Intelligent Fish feeding through Integration of ENabling technologies and Circular principle

Grant Agreement (GA) No: 818036

D2.1 iFishIENCi Data

Management and Access

Version: 4.0

Date: 10/09/2020

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

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:	30/04/2020	Date:	
Author (s)	E. Ravagnan (NORCE)				
Reviewer	F. Le Gall, Tamás Bardócz				
WP	2	Deliverable Nr.	D7	Relative Nr.	D2.1
Comments					

Document change history

Version	Date	Author	Description
v1	30/3/2020	E. Ravagnan, A. Abid, C. Dupont, F. Le Gall	First draft
V2	24/04/2020	E. Ravagnan, A. Abid, F. Le Gall	Corrections and improvements after revision
V3	14/05/2020	Tamás Bardócz	Final check and edits on the disclaimer
V4	08/09/2020	Elisa Ravagnan, A. Abid, F. Le Gall	Revision following EC reccomendations: added description of interactions between iBOSS and MetFish (Fish Digital Twin) in Section 4.2 Page 16

Disclaimer:

Contents

1 Introduction

The main aim of the iFishIENCi project is to deliver breakthrough innovations for sustainable aquaculture, supported by enabling technologies, the implementation of smart feeding and smart breeding, and circular principles. All these aspects provide the European aquaculture industry with the competitive advantage and growth stimulation needed to be forefront in revolutionizing global efficiency in fish production and meet society's needs for food from the ocean. These ambitious tasks are achieved through the implementation of the iFishIENCi Biology Online Steering System (iBOSS), significantly improving production control and management for all fish aquaculture systems. iBOSS maximises feed utilisation through smart feeding, provides continuous monitoring of fish behaviour, health and welfare and reduces response times to aberrations. iFishIENCi also targets circular principles and zero waste by qualifying new and sustainable organic value chains for feeds, and valorisation of by-products. iFishIENCi´s innovations provide important new assets to fish-farmers, feed producers and technology providers in the aquaculture sector, thus leading to market growth and job creation.

The present document describes the iFishIENCi project data management platform and procedures to collect and access the data, following the guidelines described in the iFishIENCi Data Management Plan (DMP).

2   Data management

2.1   Description of data type and format

A large number of data are being generated and collected during the iFishIENCi project, presenting an elevate degree of heterogeneity. A general classification is provided below.

The research data can be classified as qualitative data, when it refers to text, pictures, video, observations, ranges, or quantitative data, when it refers to numerical data. Also, data can be categorized as:

- **Measurement Data**: Unique data that is collected in real time and cannot be reproduced (such input data from sensors);

- **Experimental Data**: Data derived from laboratory equipment that is subject to controlled conditions (such as gene sequences);

- **Survey Data:** Data derived from surveys and interviews.

- **Simulated Data**: Data generated from the simulations of test models that study real or theoretical systems (such as climate models and digital twins);

- **Derived Data:** Data that is a result of the analysis of data or aggregates from various sources.

Table 1: Summary of data collected; more details on the data in D2.1, D4.1, D6.4, D6.6 and D6.8.

Data origin	Parameter collected	Main Storage Format
Environment (water and/or sludge)	Ammonium, Nitrite, Nitrate, Carbon Dioxide, Organic Carbon, Conductivity, Dissolved Oxygen, pH, RedOx, Salinity, Water Temperature, Turbidity, Light, Water Level, Water/Sludge Microbiome	.xls, .csv, .json
Fish	Activity/behaviour, Diving Depth, Appetite, Satiety, Food intake (modelled), Biomass, Nr. Of Fish, Average Weight, Gut/Gills/Skin Microbiome, Initial amount/Uneaten Feed, Feed ingredients/feed formulation	.mp4, .xls, .csv
Socio-economic	Attitudes towards seafood products/aquaculture, willingness to pay, social acceptance, sustainability, socio-demographic parameters	.xls
Stakeholders lists	Aquaculture Producers, Technology Suppliers, Feed Producers, Seafood Retailers, Research Organisations, NGOs, Sector Associations, Marine Clusters, Advisory Councils, Public Authorities, Policy Makers	.xls, .doc

2.1.1 FAIR principles in iFishIENCi

In iFishIENCi, making research data **findable, accessible, interoperable and re-usable (FAIR)** is an integral part of the process of open science and research, following the directives for Open Access of Publications and Research Data in the H2020 projects and in the participation on Open Research Data Pilot.

Data and metadata in iFishIENCi:

- Have a unique, persistent and global identifier, associated with their origin, for easy recovery;
- are described with well-founded metadata, including qualified references to other data and metadata;
- are registered/indexed in repositories (if allowed by privacy and IPR issues);
- metadata will be accessible for 5 years after the end of the project.
- use a formal, accessible, shared and wide applicable language for knowledge representation;
- are aligned with the community standards relevant to their domain.

2.2 Datasets and data storage

The data collected during the iFishIENCi project will be stored in several dataset following the guidelines described in the iFishENCi Data Management Plan:

- A copy of each dataset is stored locally by the data collector/analyst and a copy is stored in the iFishIENCi portal (the double storage serves as backup to lower the risk of data loss);
- Data are stored in digital, user-friendly and wide-accessible formats;
- Files are identified with descriptive names (partners, intuitive name) and the date;
- A constantly update list of datasets is stored in the iFishIENCi portal.

2.2.1 iFishIENCi datasets

The data collected and/ or generated by iFishIENCi are assembled in separate datasets with different purpose and access policies.

In this section the datasets created in iFishIENCi at M18 of the project are summarised.

Table 2. Summary of datasets in iFishIENCi at M18

Type	Nr.	Description	Responsible
Project Administration	2	Project Contact Data; Project Administrative Data	ABT
Field and laboratory trials	7	RAS trials: ABT, HCMR, SZIU, GE; net-pen trials: HCMR; breeding/ selection of fish	ABT, HCMR, GE, SZIE

Feeds and components	6	Evaluation of ingredients and feed formulations for different species; Microbial communities from feed trials; Microalgae production; Fungi production; Fish feed raw materials and manufacturing processes.	AAR, LEITAT, NORCE, Vitafor,
Sustainability and circularity	1	Characterisation of waste and dirty water	LEITAT
Socio-economic analysis, Stakeholders engagement	5	Organoleptic and nutritional values of fish; Social acceptance analysis; Mapping Stakeholders and events; Exploitation Plan questionnaire	TTZ, NORCE, SIE
Modelling and data analysis	2	Data future climate projections from climate models, fish digital twin output (fish behaviour and performance)	NORCE, UiB

2.3   Interface to data

Data collected within the iBOSS will be primarily utilized over web REST interfaces by other programs, such as dashboard form Oxyguard and Bioceanor clouds, and the digital twin software FishMet, to provide aggregated figures to end users. For research work, most of the project partners will simply rely upon an export of data to a CSV file which can be consumed into standard programs as R or SPSS.

Some partners requested to have an interface to ease filtering, sorting and displaying of data within an interface with some plotting capabilities. For that purpose, the integration of the Knowage[1] enabler from the FIWARE community will be considered. It will however require the update of its interfaces from NGSIv2 to NGSI-LD.
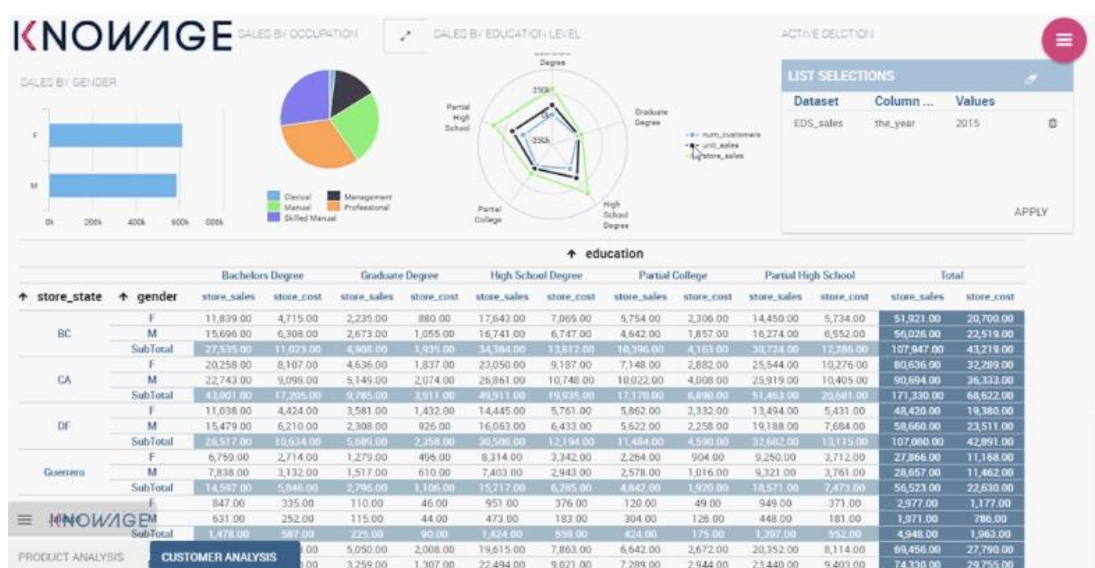

Figure 1: envisioned interface for friendly access to iBOSS data.

---

[1] https://knowage.readthedocs.io/en/latest/

3   Database structure and access

To implement the iBOSS tool developed within iFishIENCi and to allow comparative analysis of specific parameters, several datasets will be merged through the iBOSS platform, stored in partner private clouds.

3.1   iBOSS Cloud Platform

One of the main functionalities of the iBOSS cloud platform is to manage scientific data by providing tools for persisting, querying and updating the collected data.

The iBOSS cloud platform follows the NGSI-LD specification for managing scientific data through several services via Restful API endpoints. The ways of accessing, handling and using this API for managing data are depicted below.

a. Authentication:

The iBOSS is protected by an OpenID Connect compliant authentication server. Thus, any call made to the iBOSS must include an Authorization header containing a Bearer access token.

An access token can be obtained in two ways:

- If a client has its service account enabled, an access token can be obtained with the following request:

    The <client_id> and <client_secret> are provided by the iBOSS administrator system.

```
 http --form POST https://data-
hub.eglobalmark.com/auth/realms/datahub/protocol/openid-
connect/token client_id=<client_id>
client_secret=<client_secret> grant_type=client_credentials
```

- If a client wants to make API calls on behalf of an end user, an access token can be obtained in exchange of the authorization code contained in the redirect URL

```
export TOKEN=$(http --form POST https://data-
hub.eglobalmark.com/auth/realms/datahub/protocol/openid-
connect/token client_id=<client_id>
client_secret=<client_secret> grant_type=client_credentials |
jq -r .access_token)
```

In the following an example of running a query on the iBOSS cloud platform in order to get a token. The example uses the first method for getting a token (i.e. with a client service account. The POST query is executed through the postman platform (https://www.postman.com/). The figure below shows the main elements of the query and the corresponding response of the iBOSS (red boxes). <client_id> and <client_secret> and the iBOSS responses (<token>and <refresh_token>) were hidden in the figure.
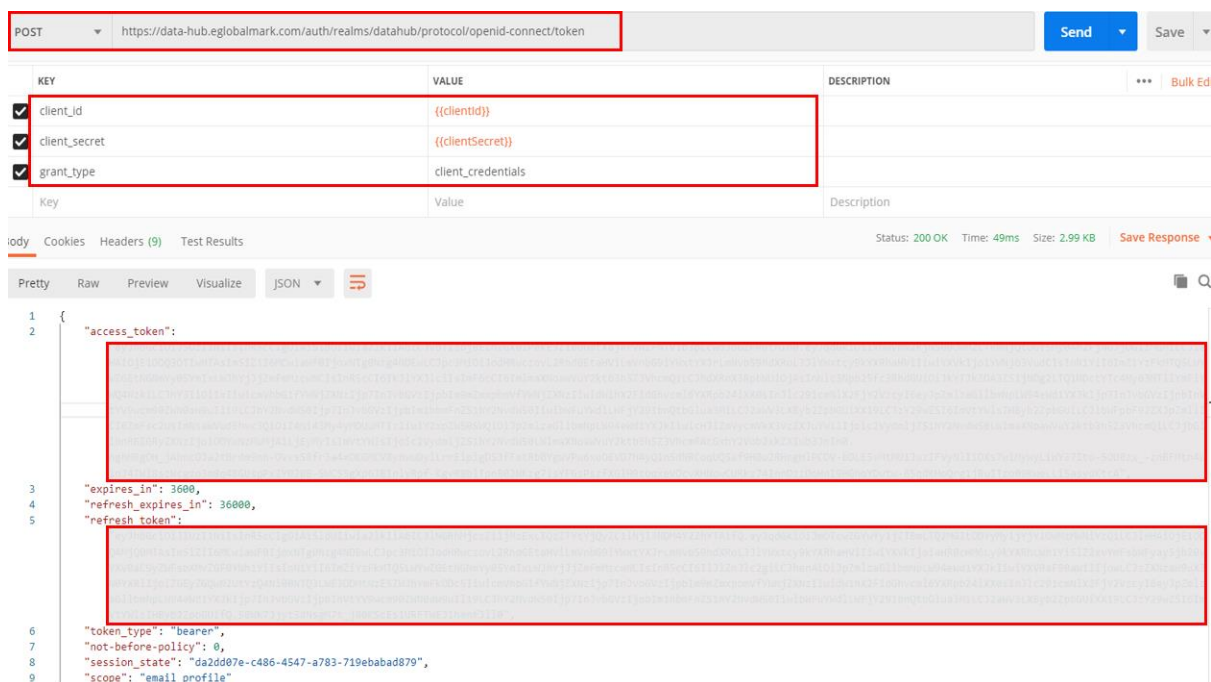
Figure 2: main elements of an Authentication query and corresponding response in the iBOSS

b. iBOSS endpoints

Once the token is generated, it is possible to interact with the iBOSS through the following endpoints and via the address of the iBOSS server: https://data-hub.eglobalmark.com

Object	Method	Path
Create an entity	POST	/ngsi-ld/v1/entities
Search entities	GET	/ngsi-ld/v1/entities
Get an entity by id	GET	/ngsi-ld/v1/entities/{entityId}
Append an attribute to an entity	POST	/ngsi-ld/v1/entities/{entityId}/attrs
Update attributes of an entity	PATCH	/ngsi-ld/v1/entities/{entityId}/attrs
Partial update of an entity attribute	PATCH	/ngsi-ld/v1/entities/{entityId}/attrs/{attrId}
Delete an entity	DELETE	/ngsi-ld/v1/entities/{entityId}
Create a batch of entities	POST	/ngsi-ld/v1/entityOperations/create
Get the temporal evolution of an entity	GET	/ngsi-ld/v1/temporal/entities/{entityId}
Create a subscription	POST	/ngsi-ld/v1/subscriptions
Query subscriptions	GET	/ngsi-ld/v1/subscriptions
Get a subscription by id	GET	/ngsi-ld/v1/subscriptions/{subscriptionId}
Update a subscription	PATCH	/ngsi-ld/v1/subscriptions/{subscriptionId}
Delete a subscription	DELETE	/ngsi-ld/v1/subscriptions/{subscriptionId}

c. NGSI-LD model et @context

The iBOSS cloud API, is fully based on the NGSI-LD specification. An NGSI-LD entity is serialized in JSON-LD format. The structure must comply with some requirements amongst them:

- an entity must have an id (represented by uri:ngsi-ld:<EntityType>:<UUID>)
- an entity must have a type,
- an attribute denotes a property or a relationship,
- an entity may have properties,
- an entity may have relationships with other entities,
- a property may have properties and relationships with other entities,
- a relationship may have properties and relationships with other entities,
- all unitCode(s) have to follow the CEFACT code,
- all properties, relationships and entity types may to have a link (URI) in the @context field (a json-ld specification)

The illustrated use case consists on deploying an Aqua-box device from the partner Bioceanor in a fish containment or sea weed containment in order to monitors various water parameters such as dissolved oxygen, pH and salinity.

The NGSI-LD data model includes 3 main NGSI-LD entities:

- **FishContainment** presents the entity that will contain all sensor measures within its properties,

- **AquaBox** is the device deployed in the Fish containment to monitor it.

- **Sensors** which will provide all measures of the Fish Containment. Sensors are connected to the Aquabox using the NGSI-LD Relationship "connectsTo".

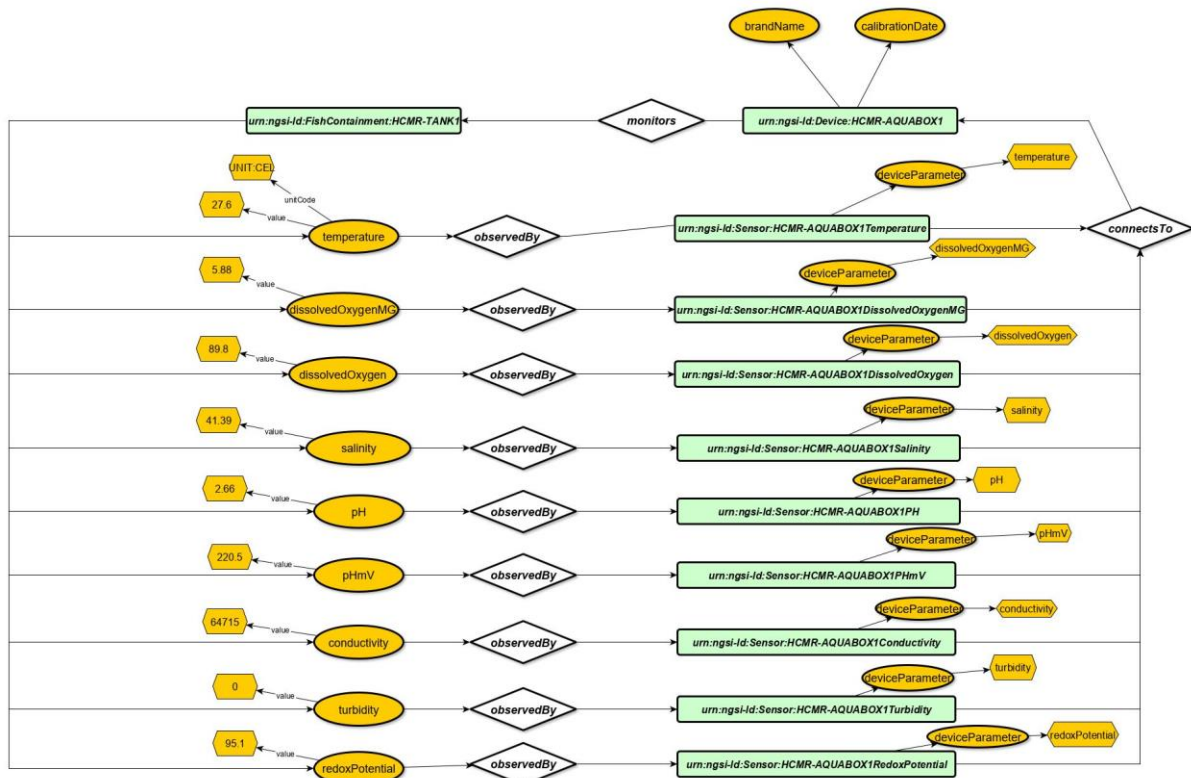The graphical presentation of the NGSI-LD data model is depicted in Figure 3.

Figure 3: simple NGSI-LD model of a fishContainment monitored by an Aquabox

An example of the serialization of the FishContainment entity is presented in Figure 4:

```
{
"id": "urn:ngsi-ld:FishContainment:HCMR-TANK1",
"type": "FishContainment",
"temperature": {
    "type": "Property",
    "value": "27",
    "unitCode":"CEL",
    "observedBy": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:Sensor:AQUABOX1-Temperature"}
},
"@context": [URL to the context]
}
```

Figure 4: example of ngsi-ld serialization of a sensor entity

This data model is a part of the aquaculture model. We are currently working on publishing the aquaculture data model within the Fiware smart data models (https://www.fiware.org/developers/smart-data-models/). This data model is expected to be published in the Aquaculture Repository (https://github.com/smart-data-models/dataModel.Aquaculture)

d. Querying the iBOSS

In this section we execute several queries. In each query example we present its parameters and a screenshot in postman showing the query URL and parameters, and the iBOSS responses.

- **Create Entity:** A query for creating a new entity in the Cloud.

```
Query URI:  https://data-hub.eglobalmark.com/ngsi-
ld/v1/entities/
Query method: POST
Query body: The previous NGSI-LD entity
```
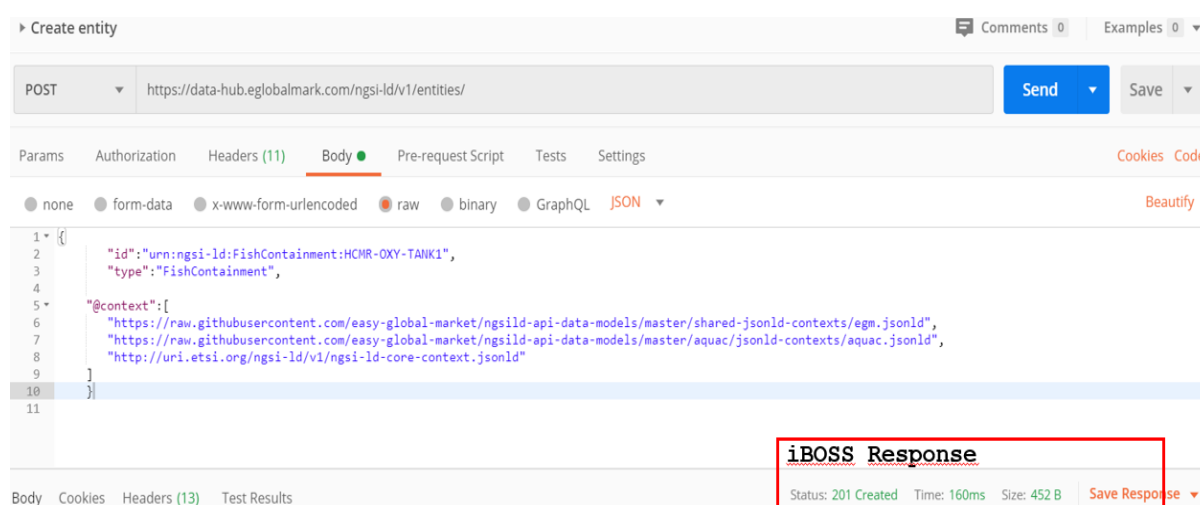

Figure 5: A Create Entity Query executed from the Postman interface and corresponding response in the iBOSS

```
The iBOSS possible (http) Responses for this query are:
- 200 OK, if the entity is created
- 409 Conflict, if the id of the new entity already exists in the
iBOSS,
- 400 Bad Request, if there is a syntactic error in the payload.
```

As a result of the previous Create entity query on the iBOSS a new entity is created in the internal graph database. The Figure below presents an example of the created FishContainment entity in the internal graph database of the iBOSS cloud server.
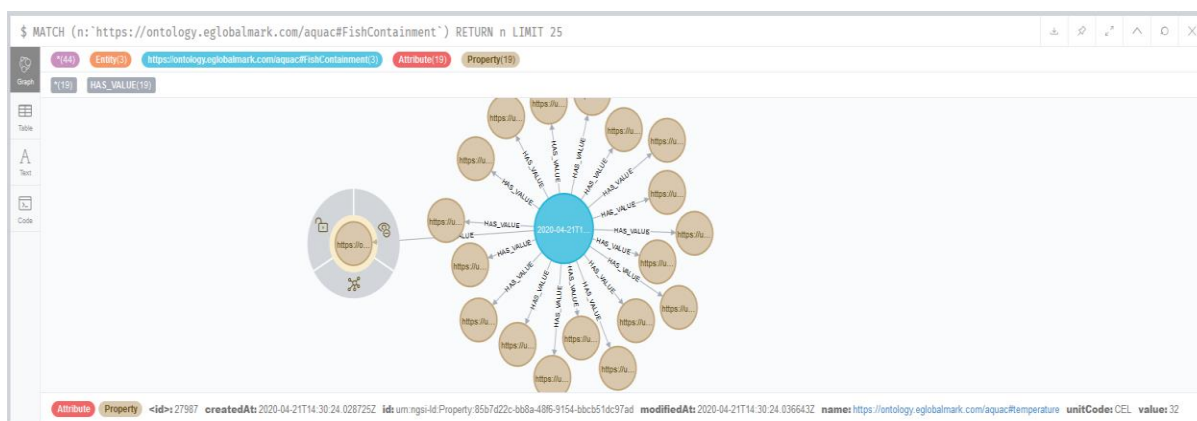

Figure 6 Example of FishContainement Entity in the internal graph database of the iBOSS

- **Create a batch of entities :** A query for creating a set of new entities in the Cloud with one query.

```
Query URI:  https://data-hub.eglobalmark.com/ngsi-
ld/v1//entityOperations/create
Query method: POST
Query body: A Json document containing a set of entities.
```
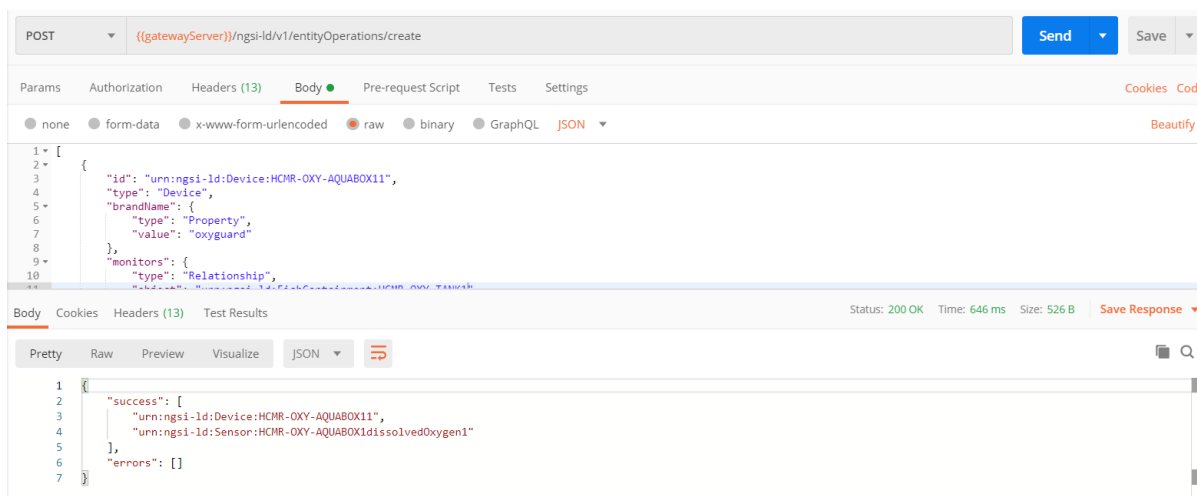

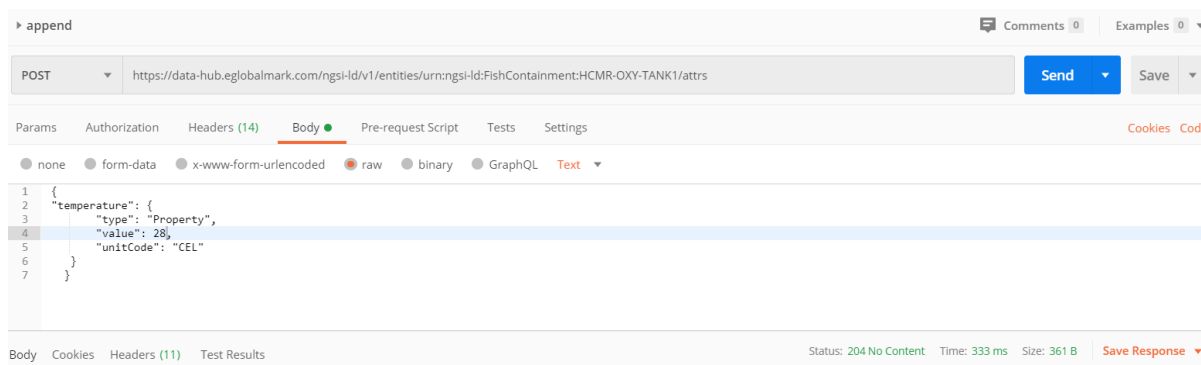Figure 7: Create A Batch of Entities query and corresponding response in the iBOSS

```
The iBOSS possible (http) Responses for this query are:
- 200 OK, with a list of ids of the created entities and list of
errors for the non-created entities.
- 400 Bad Request, if there is a syntactic error in the payload.
```

- **Append an attribute to an Entity:** A query for adding a new attribute to an existing entity

```
Query URI:  https://data-hub.eglobalmark.com/ngsi-
ld/v1/entities/urn:ngsi-ld:FishContainment:HCMR-OXY-TANK1/attrs
Query method: POST
Query body: {"temperature": {"type": "Property","value": 27,
"unitCode": "CEL"}}
```


Figure 8: Append Query and corresponding response in the iBOSS

```
The iBOSS possible (http) Responses for this query are:
- 204 No content, if the attribute is appended,
- 400 Bad Request, if there is a syntactic error in the payload.
Note that appending an existing attribute is possible and will be
considered as an update Query.
```

- **Update entity attribute:** A query for updating the value of an attribute

```
Query URI:   https://data-hub.eglobalmark.com/ngsi-
ld/v1/entities/urn:ngsi-ld:FishContainment:HCMR-OXY-TANK1/attrs
Query method: GET
Query body: {"fishNumber": {"type": "Property","value": 2750}}
```
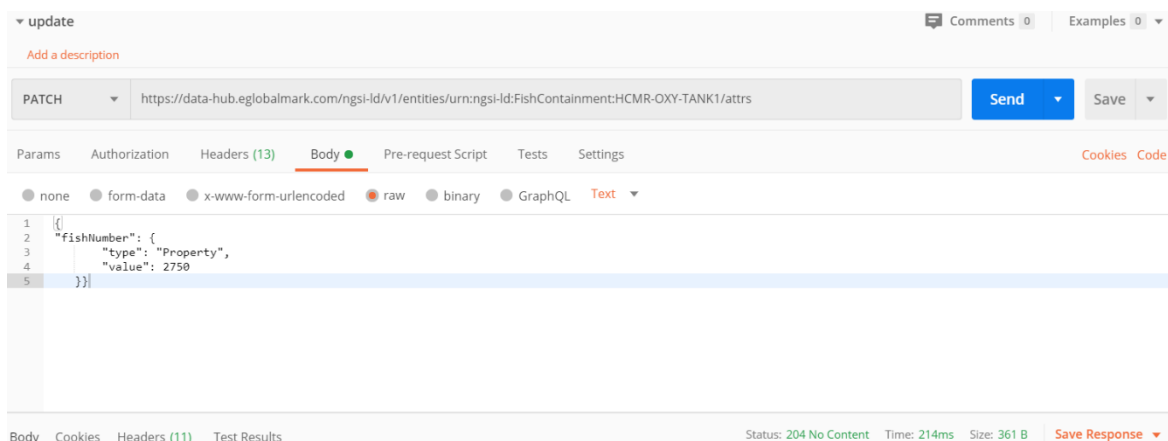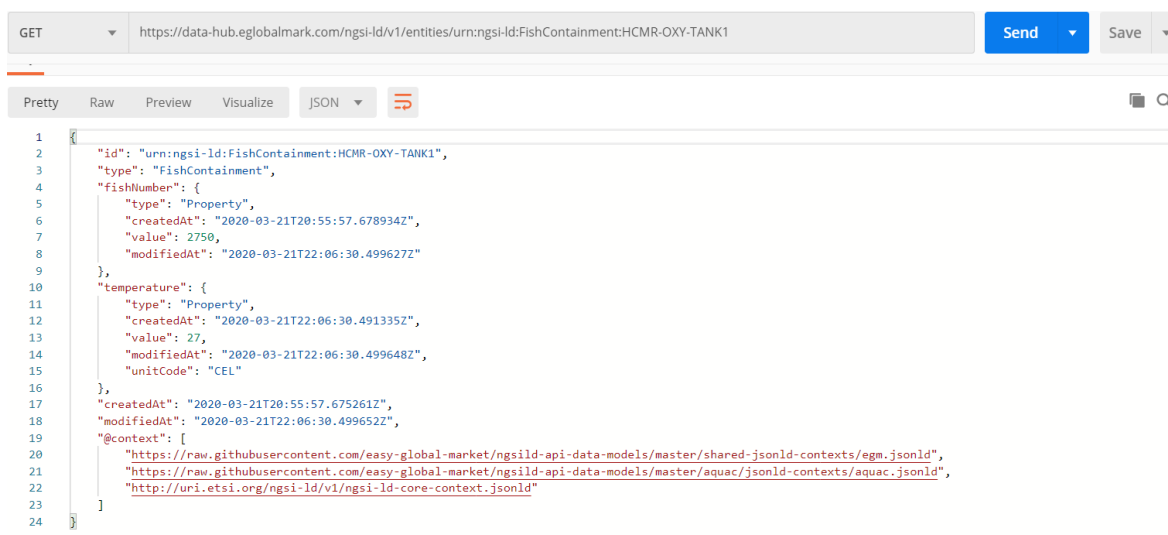
Figure 9: Update Query and corresponding response in the iBOSS

```
The iBOSS possible Responses for this query are:
- 204 No content, if the attribute is updated,
- 207 Multi-Status, with a list of updated on non-updated
attributes.
- 400 Bad Request, if there is a syntactic error in the payload.
Note that appending an existing attribute is possible and will be
considered as an update Query.
```

- **Get an Entity by id:** A query for querying all attributes (properties and relationships) of an entity based on its id.

```
Query URI:   https://data-hub.eglobalmark.com/ngsi-
ld/v1/entities/urn:ngsi-ld:FishContainment:HCMR-OXY-TANK1
Query method: GET
```


Figure 10: get by Id Query and corresponding response in the iBOSS

The createdAt and the modifiedAt properties are automatically generated by the iBOSS and added for all entities, properties and relationships. The createdAt property presents the time of the first creation of an entity, a property or a relationship. The modifiedAt property presents the time of the lats modification that occured on an entity, a property or a relationship.

- **Create Subscription:** A query for creating a subscription on an event in of an entity.

```
Query URI:  https://data-hub.eglobalmark.com/ngsi-ld/v1/subscriptions
Query method: POST
Query body: subscription in Json format (see the example below)
```

The example below, presents the request body of a subscription on notifications when the value of fishNumber property in the entity fishContainement HCM-OXY-1 exceeds 200. The notification will be sent for example to the adress locahost:8084.

```
{"id":"urn:ngsi-ld:Subscription:03",
"type":"Subscription",
"entities": [
{"type": "FishContainment"},
{
"id": "urn:ngsi-ld:FishContainment:HCMR-OXY-TANK1",
"type": "FishContainment"
}],
"q": "fishNumber>200",

"notification": {
    "attributes": ["fishNumber"],
    "format": "normalized",
    "endpoint": {
        "uri": "http://localhost:8084",
        "accept": "application/json"
    }
},
"@context": [
"https://schema.lab.fiware.org/ld/context",
"http://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"]
}
```

The iBOSS cloud server supports several types of subscriptions on events that are happening over attributes of an entity. The subscription conditions may be combined also with time and geop-properties of the entity.

- Get Subscription by id: A query for getting a subscription by id

```
Query URI:  https://data-hub.eglobalmark.com/ngsi-
ld/v1/entityOperations/create
Query method: POST
Query Body: List of Entities
```
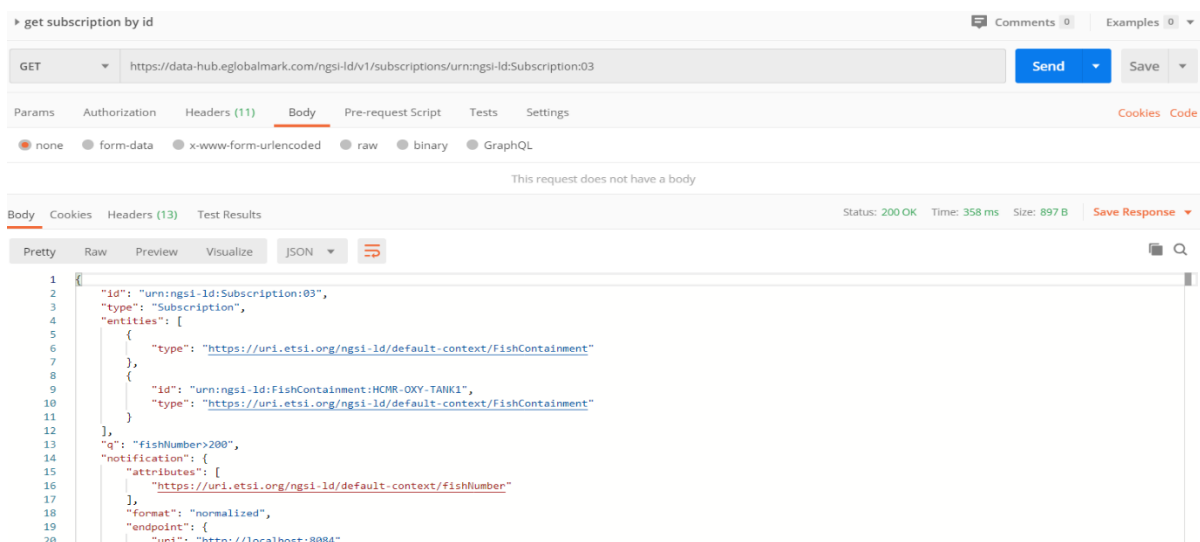
Figure 11: Get Subscription by Id Query and corresponding response in the iBOSS

- Delete Entity: A query for deleting an Entity by id

Query URI: https://data-hub.eglobalmark.com/ngsi-ld/v1/subscriptions/urn:ngsi-ld:Subscription:03
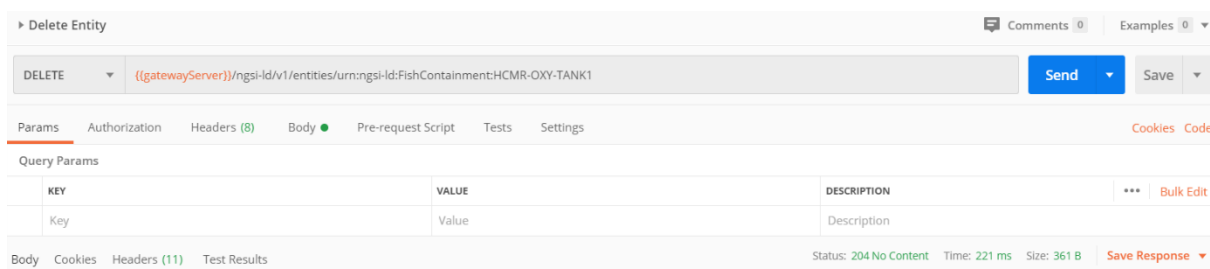Query method: DELETE

Figure 12: Delete Entity Query and corresponding response in the iBOSS

4 Applications interactions with the iBOSS broker

4.1 Data visualisation of collected data (dashboard)

When attributes are updates, the oldest values are transmitted to an internal time series database in the iBOSS cloud server. In this section we present examples of already pushed data from Bioceanor and OxyGuard partners to the iBOSS cloud server. Using Graphana (https://grafana.com/) as visualisation tool, it is possible to query and vizualize all attribute data of an entity in a specifique periode of time.

The figure below presents all pH value of the *HCMR-TANKN1 FishContainment* entity during the last 7 days.

Figure 13: pH values in the HCMR-TANKN1 FishContainment

Within pushed time series data in the iBOSS cloud server, it is possible to create real-time charts during a specific period of time.

Below a second example showing the current values of temperature, salinity and pH in the interval of minimal and maximal values during last three days and charts showing the evolution of the dissolved oxygen and dissolved solids of the *HCMR-TANKN1* FishContainment entity.
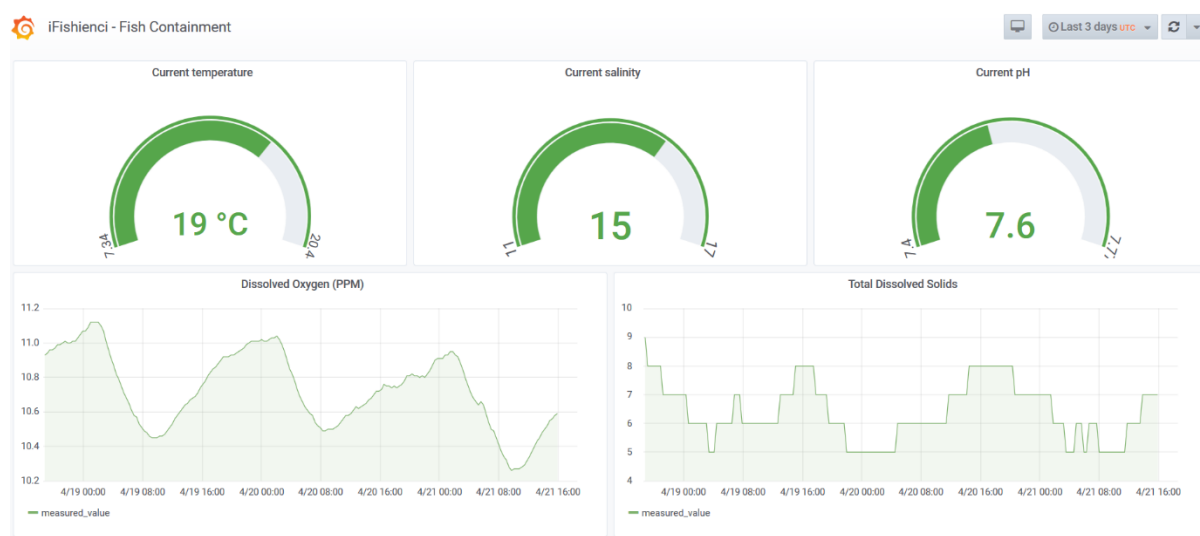

Figure 14: Current values and Charts of temperature, salinity, pH dissolved Oxygen and Dissolved Solids values in the HCMR-TANKN1 FishContainment

4.2   Digital twin

Digital twins (DT) are virtual representations of physical entities, providing augmented views of the physical entities. The NGSI-LD based iBOSS broker Is a central component in digital twins deployment as it allows to update the DT state to the one of the physical entity by subscribing to its change through NGSI-LD subscription mechanisms whereas the DT can publish its information back to the broker. The core of a DT is a simulator of the physical entity. Examples of interaction scenarios with the digital twin include (Figure 15):

- **Anomaly detection**: raising alerts when the physical system does not evolve as expected

- **Predictive evolution**: simulation the evolution of the physical entity from its current state

- **What-is scenarios**: evaluate impact of changing one or several parameters of the physical system.
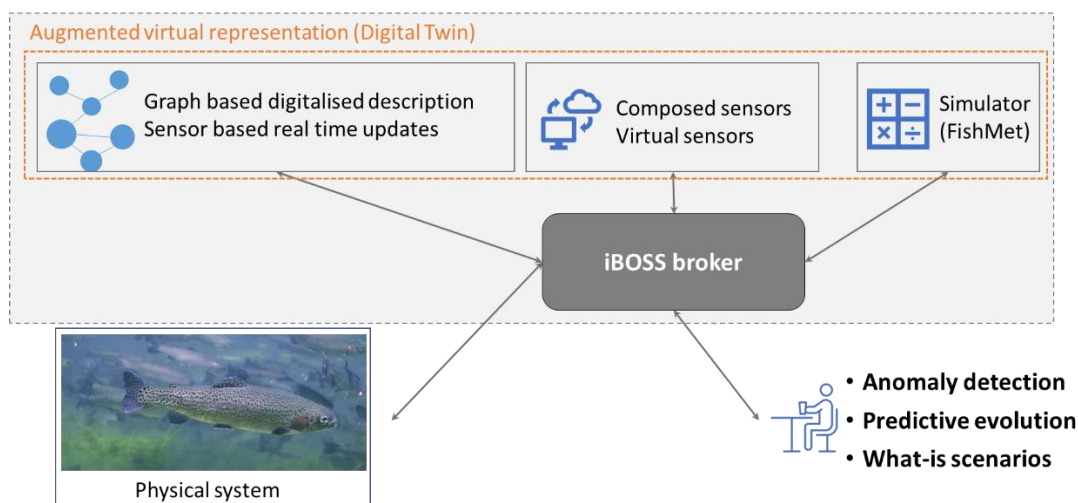

Figure 15: Digital twins integration with the iBOSS NGSI-LD broker.

Following this paradigm, the fish digital twin (FishMet) is integrated within the iBOSS as follow:

- **Data are produced and consumed by FishMet through the iBOSS broker:** FishMet accepts data on the environment conditions, the feeding protocol, various parameters of the fish morphology and physiology. The output of the model is then the pattern of the fish feeding behaviour (e.g. ingesting food items), its internal state (e.g. the level of hunger motivation) and parameters of the gastrointestinal system functioning (e.g. stomach and gut fullness, absorption, evacuation of faeces). Due to the stochastic simulation nature of the model can, any kind of realisatic stochastic input can be used, notably individual and group differences between the fish.

- **Data that will be exchanged with the iBOSS cloud during development phase (to train the FishMet):** FishMet can input data on characteristics of the food (i.e. palatability), parameters of the feeding protocol, basic biological characteristics of the fish (including group and individual variability). Then it will output prediction data on fish behaviour, its individual internal state (e.g. hunger and potentially stress and welfare) and the ongoing physiological state and growth.

- **Data that will be exchanged with the iBOSS cloud during operational phase (to provide recommendations and decision support):** iBOSS system can send the data on the environment (e.g. physical conditions, feeding protocol etc.) and in return obtain model predictions on the feeding behaviour of the fish, Feed conversion ratio, waste (uneaten food), its motivational and physiological state. Any stochastic data can be used, making it possible to test diverse scenarios, even those that are difficult or impossible to implement physically.

- **Who and when to trigger the execution of FishMet simulation:** The FishMet simulations can be conducted directly by the end users, both manually using the graphical interface and through programming complex scenarios in model scripts. Additionally, simulations can be automatically administered by the iBOSS system. The latter case includes provision of iBOSS machine-generated modelling scenarios for fully automatic functioning.

FishMet can be running on-demand. The multiple types of user interface implemented in the FishMet model allow diverse scenarios for running it on demand. These range from the program own graphical interface, allowing simple application by a general non-IT user and running complex simulation batches by programming the model through its command line interface..

5 Preliminary analysis

In this paragraph preliminary analyses of an existing dataset provided by AquaBioTech are shown, as example of the data quality check and correlation analysis performed.

These data come from 2 different periods:

- From 14/11/18 to 09/01/19
- From 19/01/17 to 17/03/17

The frequency of measured data is also different:

- In Sump, pH, NO2 (nitrate), NO3 (nitrite), TAN (total ammonia nitrogen) are collected daily.
- In Tanks, dissolved oxygen and water temperature are collected every 10 minutes.

The purpose of this analysis was to highlight a possible correlation between water parameters and feed. However, information on feed were insufficient, so we focused our study on water parameters.

Step 1: data cleaning

In every data analysis, we need to clean the data before proceeding to any statistical testing. The abnormal data based on biological statement (for example pH must be between 0 and 14) were eliminated and the outliers (data suddenly out of range, for just one occurrence) based on statistical treatment. For that we applied Grubbs' test.

We AGGREGATED the data hourly, to get the same timestamp to each parameter.

Figure 16: several steps of data cleaning : 1 – before data cleaning 2 – after eliminating outliers 3- after regrouping data by hour.

Step 2: correlation study between factors in the same tanks

We used the Pearson Method to examine linear correlations between parameters. Some parameters were observed to be correlated, but results were very different between the tanks.

For example, in tank 5 we found a positive correlation between temperature and pH (see Figure 17) and in tank 8, we observed a negative correlation between those same parameters (see Figure 18).

In other words, if temperature raise, in tank 5, pH will also raise, whereas in tank 8, pH will decrease. With no other metadata or explanation on the feed, it is difficult to interpret those difference. It could be for example a difference in feed, in fish development stage or in biomass.
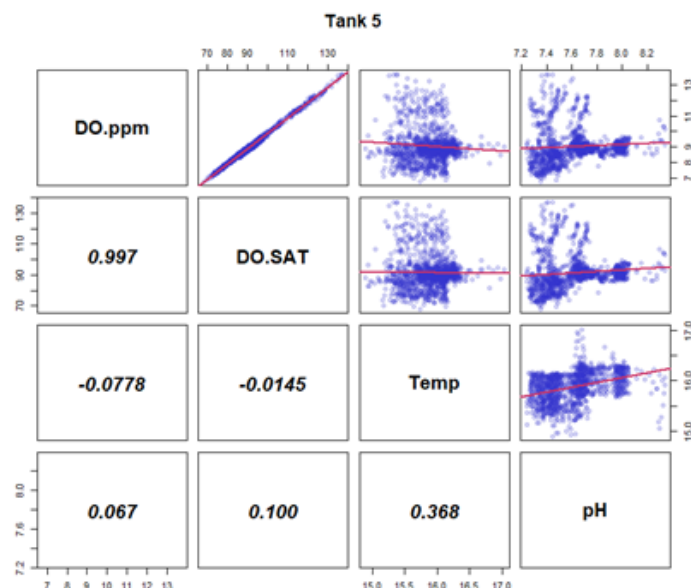

Figure 17: Correlation matrix based on Pearson method between dissolved oxygen concentration (DO.ppm), dissolved oxygen saturation (DO.SAT), water temperature (Temp) and pH for Tank 5. The closer the value is to 0, the less the parameters are correlated. The closer the value are to 1 or -1, the more the values are correlated. The noticeable result here is temperature positively correlated to pH with a value of 0.368.
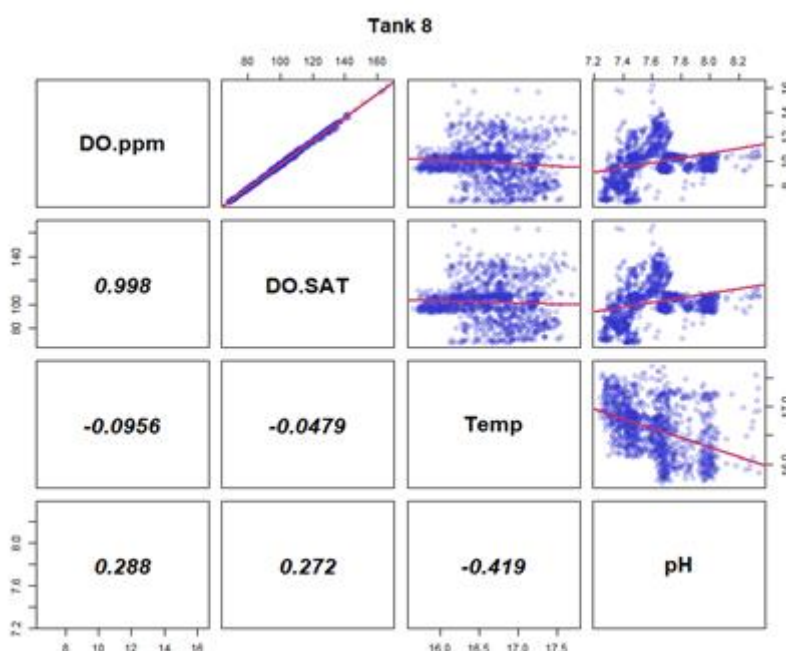

Figure 18: Correlation matrix based on Pearson method between dissolved oxygen concentration (DO.ppm), dissolved oxygen saturation (DO.SAT), water temperature (Temp) and pH for Tank 8. The closer the value is to 0, the less the

parameters are correlated. The closer the value are to 1 or -1, the more the values are correlated. The noticeable result here is temperature negatively correlated to pH with a value of -0.419.

Step 3 – Correlation study between factors in all tanks

As the previous step highlighted different correlations for the different tanks, in this step we investigate correlation between pH and temperature or pH and dissolved oxygen in all tanks. We wanted to see if some groups of tanks could appear and thus giving us an indication to explain the observed correlation.

For that we displayed a heat map of the different linear correlation that we could observe in step2.

The results show that the correlation of pH with DO is not consistent within all tanks (Figure 19). No particular groups can be clearly distinguished. However, the correlation of pH with temperature within the tanks can be organized in 2 groups (Figure 20). One group have stronger correlation than the other. However, with no other metadata, we cannot correlate this result to any particular event.

It will be mandatory for the next analyses, to gather as much metadata as we can to be able to explain why we can have different correlations between parameters in different tanks.
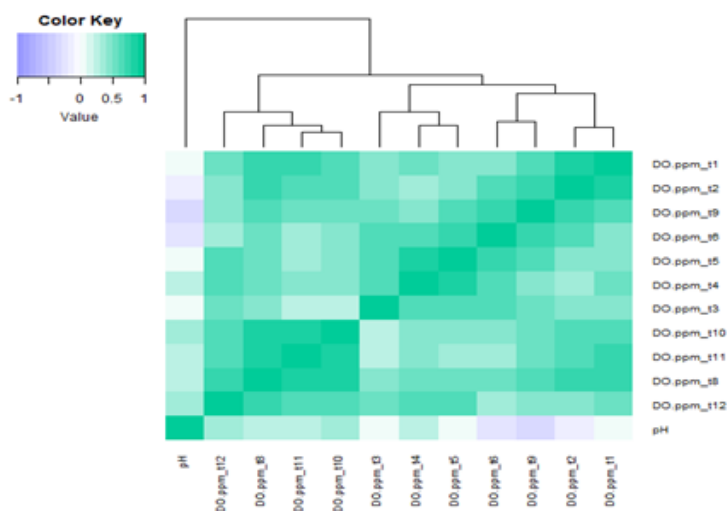

Figure 19: Heat map of linear correlation between pH and DO in all tanks. No big groups can be determined; however some small groups appear.
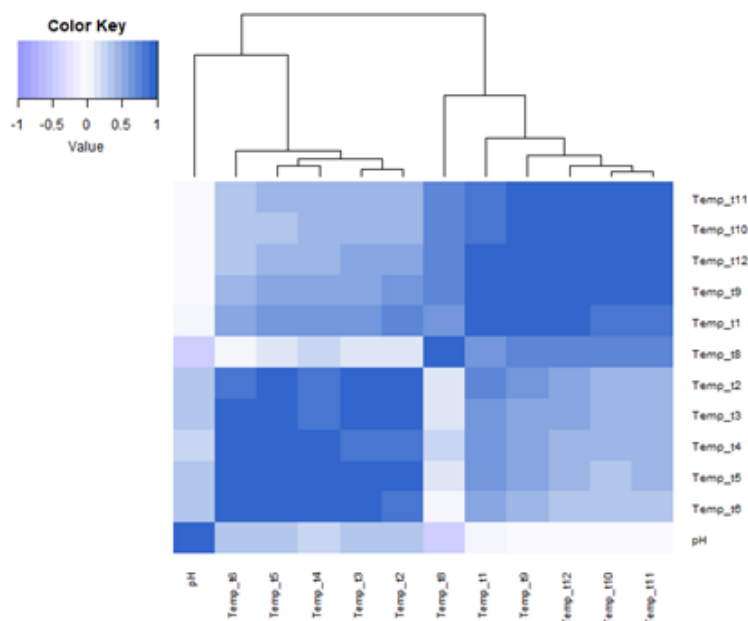
Figure 20: Heat map of linear correlation between pH and temperature in all tanks. 2 big groups can be distinguished.

Step 4 : Comparison of a single factor between tanks

After comparing parameters between them, we wanted to compare the distribution of only one parameter between tanks.

First, we used a Kruskal-Wallis test on DO.ppm parameter and temperature. It is a non-parametric method for testing whether samples originate from the same distribution. It returns a p-value that must be below 0.05 to prove that there is significative difference between samples.

Here are the results, where p-value is always below 0.05 :

```
        Kruskal-Wallis rank sum test

data:  DO.ppm by Group
Kruskal-Wallis chi-squared = 1744.5, df = 10, p-value < 2.2e-16
```

```
        Kruskal-Wallis rank sum test

data:  Temp by Group
Kruskal-Wallis chi-squared = 3493.4, df = 10, p-value < 2.2e-16
```

Then we plotted the results on a graph to see if some groups can appear again.

For DO dispersion (Figure 21), we can see 2 groups :

 - Tanks 1 to 6

- Tanks 8 to 14

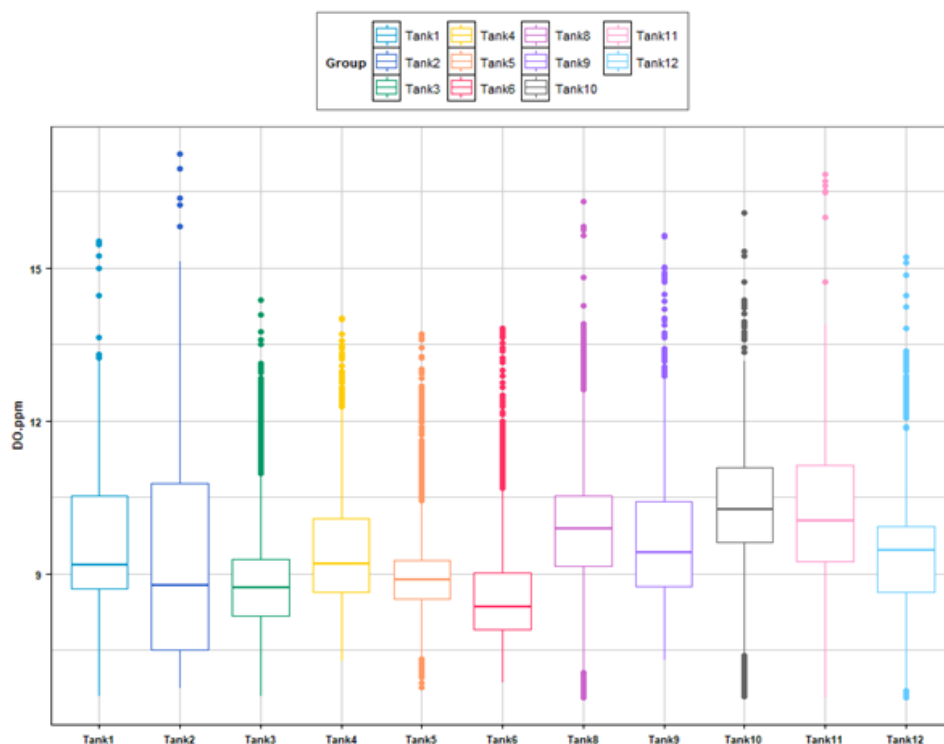Nevertheless, the difference is not very significative between the 2 groups.

Figure 21: Dispersion of DO distribution in each tank.

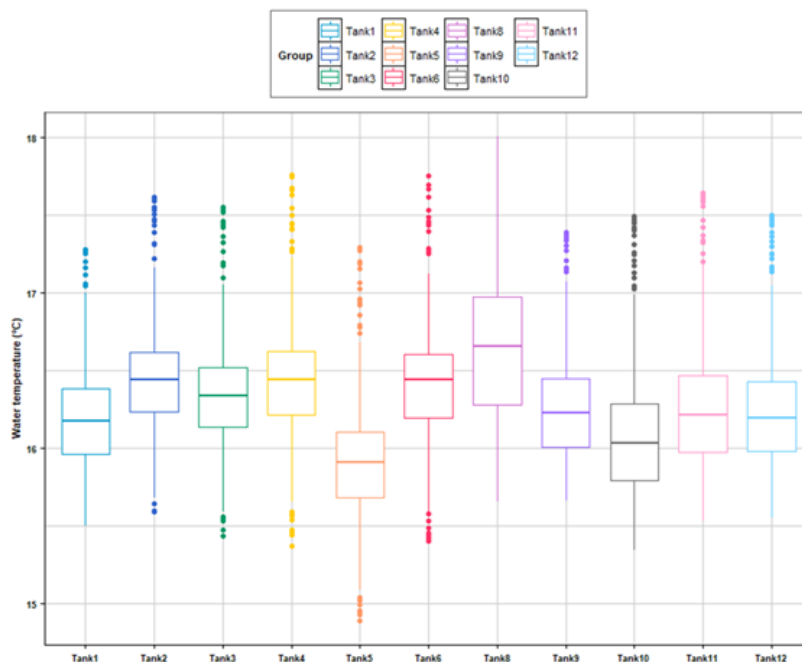For temperature distribution, no clear group appears (Figure 22).

Figure 22: Dispersion of temperature distribution in all tanks.

Step 5 : Correlation with Biomass

We also got some information about biomass in all tanks. It was only one data every 2 weeks approximatively. Thus, it was difficult to correlate it to physico-chemical parameters.

Nevertheless, we plotted the result to see if the biomass evolution could be different between the tanks.

With the regular plotting, we did not see much difference (Figure 23), however, the log 2-fold plotting highlighted some difference (Figure 24). As we wanted to compare it to the feed in every tank, the information was not available, as we could retrieve only total feed in all tanks. It could have been very interesting to know exactly what the feed uptake in every tank is, that could explain such growth difference.
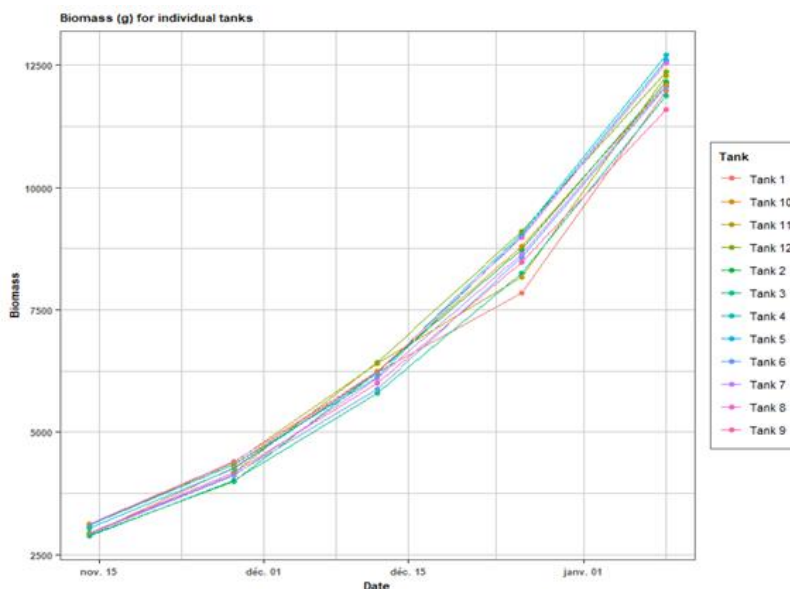

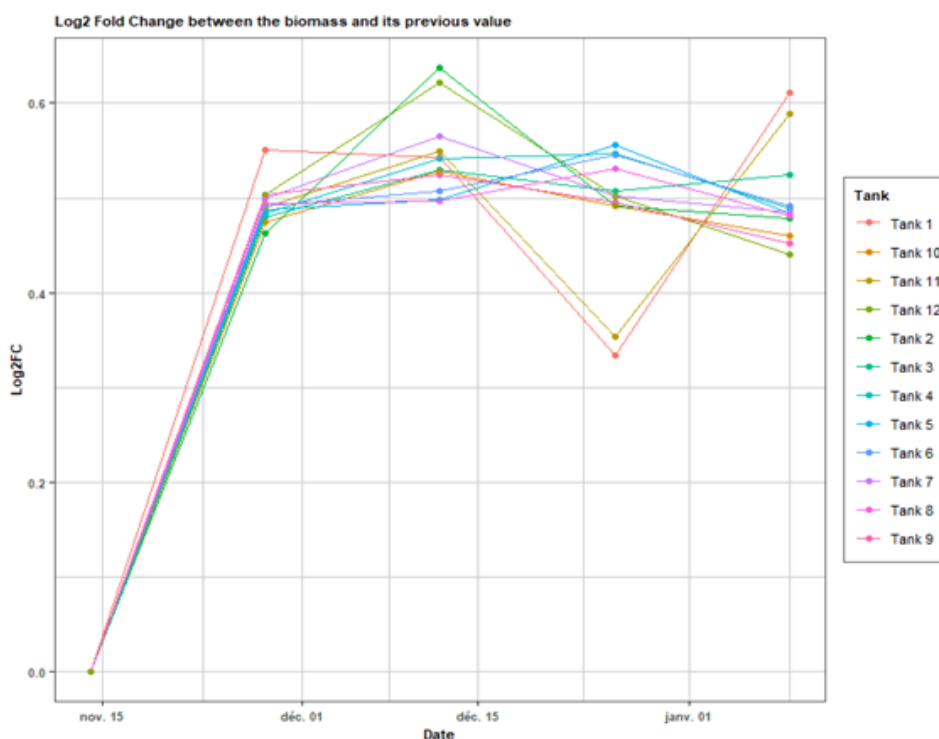*Figure 23: Biomass evolution with time (fish growth)*

Figure 24: Biomass evolution with time (fish growth) with log 2- fold scale

In conclusion, some statistical analysis was made on the data we could obtain from AquaBioTech, that highlight some information in the dataset. However, to continue with AI analysis, we were missing some valuable metadata information, such as precise feed per tank, human manipulation, inputs, etc.

We will consider this lack of information in the future iFishIENCi AI strategy, to be sure that it contains all needed metadata.

6 Conclusions

In this document the iFishIENCi data management procedures has been described. The procedures are in line with the guidelines outlined in the Data Management Plan (DMP).

A summary of the type of data collected is provided, describing the main storage format. In addition, a summary of all the datasets collected by M18 is provided, while the detailed overview of all the datasets is preserved in the iFishIENCi SharePoint and available to the project partners. The full list of datasets is constantly updated and represents a living document for the whole duration of the iFishIENCi project.

Accordingly, with the main objectives of the iFishIENCi project, the development of the iBOSS system is cardinal to improve production control and management of all fish aquaculture systems. The data necessary for the implementation of the iBOSS system, both from sensors, manually collected and generated by model's simulations, are also stored in the iBOSS Cloud Platform, following NGSI-LD specifications.

Finally, an example of data quality check and preliminary parameters correlations is shown; the data used in these analyses belong to a dataset collected before the start of the iFishIENCi project but contain the same type of data.