relations between entities in NGSI-LD

Below we present a use case example for modelling data and context using the NGSI-LD. The example consists on a sensor that returns measure of dissolved oxygen. This sensor has an id which is

16 Review of Standard Ontologies for the Web of Things: W.Li, G.Tropea, A.Abid, A.Detti, F.Le Gall

`urn:ngsi-ld:Sensor:001`. This sensor is attached to a buoy identified by `urn:ngsi-ld:Buoy:001`. This sensor has a measure and measure accuracy (literals).

To model this example, Figure 2 presents main symbols signification used in the medialisation task.

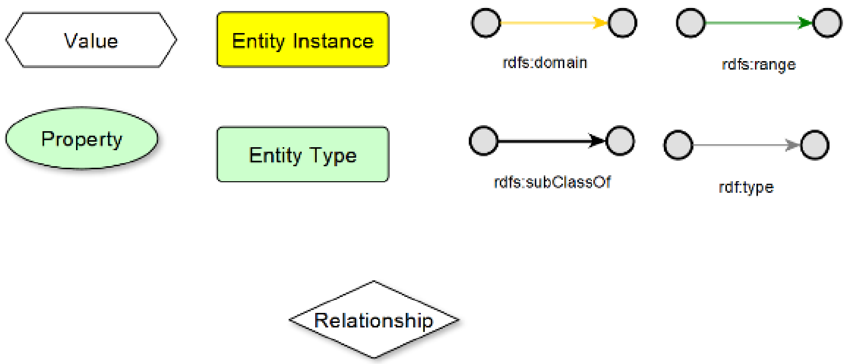


Figure 4: Main Symbols Definition

This example refers to two domains which are data sources (sensor) and Aqua-Thesaurus (dissolved Oxygen and Buoy). As announced in the previous section, iFishIENCi data model will cover several domains and follow the NGSI-LD data model specification at the same time. For this purpose, the domains data sources and Aqua-Thesaurus in this example are considered as subclasses of the NGSI-LD entity.

The Entity sensor (since it is a subclass of NGSI-LD Entity) is instantiated with the sensor `urn:ngsi-ld:Sensor:001`. Parameters such as `hasMeasurement` and `measureAccuracy` are added then to this entity in order to show the value.

2 relationships are defined in this example the first (`measurementParameter`) is a relationship between the sensor and the class Dissolved Oxygen. Dissolved Oxygen is a part of the aqua-thesaurus. The second relationship (`connectsTo`) was initially proposed in the NGSI-LD cross domain ontology as a location relationship. We are using this relationship in order to express that the sensor `urn:ngsi-ld:Sensor:001` is attached to the buoy `urn:ngsi-ld:Buoy:001`.

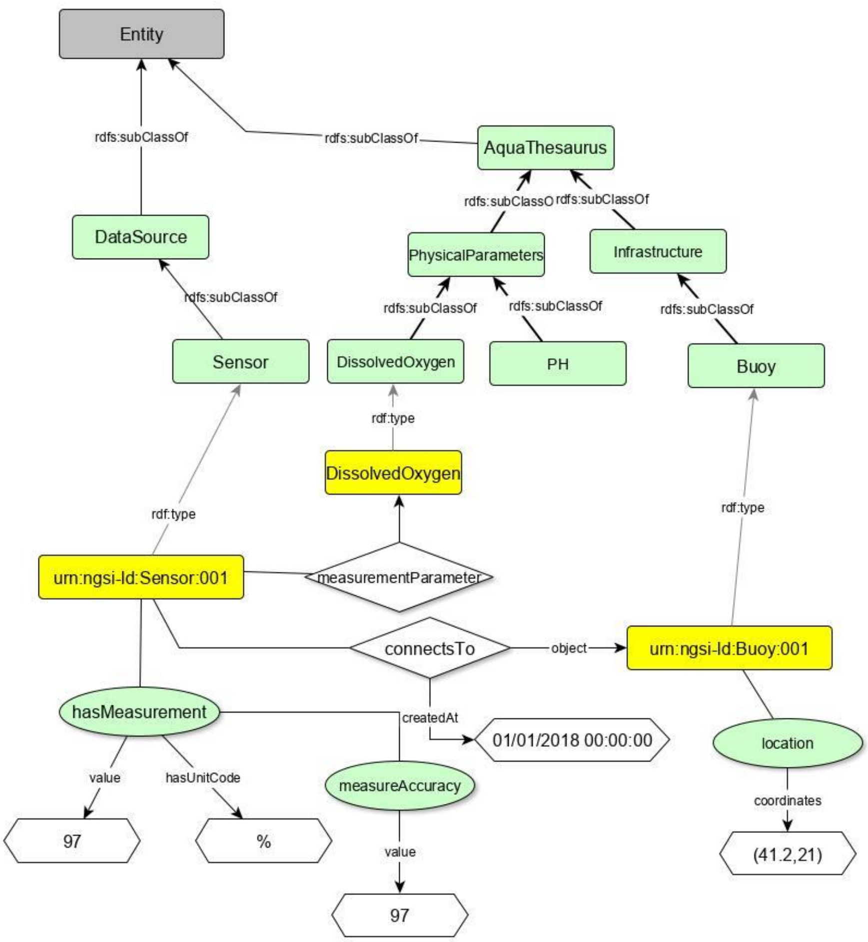


Figure 5: Illustration of NGDI-LD usage

NGSI-LD Data API

The NGSI-LD API supports several operations, with messages expressed in JSON-LD. The API is the standard for management of context information (which can be summarised has being any piece of information associated to a context such as time-location information). The Overall NGSI-LD API operations include:

- **General Operations**
  - Entity create
  - Entity update
  - Entity partial update
  - Entity delete
  - Entity retrieval
  - Queries
  - Subscriptions
- **Registry Operations**
  - CSRegistryEntry create
  - CSRegistryEntry update
  - CSRegistryEntry partial update
  - CSRegistryEntry delete
- **Batch Operations**
  - Batch Entity Creation
  - Batch Entity Create/Update (Upsert)
  - Batch Entity Update
  - Batch Entity Delete
- **Temporal Operations**
  - Create/Update Temporal Entity Representation
  - Add Attributes to Temporal Entity Rep.
  - Delete Attribute from Temporal Entity Rep.
  - Modify Attribute Instance in Temporal Entity Rep.

- CSRegistryEntry retrieval
- CSRegistryEntry query
- CSRegistryEntry subscription
- Delete Attribute Instance from Temporal Entity Rep.
- Delete Temporal Entity Representation
- Retrieve Temporal Entity Evolution
- Query Temporal Entity Evolution

Figure 6: NGSI-LD operations

This API relies on the NGSI-LD data model introduced in section 4. In short, this model makes use of the JSON-LD serialisation format which adds linked data capabilities to the JSON format. The core of the model builds upon the concept of **Entity** where an entity can have **Properties** and **Relationships** to other entities, building a property graph model.

The JSON-LD format allows to create a network of standards-based machine interpretable data across different sources. The JSON-LD format includes an @Context clause used to map short terms used in the serialization to URIs uniquely identifying concepts and mapping to specific types (e.g. Date Time).

In the following, we present the modelling process of the previous example using the NGSI-LD API based on Json-LD messages for creating and querying instances of Sensor and Buoy.

Create Buoy Instance

Below we show an example of NGSI-LD query for creating an instance of a Buoy. The header part contains details about the query type, address and the content-type. The @context, refers first to the NGSI-LD core context, where main concepts of the NGSI-LD are defined. Then, main used concept such as the type Buoy and Location are referred. The parameter "id" is used to identify the instance.

```
POST /ngsi-ld/v1/entities/ HTTP/1.1
 Host: localhost:9090
 Content-Type: application/json
Accept: application/ld+json
 {"@context": [
  "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  {"Buoy": "https://data. iFishIENCi .eu/core/Equipment/Buoy",
   "location": https://data.iFishIENCi .eu/core/Position/location }
  ],
 "id": "urn:ngsi-ld:Buoy:001",
"type": "Buoy",
"Location": {
"type":"Property",
"value":{
          "coordinates": {
                          "alt":"41.2",
                          "log":"21.0"     }
 }}}
```

Create Sensor Instance

Here we show the main content of the Json-LD message for creating a sensor instance attached to the created buoy. Thus, we define the relationship connectTo in the @context, then we add the id of the buoy as a value to its parameters. We can also add other parameters defined by the NGSI-LD such as createdAt.

```
…{
"@context": [
```

```
  "http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  {
  "Sensor":"https://data.iFishIENCi .eu/core/Equipment/Sensor",
  "location":"https://data.iFishIENCi .eu/core/Position/location ",
  "connectsTo":"http://uri.etsi.org/ngsi-ld/Relationship/connectsTo",
  "DissolvedOxygen":"https://data.iFishIENCi
.eu/core/PhisicalParameter/DissolvedOxygen",
  "ValueType" : "https://data.iFishIENCi .eu/core/ValueType"} ],
"id": "urn:ngsi-ld:Sensor:001",
"type": "Sensor",
"connectsTo" : { "type": "Relationship",
                 "value": "urn:ngsi-ld:Buoy:001",
                 "CreatedAt":"01/01/2019 00:00:00"},
"ValueType":{  "type": " Relationship ",
               "value" :{
                         "type" : "DissolvedOxygen" }
    }}
```

Retrieve Entity by Id

Here, we show an example of retrieving entity by Id query:

```
GET /ngsi-ld/v1/entities/ urn:ngsi-ld:Buoy:001 HTTP/1.1
Host: localhost:9090
Content-Type: application/ld+json
Accept: application/ld+json
```

Delete Entity by Id

```
DELETE /ngsi-ld/v1/entities/urn:ngsi-ld:Sensor:001 HTTP/1.1
 Host: localhost:9090
 Content-Type: application/json
 Accept: application/ld+json
```

6 Overall architecture

6.1 Introduction

A number of requirements have been identified in section 3.2 and overall the iBOSS has to provide a local control of the feeding process building upon advanced algorithms to optimise that feeding process. Optimisation of the algorithms is intended to make use of machine learning approaches which implies having access to local and distant information as well as calculation placement agility to optimise time and energy consumption on the edge. Indeed, the iBOSS system is planned to be deployed in Recirculating Aquaculture Systems (RAS) where power and network connectivity exist on-site. However, some interests have been raised to make sure of the iBOSS system in open area context where connectivity resources may be scarcer.

The aquaculture automation traditionally builds upon RS485/Modbus standards family (see for example: Oxyguard Pacific suite[17], YSI aquaculture[18], etc.).

Serial Modbus is a serial communications protocol originally developed for use with programmable logic controllers (PLCs). Simple and robust, it has since become a standard communication protocol,

17 http://www.oxyguard.dk/products/stationary-oxygen-systems/pacific/the-pacific-system-overview/

18 https://www.ysi.com/ysi-blog/water-blogged-blog/2017/06/aquaculture-facilities-benefit-from-automation

and it is now a commonly available means of connecting electronic devices in industrial installation. The main reasons for the use of Modbus in the industrial environment are:

- Developed with industrial applications in mind.
- Openly published and royalty-free.
- Easy to deploy and maintain.
- Moves raw bits or words without placing many restrictions on vendors.

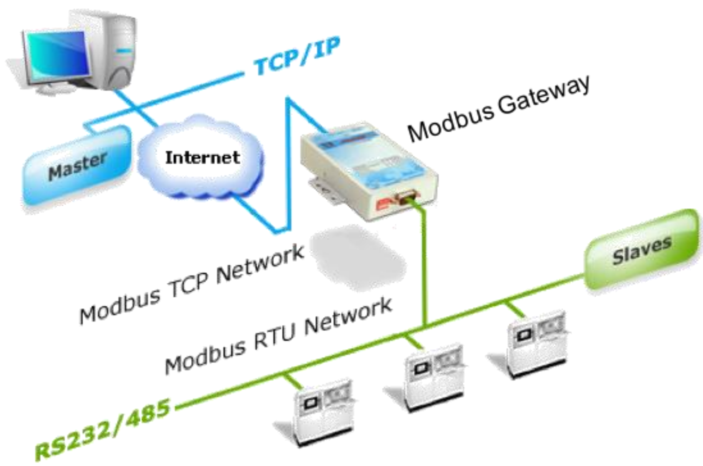


Figure 7: General illustration of a Modbus network

Each device intended to communicate using Modbus is given a unique address. In serial and MB+ networks, only the node assigned as the Master may initiate a command. A Modbus command contains the Modbus address of the device it is intended for (from 1 to 247). Only the intended device will act on the command, even though other devices might receive it. All Modbus commands contain checksum information, to allow the recipient to detect transmission errors. The basic Modbus commands can instruct an RTU to change the value in one of its registers, control or read an I/O port, and command the device to send back one or more values contained in its registers.

The Modbus messages have different formats for communicating via Serial (RTU) and via TCP/IP. For Serial communication, the message is composed by a header, function code, data and CRC. The header of the message only takes the SlaveID in the RTU case. It needs the CRC check to ensure data integrity. For TCP/IP the header contains three more parameters, the Transaction ID, Protocol ID and Length of the message. It also contains the UnitID that corresponds to the Slave ID in the RTU protocol. The function code and data are the same for both protocols. In the TCP/IP communication there is no need for the CRC bytes because the TCP protocol already has built in error detection to guarantee data integrity.

However, Modbus suffers from a number of limitations inherited from its history. These includes a limited capacity to handle complex object or deal with data semantics. However, its main constraints lie into it master/slave approach which implies that the device cannot decide by itself when to report a data or an alert. It has to be pulled by the master which means for the device to be constantly listening. This is not compliant with the developing Internet of Things (IoT) model which includes many energy autonomous devices. These devices need to save energy and avoid being in constant listening mode but rather wake from time to time to send their data and asking if there is any downlink message (such as reconfiguration parameters) to be downloaded. This is the classical approach for Class A devices specified by the LoRaWAN Alliance which defines listening windows for the devices after each sending. The LoRaWAN is a typical IoT standard building upon the LoRa radio specification which permits long distance (2 to 15 km) communication of small payloads (<50 bytes) with a very small power consumption. However, many other protocols are being developed: Modbus/TCP now provides full duplex communication using Modbus protocol, OPC-UA is gaining in popularity in the industry as a service oriented architecture[19], etc. The Figure 8 provided by the Alliance for IoT Innovation[20] illustrates the complexity of standards in the IoT area and this complexifies day after day as new standards emerge. **Capability to interoperate with many standards to provide a unified service is thus a primary objective for any IoT systems and has to be included within the iBoss architecture to make it an evolutive platform.**

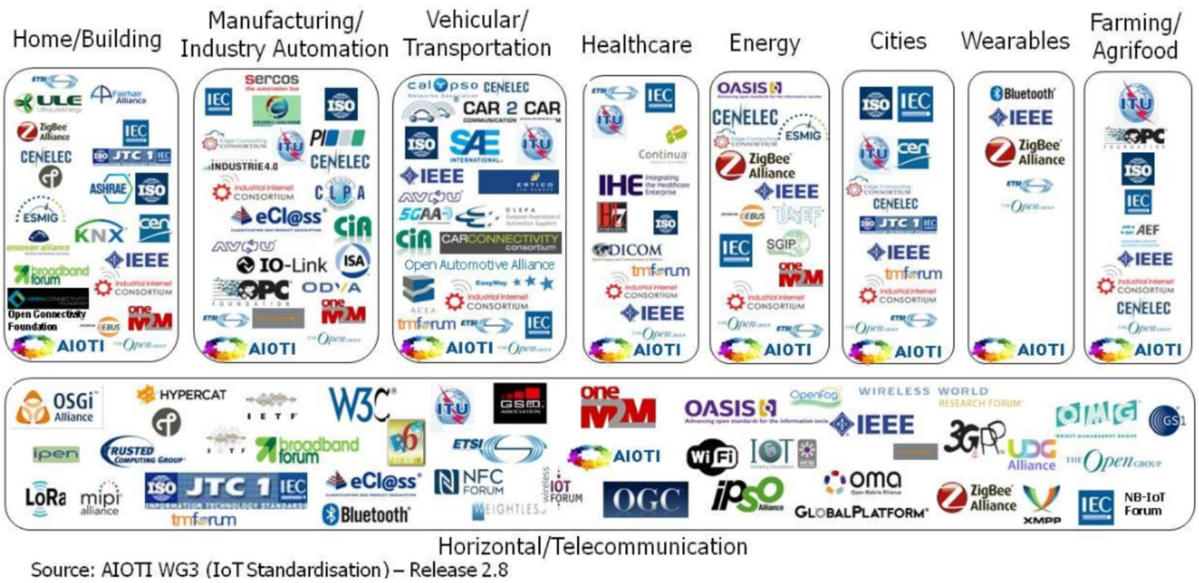


Figure 8: IoT standardisation landscape (source AIOITI-2017)

6.2 Architecture overview

The Figure 9 provides a representation of the iBOSS positioning. It can be seen as a box which locally connect to available sensors and actuators (i.e. feeders) to provide smart services locally (on the edge). In the iFishIENCi context, the collected data serves 2 populations:

- The fish farmers to monitor and control their processes. For that purpose, a local interface has to exist (embedded screen, tablet, etc.).
- The data scientists to collect all information in a controlled way.

A connection to the cloud allows:

- To share data set in an interoperable way
- To update iBoss control algorithms based on new parameters calculated in the cloud.
- Gather additional information such as weather predictions

Four main interfaces have been identified in the iBOSS context:

19 https://opcfoundation.org/about/opc-technologies/opc-ua/

20 https://aioti-space.org/wp-content/uploads/2017/06/AIOTI-WG3_sdos_alliances_landscape_-_iot_lsp_standard_framework_concepts_-_release_2_v8.pdf

- Interface 1 is low level to adapt to the various protocols used in the sensors and actuators area (Modbus, LoRaWAN, PWM, 4-20mA, I2C, etc.)
- Interface 3 and 4 build upon the http REST NGSI-LD API (see section 5) which allows context based representation of the data and offers extensive geotemporal queries.
- The interface 2, between the iBOSS and the cloud basically builds also on http REST NGSI-LD but alternatives such as MQTT may be proposed to ease ingestion of the data stream.

Security has to be implemented on all interfaces. OAuth2 authentication mechanisms are well developed for web services and will be used as a basis. The authorization framework may build upon simple Access Control Lists (ACL) or more atomic managements using the semantic ontologies.

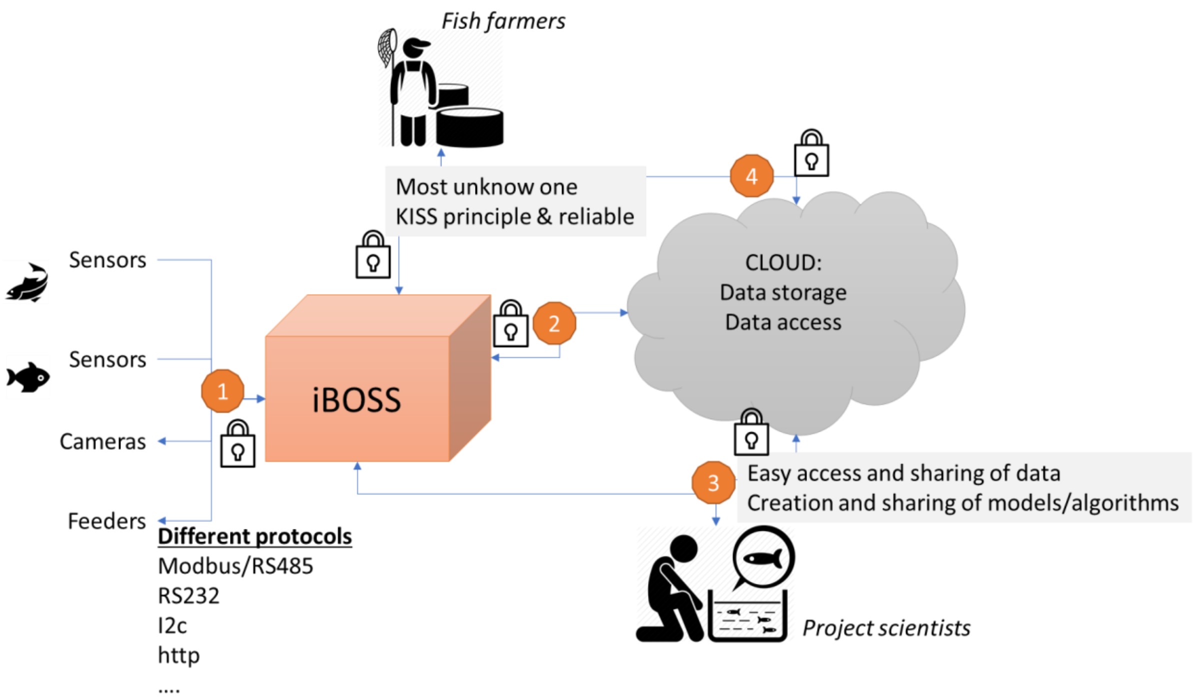


Figure 9: iBOSS positioning

6.2.1 iBOSS Cloud architecture

In more details, the iBOSS cloud platform has to include several components:

- A bus service to ensure exchange of data between the different cloud services, following a microservice based architecture
- A security server in charge of authentication and authorization of the different persons or services to access a given service
- Database to provide persistence of the collected data.
- Interface for querying, storing and updating the data
- Device management module for provisioning, monitoring and maintained the deployed devices.
- Cloud-edge orchestration module allowing the distribution of services between the cloud and the edge to achieve the edge processing paradigm.
- Data analytics modules for stream processing (i.e. complex event processing) or more advanced (i.e. Neural network).

6.2.2 iBOSS edge architecture

On its side, the iBOSS edge would be a light replication of the cloud architecture. It will include

- A security layer controlling access to the different interfaces
- Storage for configuration, data logging, etc.
- Processing capabilities: with possibly extended calculation (i.e. GPU) capabilities to provide on-board video analysis.
- User interface: such as an embedded screen or mobile app.
- Communication interfaces:
  - Northbound: toward cloud services and local mobile application
  - Southbound: toward sensors and actuators deployed on site

Figure 10: possible hardware option for the local iBOSS.

7 Building blocks development and deployment planning

Several of the core components are already available as part of partners portfolio:

- Oxyguard proposes the Pacific system, a PLC aggregating data from local Modbus oxyguard sensors with a local graphical interface for reading and configuration. A web-based dashboard is under deployment
- Bioceanor proposes a web service providing predictive analytics to fish farmers. It is vendor neutral in respect with sensors but has developed a specific offer for long distance low power (LoRaWAN) sensors connectivity.
- EGM proposes integration services for IoT sensors and data platforms bases on open standards. It can deploy the core components of the cloud and edge system.
- HCMR proposes a video analysis system that estimates fish movement parameters (velocity vector) correlated to behavioural characteristics, using live or stored video streams from IP cameras placed in fish tanks. The software will be running on a dedicated processing unit and the results will be stored in the iBOSS cloud data base.

These components will be integrated first to provide the data collection infrastructure allowing experimenters to collect and analyse their data.

A number of components remain to be developed and integrated to reach the ambition level of the iFishIENCi project in terms of embodied intelligence. These components include the specific (fish species, feeding regime, etc.) domain models able to provide targeted recommendations based on actual measurements made in the farm. The technological approach is to implement Digital Twins being digital representation of a real artefact. These digital twins can provide information about the actual state of the artefact but also provide model based short term predictions or multiparameter optimisations. They rely on models such as FishMET.

Looking at the project time plan which foresee an initial deployment of the core platform including initial AI building blocks by M18 (D2.3), demonstrations running from M30 (D3.1) and a final report on the smart feeding implementation by M36 (D2.4), a 3 waves (see Figure 11**Erreur ! Source du renvoi introuvable.**) deployment plan is foreseen :

1. **First wave**: existing systems are integrated by the technology partners, building upon the specific iFishIENCi data model. The proposed solution allows to connect existing sensors from

the pilot sites and experimenters to query aggregated data and start development of models. **Planned availability: M12 (October 2019).**

2. **Second wave**: the iBOSS edge initial implementation is deployed on site with a software container-based orchestration from the cloud. **Planned availability M16 (February 2020).**
3. **Third wave**: initial smart functions deployed in the cloud. **Planned availability M18 (April 2020).**

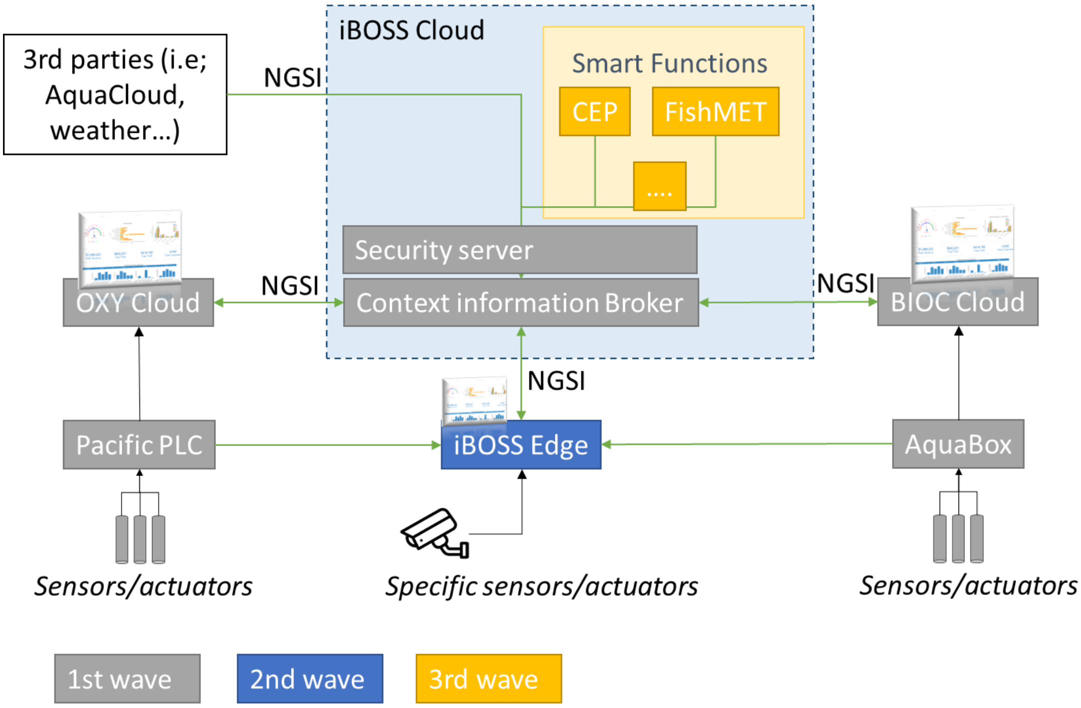


Figure 11: iBOSS deployment steps

8 Conclusion

This document produces a first analysis of the iBoss system. It clarifies the positioning in the edge (iBoss edge) and the cloud (iBoss cloud) of the system in respect with other iFishIENCi components and actors. The main system requirements have been identified and functions to be embedded have been listed. A standardised API building upon a semantic model is proposed for exchange of context information within cloud services. This data model expand work from the FAO with whom a formal contact has been established. An initial deployment plan in 3 waves is proposed to deliver the iBOSS system to the experimentation and will be the basis of forthcoming activities in WP2.