HORIZONS 2020 PROGRAMME

Research and Innovation Action – FIRE Initiative

Call Identifier:	H2020–ICT–2014–1
Project Number:	643943
Project Acronym:	FIESTA-IoT
Project Title:	Federated Interoperable Semantic IoT/cloud Testbeds and Applications

D5.2 - Experiments Implementation, Integration and Evaluation

Document Id:	FIESTAIoT-WP5-D52-20170612-V28
File Name:	FIESTAIoT-WP5-D52-20170612-V28.docx
Document reference:	Deliverable 5.2
Version:	V01
Editor:	Flavio Cirillo
Organisation:	NEC
Date:	12 / 06 / 2017
Document type:	Deliverable
Dissemination level:	PU

DOCUMENT HISTORY

Rev.	Author(s)	Organisation(s)	Date	Comments
V01	Flavio Cirillo	NEC	2016/11/30	ToC definition
V02	Flavio Cirillo	NEC	2016/12/09	Final ToC definitions after comments from UC and INRIA.
V03	Rachit Agarwal	Inria	2016/12/22	First verison of contribution
V04	Flavio Cirillo	NEC	2016/12/22	NEC first draft contribution
V05	David Gómez/ Jorge Lanza/ Luis Sánchez	UNICAN	2016/12/22	First round of contributions of Dynamic Discovery Experiment
V06	Rachit Agarwal	Inria	2017/01/18	Second round of contributions (Section 2.3, Section 3.3, Appendix)
V07	Flavio Cirillo	NEC	2017/01/24	Merged contributions
V08	Denis Rousset/ Konstantinos Bountouris	Com4Innov	2017/01/25	Added Appendix on Cloud support to Experimenter
V09	Rachit Agarwal	Inria	2017/01/28	Address comments from Editor
V10	David Gómez/ Jorge Lanza/ Luis Sánchez	UNICAN	2017/02/09	Finalized 2.2 and 3.2 sections
V11	Mengxuan Zhao	EGM	2017/02/10	Evaluation section
V12	Flavio Cirillo	NEC	2017/02/15	Added Executive Summary, Conclusions. Addressed comments for each section
V13	Aqeel Kazmi	NUIG	2017/02/21	Quality Review
V14	Ronald Steinke	FOKUS	2017/02/21	Technical Review
V15	Tiago Teixiera	UNPARALLEL	2017/02/21	Technical Review
V16	David Gómez Rachit Agarwal Mengxuan Zhao Flavio Cirillo	UNICAN INRIA EGM NEC	2017/02/22	Reviewers' comments addressed
V17	Flavio Cirillo	NEC	2017/02/24	Ready for Submission
V18	Mengxuan Zhao	EGM	2017/05/04	Evaluation and validation sections re-structure
V19	Mengxuan Zhao, Flavio Cirillo, Luis Sanchez, Rachit Agarwal	EGM NEC UNICAN INRIA	2017/05/11	Final version of the structure of Section 3 and 4
V20	Mengxuan Zhao, Flavio Cirillo, Luis Sanchez, Rachit Agarwal	EGM NEC UNICAN INRIA	2017/05/17	First round of contribution to Section 3 and 4
V21	Mengxuan Zhao, Flavio Cirillo,	EGM NEC	2017/05/26	Completed all the contribution to Section 3 and 4 by all partners

	Luis Sanchez, David Gómez Rachit Agarwal	UNICAN INRIA		
V22	Mengxuan Zhao	EGM	2017/05/29	Completed section 3 and 4
V23	Flavio Cirillo	NEC	2017/05/30	Compiled the full document. Updated Introdcution and Conclusions. Added captions to tables. Fixed reference and citations.
V24	Elias Tragos	NUIG	2017/06/01	Quality Review
V25	Tiago Teixeira	UNPARALLEL	2017/06/05	Technical Review
V26	Ronal Steinke	FOKUS	2017/06/12	Technical Review
V27	Mengxuan Zhao, Flavio Cirillo, Luis Sanchez, David Gómez Rachit Agarwal	EGM NEC UNICAN INRIA	2017/06/12	Comments addressed
V28	Flavio Cirillo	NEC	2017/06/12	Document finalized. Ready for Re-Submission

OVERVIEW OF UPDATES/ENHANCMENTS

Section	Description
Section 1	Updates introduction text explaining the new contents of Section 3 and Section 4
Seciton 1	Explanation that the “validation of FIESTA-IoT platform from the point of view of testbeds” will be left to deliverable D6.3
Section 3	Completely restructured section and described the difference between methodologies for validating FIESTA-IoT and evaluation of the experiments at different staegs of their life-cycle.
Section 3.1	Added methodologies for evaluating experiments: evaluating achievements, evaluating the integration and implementation process and evaluation as advancement in the state of the art.
Section 3.2	Added methodologies for validating FIESTA-IoT platform through experiments: validatation of concepts, valdiationd of tools and validation of resources.
Section 4	New section comprehensive of the meticolous application of both evaluation of experiments and validation of FIESTA-IoT
Section 5	Update conclusion in order to includes the new outcomes of Section 3 and Section 4.
Annex III	Moved questionnaire from Section 4
Annex IV	Create the checklist for evaluating the integration and implementation of experiments. The checklist has been answered by all the three in-house experiments

Annex V	Create the questionnaire for validating the usability and the resources of FIESTA-IoT platform. The questionnaire is containing questions regarding the usability of tools and the time spent to integrate experiments. All the three in-house experiments have answered to the proposed questions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMS AND ACRONYMS

API	Application Program Interface
CM	Context Management
DB	Database
DoW	Description of Work
EEE	Experiment Execution Engine
EMC	Experiment Management Control
ERM	Experiment Registry Module
FED-Spec	FIESTA-IoT Experiment Description
FEMO	FIESTA-IoT Experiment Model Object
FIRE	Future Internet Research and Experimentation
FISMO	FIESTA-IoT Service Model Object
GE	Generic Enabler
GEri	Generic Enabler reference implementation
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ID	Identifier
IoT	Internet of Things
KPI	Key Performance Indicator
NGSI	Next Generation Service Interface
QoE	Quality of Experience
SCM	Smart City Magnifier
SMG	Semantic Mediation Gateway
SPARQL	SPARQL Protocol and RDF Query Language
UI	User Interface
VE	Virtual Entity
WP	Work Package

1 EXECUTIVE SUMMARY

This deliverable is reporting the contribution of the whole experimentation work package (WP5) by describing the experiments implementation and integration (task T5.2), and validation and evaluation of experiments (task T5.4). The integration of experiments of third-parties has been neglected in this document, since at time of reporting, no third-parties (from Open Calls) have been acknowledged to join the project.

This deliverable addresses future FIESTA-IoT users, like researcher on Future Internet Research and Experimentation (FIRE), members of other Internet of Things (IoT) communities and projects and entrepreneurs and application developers. It provides some examples on how to use and leverage the FIESTA-IoT platform, tools and concepts. Furthermore, it also addresses the researchers and engineers within the FIESTA-IoT consortium to acquire feedback on which aspects of the platform should be improved, what is already possible to achieve with the platform, and what are the expectation for third year of the project from experimenters' point of view.

The three in-house experiments delineates three different approaches on how to leverage the FIESTA-IoT platform for IoT experiments having different scenarios. For example, the adoption and implementation of the concept of Virtual Entities in the case of the Data Assembly and Services Portability Experiment, the massive usage of the IoT Service Discovery for the Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access, and the usage of the FIESTA-IoT execution engine by the Large Scale Crowdsensing Experiments (Appendix II provides experiment specification in FED-Spec form). Other components of the FIESTA-IoT platform, such as, the security functions, the meta-cloud storage and the communication functions, are transversely used by all the experiments.

The first two sections of this deliverable provide a status report of the three in-house experiments. Because of the demonstrator typology of this deliverable, this document is to be considered complementary for the screencast video linked in the main experimenters webpage of the FIESTA-IoT project[1]. Section 2 depicts the actual achievements of each experiments and the comparison with original plan and design phase carried out during the task T5.1 (that has ended with the report contained in (FIESTA-IoT D5.1, 2016)). Section 3 describes the requirements requested by the experimenters that have been already satisfied by the FIESTA-IoT platform and the achieve KPIs, specified in (FIESTA-IoT D5.1, 2016).

Section 3 describes in very detail the methodology of the evaluation that is used during the FIESTA-IoT project to bring improvements on both sides, the FIESTA-IoT platform and the experiments. The tools (such as questionnaires and checklists) used for performing the validation and the evaluation are reported in Annex III Questionnaire of experiment evaluation from FIESTA-IoT point of view, Annex IV EvaluatION experiment integration and implementation Checklist and Annex V QuestionNaire: Validation of the FIESTA-IoT resources.

[1] http://fiesta-iot.eu/fiesta-experiments/

The validation of FIESTA-IoT platform from the point of view of testbeds will be left to deliverable D.6.3, which is about certification, but it will be open to host the feedbacks of testbeds owners to the FIESTA-IoT platform. This document is focused only on experimentations and in particular it is a demonstrative document about experimentation in FIESTA-IoT.

The methodologies are then applied as exercise to all the three in-house experiments, which are the only one, at the time of creation of this deliverable, that have been designed, and partially integrated and implemented. The outcome of this process is depicted in Section 4.

In Annex I the technical support for cloud resources, that are hosting the FIESTA-IoT platform used by each of the three in-house experiments, is reported.

Finally, in Annex II the example of experiment specification, in the form of FEDSPEC, for the FIESTA-IoT experiment engine used by one of the in-house experiments (viz. Large Scale Crowdsourcing experiments) is reported.

2 EXPERIMENTS SELECTION, IMPLEMENTATION AND INTEGRATION

This section describes the three in-house experiments from the point of view of their actual implementations and achievements. In particular, it describes the use-cases implemented amongst the ones designed in Task 5.1 (see (FIESTA-IoT D5.1, 2016)). For each of the 3 experiments there is a detailed description of the architecture and the workflow realized.

In addition, each experiment describes its interaction with the FIESTA-IoT platform by listing the dataset used and formalized with the FIESTA-IoT ontology, and the usage of the tools of FIESTA-IoT platform (together with external tools).

Finally, at the end of each sub-section there is a report on the actual outcomes of the experiments and the future work plan for the third year of the FIESTA-IoT project in order to improve the actual implementation and meet the foreseen outcomes described in (FIESTA-IoT D5.1, 2016).

2.1 Data Assembly and Services Portability Experiment

The Data Assembly and Service Portability experiment has been shaped as a smart city application, named Smart City Magnifier, which is capable of showing situations, trends and forecasts of a city at different levels of detail. The level of detail is a parameter with 3 degrees of freedom (i.e. axes):

1. the space which specifies the geographic scope;
2. the time, which has effect on the time window of time series evaluation, on the historical visualization and on the forecasting horizon;
3. abstraction, which defines the details of the situation to be shown to the user.

The last axis can be seen again as a multi-dimension parameter. For instance, the abstraction axis can vary over the abstraction of the situation, e.g. lower abstracted situation might be the temperature situation, air pollutant concentrations or also occupied parking slot percentage in the focused area, whilst an higher abstracted situation might be traffic situation which may take into consideration all the previous situations. The abstraction axis can also vary on the abstraction level of the subject of the analysis, e.g. from low to high abstracted subject a situation might refer to a building, a street, a suburb, a city, a region or a country (see Figure 9).
The first implementation of this experiment has taken into consideration only the space and the abstraction axes, whilst the time axes will be a future development.

2.1.1 **Use-case selection**

For our experiment, we have envisaged several use-cases (see **(FIESTA-IoT D5.1, 2016)**): Resource-Oriented analytics, Observation-Oriented analytics, Knowledge-Produced analytics and Hybrid analytics.

During the first implementation, we have focused our attention on implementing a first complete application on a single use-case: the Observation-Oriented Analytics. Others will be extensions of the core application described below.

Observation-Oriented analytics.

An observation oriented analytics is interested on actual measurements made by resources combining historical data and new observations coming from the underlying testbeds.

For this kind of analytics we have implemented two different use-cases, in order to make use of data coming from testbeds that are exposing themselves as IoT services and testbeds that are pushing their observations into the Semantic Data Repository.

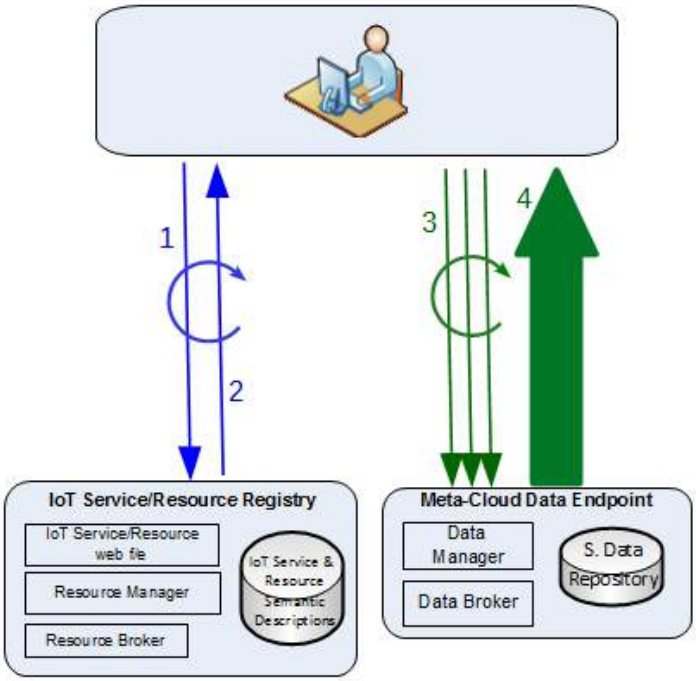

Figure 1 Observations request via IoT Services

In case of endpoints available the use-case implemented is the one shown in Figure 1. The Data Assembly and Service Portability Experiment first performs a SPARQL query (step 1) to the IoT Service Resource Registry in order to infer the available endpoints (returned at step 2). Then it is querying in polling each of the endpoints (step 3) in order to get the latest available data (step 4). Step 1 and 2 are continuously repeated with a predefined frequency in order to catch any changes in the endpoint set.

This use-case differs from the one depicted in **(FIESTA-IoT D5.1, 2016)** in two ways: the metadata cloud is not queried in order to get historical data and the data requests are made in polling. The first difference is due to the fact that at the state of the experiment implementation, the testbeds exposing endpoints were not exposing a semantic historical repository. The second difference is due to the fact the Subscription Manager functionalities were not yet ready at the state of the first experiment implementation.

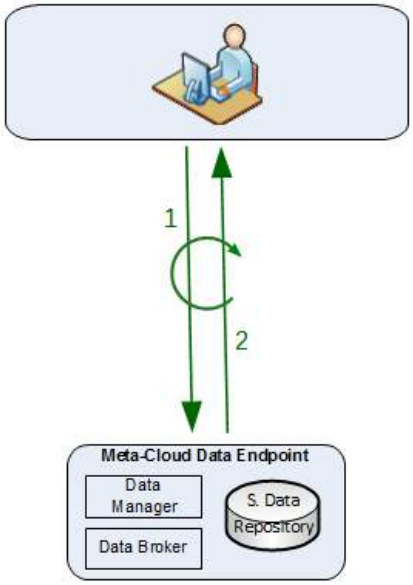

Figure 2 Observations request via Semantic Data Repository

In case of data available only in the Semantic Data Repository, the use-case implemented is shown in Figure 2.

The experiment is making a SPARQL query (step 1, see section 2.1.2.2 for SPARQL examples) to the Semantic Data Repository requesting historical in polling, with a specified period τ , with the following approach:

1) data with timestamp not older than the time of the request minus τ.

2) the response (step 2) will contain a dataset of historical data. Each observation is timestamped with a date ranged between the specified start of the time-window and the time of the request. The resulting dataset is then used for the observation analytics.

Step 1) and 2) are repeated with the period of τ; the time-window is shifted of τ plus 1s in order to start at the time of the last submitted query.

Also this use-case differs from the one depicted in the **(FIESTA-IoT D5.1, 2016)** since the data is continuously retrieved with a time-windowed historical data instead of a subscription (or polling) to IoT endpoints after the first historical query. This is due to the fact that at the time of the experiment implementation the integrated testbeds able to push data to the Semantic Repository were not exposing IoT endpoints.

2.1.2 Architecture and workflow

2.1.2.1 Architecture

Data Assembly and Services Portability Experiment (see Figure 3) is made of multiple components classified as:

- Backend components
    - *Semantic Mediation Gateway (SMG)*: fetches the data from FIESTA. It interprets always the role of FIESTA user in the FIESTA use-cases depicted in **(FIESTA-IoT D5.1, 2016)** and in the previous section. All the observations acquired are at the same time pushed to both the Context Management component and to the Contextualization Data Analytics component.
    - *Contextualization Data Analytics:* performs data analytics algorithms in order to infer new situations
    - *Context Management (CM):* manages context data (both observations and contextualized data). It stores and indexes data historically and exposes the data via an API.
- Frontend component:
    - *Dashboard:* offers a GUI to the experiment users and interact with the Context Management for retrieving the needed data.

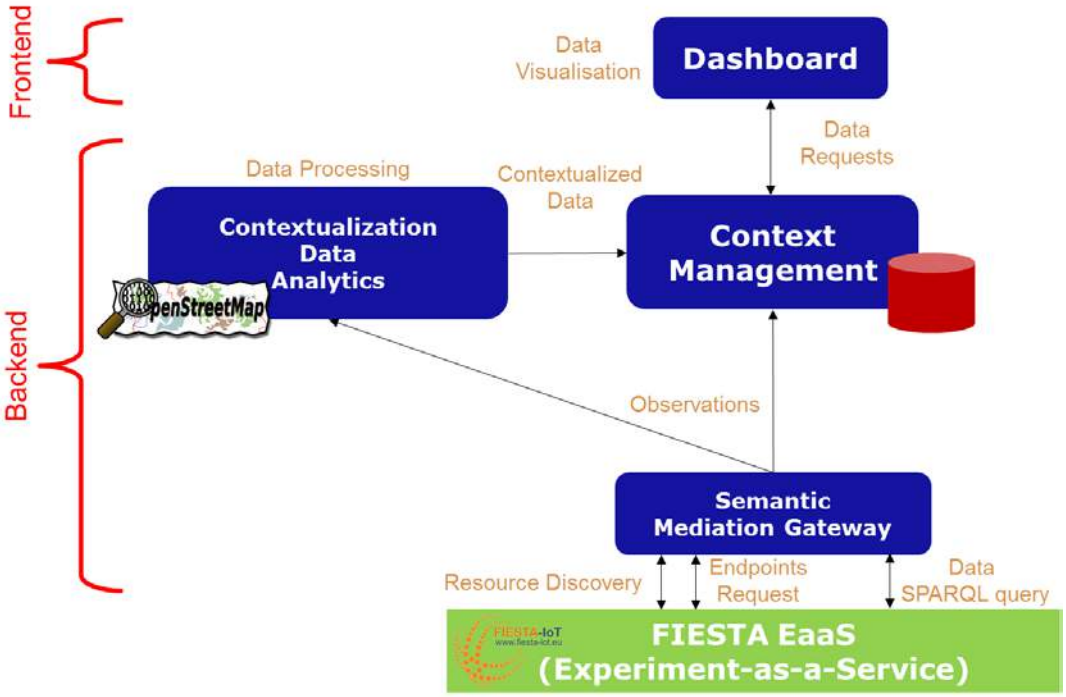

Figure 3 Smart City Magnifier architecture

2.1.2.2 Data acquisition workflow

The Semantic Mediation Gateway (SMG) component is in charge to retrieve IoT data from the FIESTA platform. In order to acquire that, it implements two different workflows at the same time, with the aim to get data from all the testbeds conntected to FIESTA.

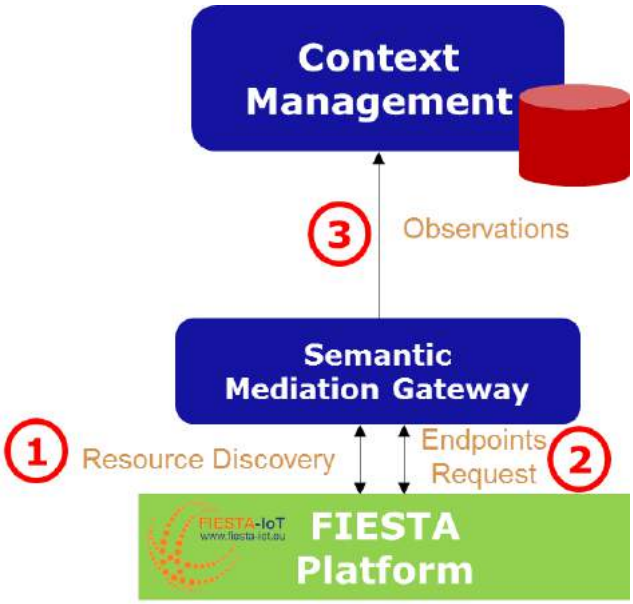

Figure 4 Data acquisition through endpoints workflow.

The first workflow (see Figure 4) corresponds to the first use-case of the Observations-oriented analytics where IoT service endpoints are available.

1. First the SMG discovers the list of resources,
2. The SMG starts to poll periodically each of the endpoints in order to get data.
3. The data is then forwarded to the Context Management.

Step 1 is periodically repeated, and in case the list of resources changes, the list of endpoints contacted at Step 2 is updated.

The resource discovery is executed via a SPARQL query to the FIESTA platform, similar to the following one:

```
PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: < http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?dev ?qk ?endp ?lat ?long
WHERE {
        ?dev a ssn:Device .
        ?dev ssn:onPlatform ?platform .
        ?platform geo:location ?point .
        ?point geo:lat ?lat .
        ?point geo:long ?long .
        ?dev ssn:hasSubSystem ?sensor .
        ?sensor a ssn:SensingDevice .
        ?sensor iot-lite:exposedBy ?serv .
        ?sensor iot-lite:hasQuantityKind ?qkr .
        ?qkr rdf:type ?qk .
        ?serv iot-lite:endpoint ?endp .
}
```


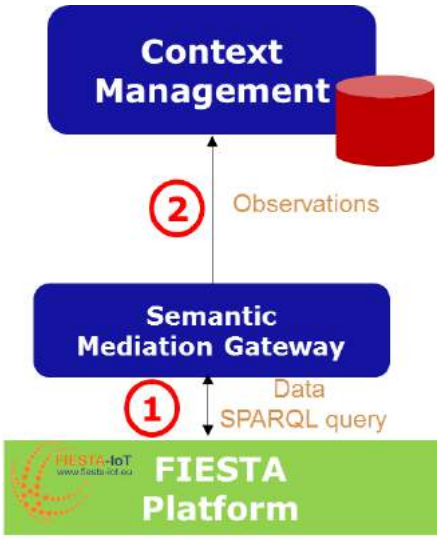

Figure 5 Data acquisition via semantic data repository workflow

The second workflow (see Figure 5) corresponds to the second use-case of the Observations-oriented analytics where no IoT service endpoints are available.

1. The SMG periodically polls the FIESTA platform for historical data with a SPARQL query specifying a time-window (which ends to the time of the request).

2. All the data retrieved is then forwarded to the Context Management which stores it.

Step 1 is periodically repeated with adjacent time-window.

The data SPARQL query used for retrieving the historical data is similar to the following one:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX oneM2M: <http://www.onem2m.org/ontology/Base_Ontology/base_ontology#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX qu: <http://purl.oclc.org/NET/ssnx/qu/qu#>
PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX m3-lite: <http://purl.org/iot/vocab/m3-lite#>
PREFIX dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT DISTINCT ?qkClass ?lat ?long ?time ?sensor ?dataValue ?observation
WHERE {
        ?observation a ssn:Observation .
        ?observation geo:location ?point .
        ?point geo:lat ?lat .
        ?point geo:long ?long .
        ?observation ssn:observationResult ?sensOutput .
        ?sensOutput ssn:hasValue ?obsValue .
        ?observation ssn:observedBy ?sensor .
        ?observation ssn:observedProperty ?qk .
```



```
            ?obsValue dul:hasDataValue ?dataValue .
            ?observation ssn:observationSamplingTime ?instant .
            ?instant time:inXSDDateTime ?time .
            ?qk rdf:type ?qkClass .
            FILTER (
                  (?time >="STARTTIME_PLACEHOLDER"^^xsd:dateTime)
            )
}
```

where instead of the "*STARTTIME_PLACEHOLDER*" it is placed the date at the time of the query.

In order to request the data, every HTTP request to the FIESTA platform is carrying the authorization token as an HTTP header.

2.1.2.3 Data contextualization workflow

The backend components perform data analytics task in order to compute the Smart City Magnifier indicators. The output data is again stored in the Context Management.

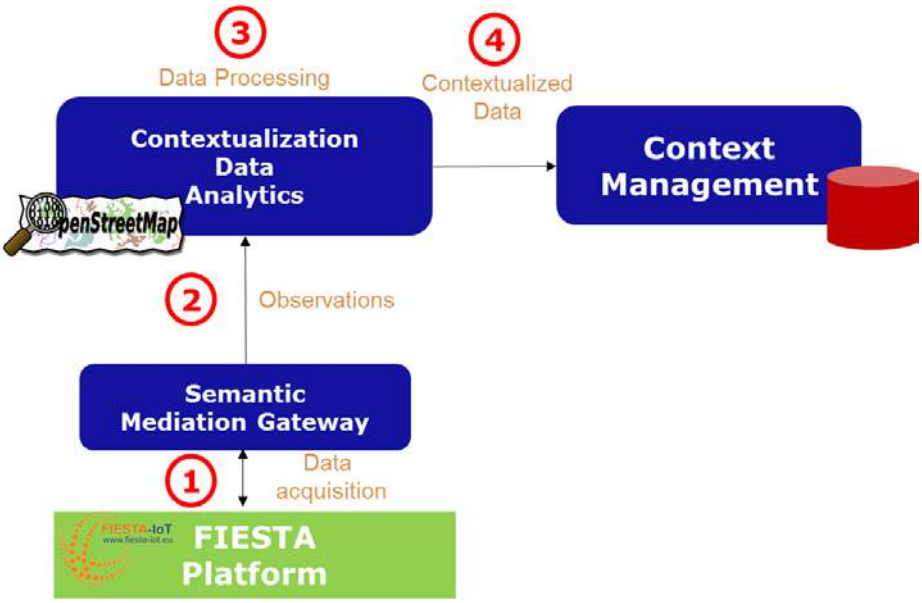

Figure 6 Data contextualization workflow

The workflow is depicted in Figure 6:

1. The Semantic Mediation Gateway (SMG) acquires data from the FIESTA platform as described in the section 2.1.2.2.
2. All the observations acquired are forwarded to the Contextualization Data Analytics component.
3. The Contextualization Data Analytics component applies algorithms in order to contextualize the observations by their location and infer situations through the instantiation of analytics functions. Contextualizing, in this scope, is the act of inferring the location context (e.g. a building, a street, a square, a suburb, a city etc.) to which each geotagged observation belongs. A new contextualized

entity is a Virtual Entity (VE). For every VE a set of analytics functions are executed with the compute data statistics on observation (average, minimum, maximum) and sensor deployment quality (observation density per area, number of active sensors of a certain type per virtual entity).

4. The inferred situation is then pushed to the Context Management.

2.1.2.4 Data Visualization workflow

The frontend component, the Dashboard, is in charge of offering a GUI to the Smart City Magnifier user, for showing both acquired observations and inferred situations (see Figure 7).

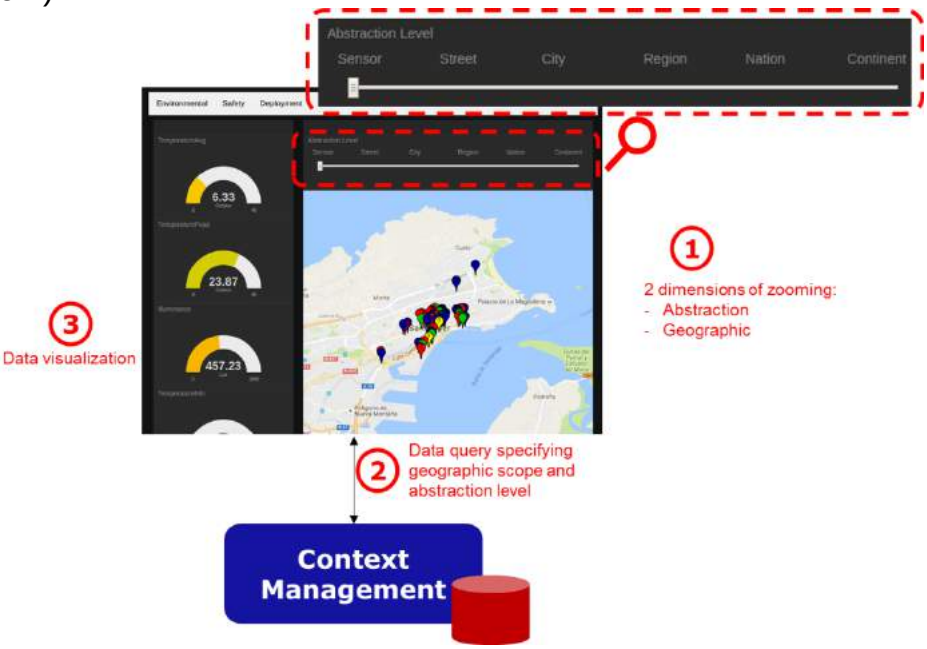

Figure 7 Visualization workflow

1. The user configures the wanted grade of zooming over the two offered dimensions, geographic and abstraction, respectively by modifying the scope and zoom of the geographic map and change the cursor of the “Abstraction level” sliding bar.
2. The Dashboard interprets the configurations of the graphical widget. Such setup is transformed in a data query to the Context Management (CM) which replies with the dataset.
3. The dataset is displayed in the graphical widgets. In the map the markers are spotting the location of the sensors (in case of a pure observation) or of the contextualized Virtual Entity (in case of higher abstraction level such as “Street”). The colour of the marker indicates the status of the situation: green for a good status; yellow for a warning; red for an alert; blue for available value but unknown situation meaning. The gauge widgets are showing an aggregation all over the map of the shown situation.

2.1.3 Dataset used: FIESTA-IoT Ontology concepts used towards building the queries

Following is the description of the dataset used (each field is tagged with the FIESTA-IoT ontology class):

- *dul:hasDataValue*: the pure observed value;
- *rdf:type (and its subclasses):* the quantity kind of the observed value;
- *geo:lat:* the exact latitude at which the value has been observed;
- *geo:long:* the exact longitude at which the value has been observed;
- *time:inXSDDateTime:* the timestamp of the observation;
- *ssn:observedBy:* the sensor that made the observation;
- *iot-lite:endpoint:* the IoT endpoint exposing last observed data of a particular sensing device.

2.1.4 Outcomes

With the first implementation of our experiment, we achieved a full Smart City application formed by a backend analytics engine and a Smart City Dashboard (see Figure 8).

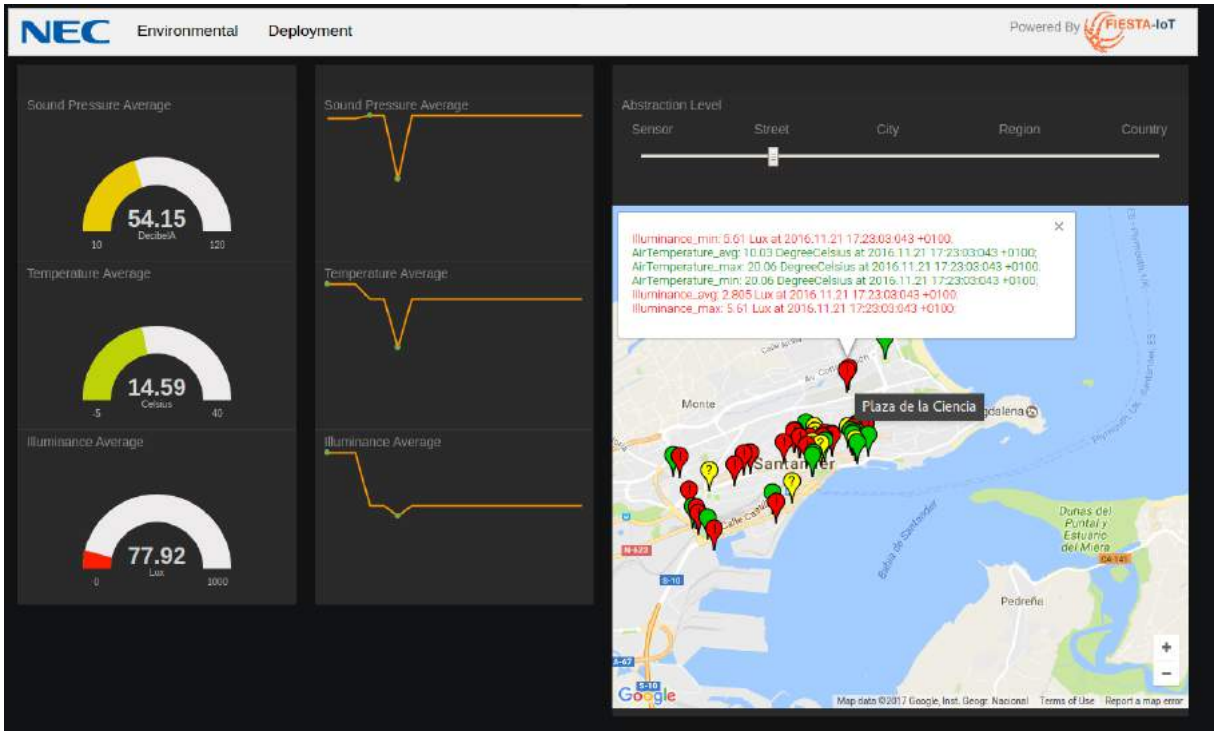

Figure 8 Smart City Magnifier

The first is able to infer situations of all kind of geographic location (e.g. streets, buildings, cities). The backend component is able to automatically infer new Virtual Entities to which the pure data, fed by the FIESTA-IoT platform, belongs.

The dashboard is a complete Smart City Dashboard for visualizing the outcome of the inference of the analytics in a map based widget where the situations are displayed as traffic-light color-schema markers for visualizing their status. In addition the situations are summarized over the geographic scope of the map on a set of single situation widgets (gauge widget and time series widget on the left side of Figure 8). Finally the analytics outcome have been classified over the geographic abstraction level (see Figure 9), selectable interactively with a slide-bar, and over the situation scopes (i.e. Environmental Scope and Deployment scope, see Figure 10).

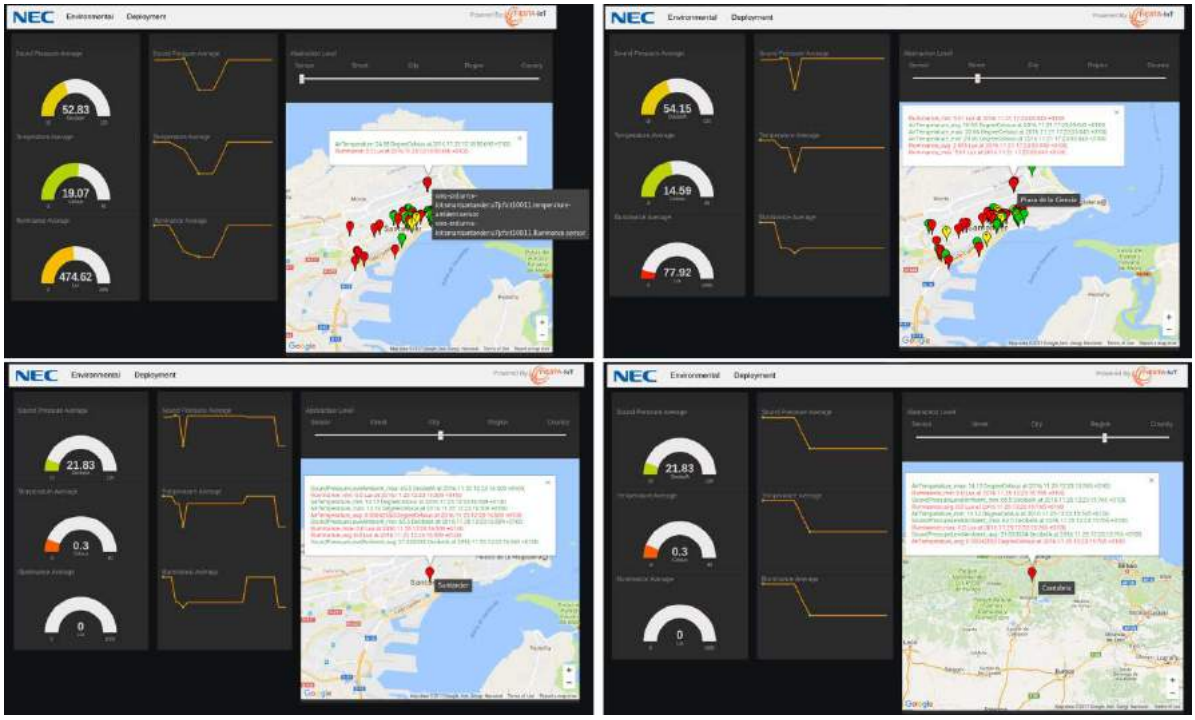

Figure 9 Smart City Magnifier: different geographic abstraction level selected with the slide-bar

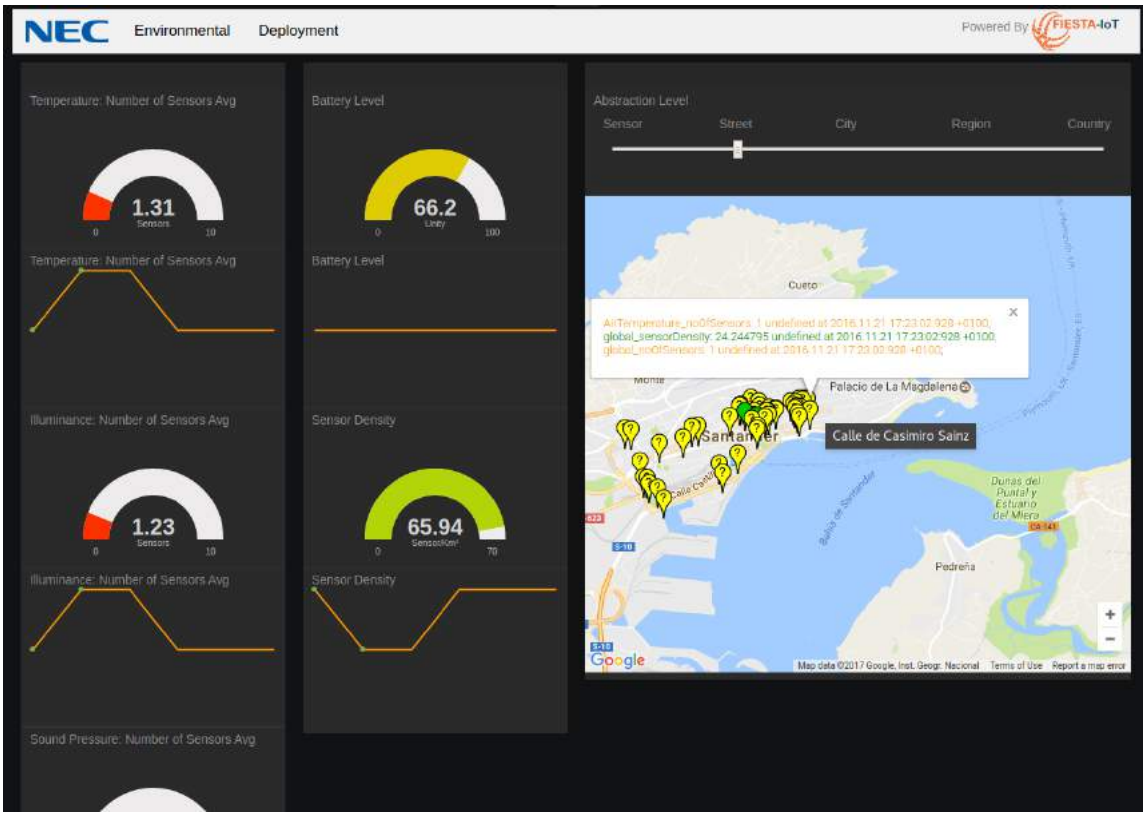

Figure 10 Smart City Dashboard: deployment situations scope

The experiment shows also its portability, since the same dashboard can visualize situations of one or the other corner of the world, seamlessly, simply moving the geographic scope of the map.

2.1.5 **Future work**

The Data Assembly and Services Portability Experiment is not yet completed and several aspects needs to be improved and enhanced in order to consider it fully satisfying the foreseen outcomes. In particular the following aspects will be addressed during the 3rd year of the FIESTA-IoT project:

- More analytics function for gathering more insight of a city.
- Perform data analytics algorithms over data coming from multiple testbeds (the actual limitation is the fact that the data analytics algorithms have a geographic scope of a size at most of a country and the testbeds are dislocated in different countries).
- Implements the Resource-Oriented analytics

2.2 Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access

As its name explicitly states, this experiment focuses on the (dynamic) harvest of IoT-based data in a testbed agnostic manner. Said in layman's terms, we aim at

retrieving data from sensors coming from heterogeneous platforms (as the ones that compose the FIESTA-IoT federation) in a single and common solution. For this experiment, and according to the legacy description of this pilot, we only focus on the weather/environmental domain. Namely, we will only show resources and observations that have to do with a subset of physical phenomena (e.g. temperature, illuminance, wind speed, etc.), where external running this experiment will be able to see the resources on a map and dynamically select subsets of them, in order to play around with the information (i.e. observations) the actual sensors generate.

Amongst the set of features that we support in this experiment, we stand out the following ones: *graphical representation of resources, location and phenomena-based resource discovery, retrieval of observations, combination of data for the generation of statistical analysis, graphical representation of these stats*, etc. The rest of the section addresses a deeper description of each of them.

The main challenges that are pursued by this experiment can be grouped as follows, embracing up to three different targets:

- **Guidance to third parties**: In order to provide some introductory guidelines to external users, we can see this as the entry point to the experimentation realm.
- **Platform performance assessment**: At the same time the experiment is running, we gather data of each of the operations that do interact with the FIESTA-IoT platform. This way, we receive some feedback of the experience achieved by experimenters and might also use this information for internal purposes (e.g. accounting, optimization, etc.)
- **Exportation of tools**: The way the experiment has been implemented allows us to straightforwardly export and encapsulate each of that shape it. Beside this, the nature of this experiment will follow an open-source approach, so third-parties might take all they need just by grabbing the piece of code that suit them.

2.2.1 Use-case selection

During the design phase of the application (addressed in (FIESTA-IoT D5.1, 2016)) we provided a mockup of the user interface that we designed in order to cover all the objectives and KPIs that were used to streamline the experiment. When it comes the time to actually implement the application, we have come to the look-and-feel shown in Figure 11 (excepting the red elements, which are there for explicative purposes). As can be appreciated, we have tried to port all the elements that were defined during the specification phase. Likewise we did in the abovementioned deliverable, we can easily split the interface into a number of use cases that actually compose the full story. In this section we proceed to briefly outline them, whereas we will break them down to explain how they operate in Section 2.2.2. Before proceeding with the individual description of each of them, the reader shall take into account that we are presenting in this document the ones that are completely functional at the time of writing, leaving aside the rest for the next version of the deliverable.

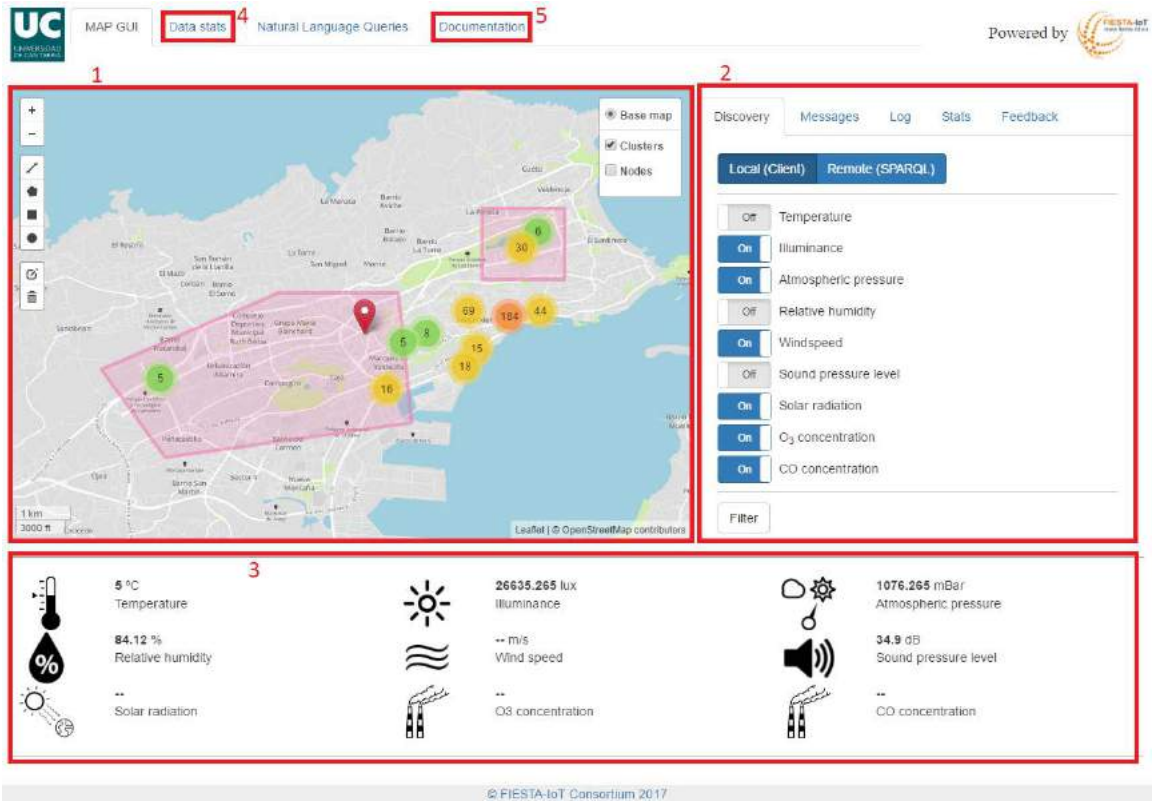

Figure 11. Screenshot of the actual status of the so-called "Dynamic Discovery"

2.2.1.1 Map-based I/O

In this first (and main) tab, the most remarkable element is a map (framed as '1' in the figure) where we can graphically see where the different resources coming from the FIESTA-IoT platform are physically allocated (at the moment of the so-called "*resource discovery*" stage).

However, the actual role of this map goes beyond this point: by clicking on any individual marker (we can thus associate the item marker to a resource itself, the system will automatically invoke the corresponding IoT service in order to retrieve all the subjacent observations (i.e. the latest ones) from that particular node[2], as shown in the outcome subsection (see 2.2.4) In addition to this, the framework we have used provides a tool that permits the creation of "graphical assets", such as polylines, rectangles, polygons or circles. We will leverage them for the manual and interactive selection of nodes, whose date will be used as input for use cases 3 and 4 (Sections 2.2.1.3 and 2.2.1.4, respectively).

2.2.1.2 Side map tab menu

Just aside the map we can find (element numbered as '2' in) a side menu that is provided to complement its behaviour. Split into 5 categories, we support the following features:

- Phenomena-based discovery:

[2] In our approach, a node might contain more than one sensor, so a single "click" might imply more than one IoT service invocation.

In order to showcase a dynamic selection or filtering of resources, we offer the possibility of selecting the subset of physical phenomena (i.e. quantity kinds in the m3-lite taxonomy) on demand, based on a toggle group. Thus, the map will only display those nodes that own at least one sensor of any of the enabled categories. In addition, we offer two different approaches to cope with this operation, as can be appreciated in the upper part of the element:

- Local (Client). In this case, the whole list of resources is available in the client's memory. Therefore, we proceed to span (iterate) among all of them displaying only the appropriate ones.
- Remote (SPARQL). The other option is based on a semantic solution, in which we generate a SPARQL query in which we request a particular subset of quantity kinds.

- Message exchange between experiment and FIESTA-IoT platform:

  All the operations that bring about a message exchange between the experimentation side and the FIESTA-IoT platform are recorded and displayed as part of the application. This way, experimenters can quantitatively check the overall performance of the operations that are being carried about behind the curtains.

- System logging (local/remote):

  Similar to the previous point, we offer a logging-like visualization for experimenters, presenting not only the exchange of messages, but also every single operation executed by either the server or client side of the experiment.

- Stat viewer:

  On top of these "loggers", experimenters might also want to gaze at some pieces of information that come from the realized operations. For instance, we have included here the total number of resources federated in the FIESTA-IoT platform, the ratio of them that have passed the phenomena-based filtering, those that have been "selected" by means of the "graphical assets", the number of these assets that are deployed all over the map, etc.

- Feedback:

  This element is not part of the current version of the experiment. We will come back to it in the future D5.2.2 (M35).

2.2.1.3 Basic output: weather station with the average values

One of the actions that are executed atomically after the creation/edition/deletion of the previously named "graphical assets" is the clustering of all the nodes into a new group of "selected devices". After this, the experiment is in charge of retrieving all the latest observations measured by these resources and properly combining them altogether (in a per phenomenon basis), yielding a weather station-ish outcome that displays the average values observed. However, this operation goes a step beyond and automatically triggers a polling service that will periodically poll the information from this list of nodes and properly updates the stats of this weather station.

2.2.1.4 Advanced output: graphical data analysis

Whereas the previous feature is a sample of this dynamic selection of resources and observations, it is here where we can directly exploit the potential of this combination of data or “composition of IoT services”. In this tab (as can be seen in Figure 11 – element number 4 – it is not part of the main view), we will output graphical and statistical information, whose input values come from the observations explained in the previous section. To cite a couple of examples, we will represent for instance a timeline of the evolution of data throughout time (combining historical data with info gathered from the periodic polling). Additionally, we can also display a detailed statistical analysis in a per phenomenon basis at a time instant ‘t’.

2.2.1.5 Documentation

During the implementation phase, we have not left aside one of the requirements that was part of (FIESTA-IoT D2.1, 2015): “*30_NFR_ACC_FIESTA_well_documented - FIESTA-IoT must be well-documented*”. Despite the fact that this experiment is not a explicit part of the platform, we do believe that it can be used by externals to learn how to actually interact with the platform.

2.2.2 Architecture and workflows

In a nutshell, the architecture that defines this experiment can be seen as shown in Figure 12. As can be seen, we have split the functionalities at the application level into two standalone modules: a server that handles the interaction between the experiment per se and the FIESTA-IoT platform and a web application (client) in charge of all the visualization and the interaction with users. In the next paragraphs we briefly streamline the main highlights of the experiment.

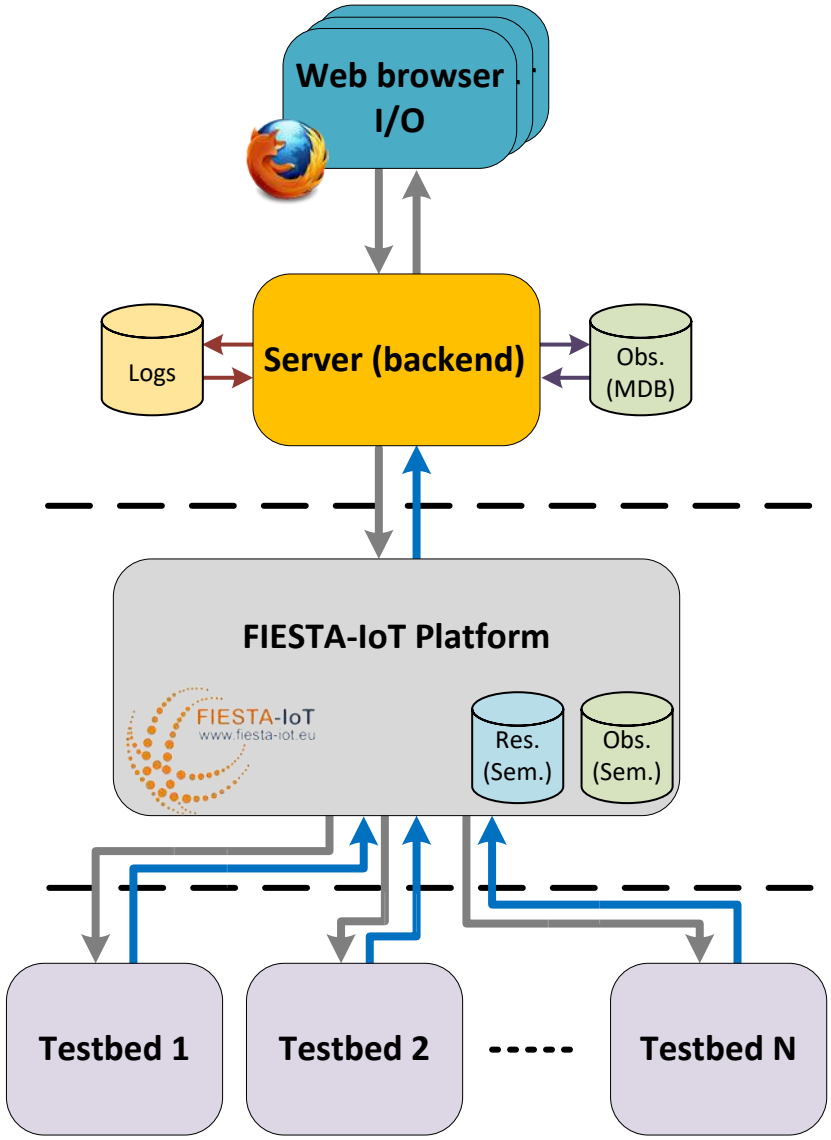

Figure 12. Dynamic discovery of resources generic architecture

- Client/server approach
  - The server side is in charge of the global discovery of resources and the execution of complex operations, like location or phenomena-based queries. In other words, it is the communication point between the FIESTA-IoT platform and experimenters that execute the web application in their own systems (see below). Therefore, there will only be a single instance of the server.
  - The client (i.e. browser app) is the actual interface between users and the server. As such, multiple clients might run at the same time. Amongst its duties, it undertakes the role of displaying the information on the map and on the visual tools, as well as of getting data from users, as has been introduced before. It fosters all the set of features that was introduced in (FIESTA-IoT D5.1, 2016), where we did not have in mind the breakdown of the experiment into these two parts, though. Hence, we have managed to significantly reduce the

computational load at this side, moving most of the heavy operations to the server, that is run within the FIESTA-IoT infrastructure (Annex I).

- To solve geographical problems (e.g. select the nodes within an area), we offer two options: on the one hand, through explicit SPARQL queries addressed to the FIESTA-IoT platform (end-to-end operation); on the other hand, we rely on a library that can solve this type of problems by means of geometrical operations. Through this, we can infer the most efficient choice in terms of performance.
- Internal storage for historical data (at experiment level). In order to allow the re-run of off-the-shelf plots, we internally keep a copy of the data at the experiment side (instead of querying all the time to the platform). This task will be carried out at the server side, so that any observer running the experiment might start with the datasets available at that moment, instead of having to query for them (thing that would overload the platform). As can be seen, in this version of the experiment we rely on a "*mongo dB"*[3] storage system for the recording of all the received observations (avoiding the replication of data).
- For the sake of carrying out a thorough analysis of the platform, this experiment includes a set of performance markers that help us evaluate and quantify the performance of the whole system. In essence, we do keep track of all the relevant operations that have to do with the core components and its interaction with users. As can be seen in Figure 12, we have defined a persistent place where we will store all the system logs.
- On top of all this, it is worth highlighting that we have opted for playing a role of an "advanced experimenter", so the communication channel with the core platform has been carried out through the direct interaction across the IoT Registry API[4]
- As for the communication between client and server at experimentation level, we have used a secure communication channel that protects and encrypts the exchange of messages between them.

Once we have briefly outlined the architecture that describes the relationship between the experimentation side and the FIESTA-IoT platform, it comes the time to dig into each of the atomic use cases that are actually behind the functionalities described so far.

2.2.2.1 Initial discovery of resources (backend) + Map visualization (Web browser)

First and foremost, before getting any kind of data, we have to know the assets that are available within the FIESTA-IoT federation. By exploiting the agnosticism that is brought about by the project, we will discover, with a single and common query (i.e. SPARQL), the whole set of resources that have been registered till the moment it is executed. Namely, the query (SPARQL) that is used for this phase is the following one:

[3] https://www.mongodb.com/

[4] https://platform.fiesta-iot-eu/iot-registry/docs/api.html


```
PREFIX iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?dev ?sensor ?qk ?unit ?endp ?lat ?long
WHERE {
        ?dev a ssn:Device .
        ?dev ssn:onPlatform ?platform .
        ?platform geo:location ?point .
        ?point geo:lat ?lat .
        ?point geo:long ?long .
        ?dev ssn:hasSubSystem ?sensor .
        ?sensor a ssn:SensingDevice .
        ?sensor iot-lite:exposedBy ?serv .
        ?sensor iot-lite:hasQuantityKind ?qkr .
        ?qkr rdf:type ?qk .
        ?sensor iot-lite:hasUnit ?unitr .
        ?unitr rdf:type ?unit .
        ?serv iot-lite:endpoint ?endp .
            VALUES ?qk {m3-lite:AmbientTemperature m3-lite:AirTemperature m3-
        lite:TemperatureSoil m3-lite:TemperatureAmbient m3-lite:Illuminance m3-
        lite:AtmosphericPressure m3-lite:RelativeHumidity m3-lite:WindSpeed m3-
        lite:SoundPressureLevel m3-lite:SoundPressureLevelAmbient m3-lite:SolarRadiation
        m3-lite:ChemicalAgentAtmosphericConcentrationCO m3-
        lite:chemicalAgentAtmosphericConcentrationO3}.
}order by asc(UCASE(str(?qk)))
```

Taking into account that the registration of resources is not a frequent activity, we do not need to re-discover of resources very often at the server level. Thus, this process will be performed once per hour (albeit this rate might change in the future) so as to periodically discover. Figure 13 presents the sequence diagram of messages exchanged in order to get the whole list of resources registered. Namely, the meaning of each of them is the following:

1. The server synchronously addresses (as mentioned before, once per hour) the previously introduced SPARQL query to the FIESTA-IoT platform.
2. The platform internally processes the query and sends back the corresponding response to the server, which keeps the resultset in memory, waiting for requests coming from the client side.
3. In a completely independent manner of the previous two messages, a client runs the application. Immediately, a resource discovery request (in this case, it is not a SPARQL query) is sent from the browser to the server. We have to take into account that this process does not lead to the exchange of any message between the experiment and the platform.
4. The server proceeds to reply the client with the list of resources discovered in the first two steps.

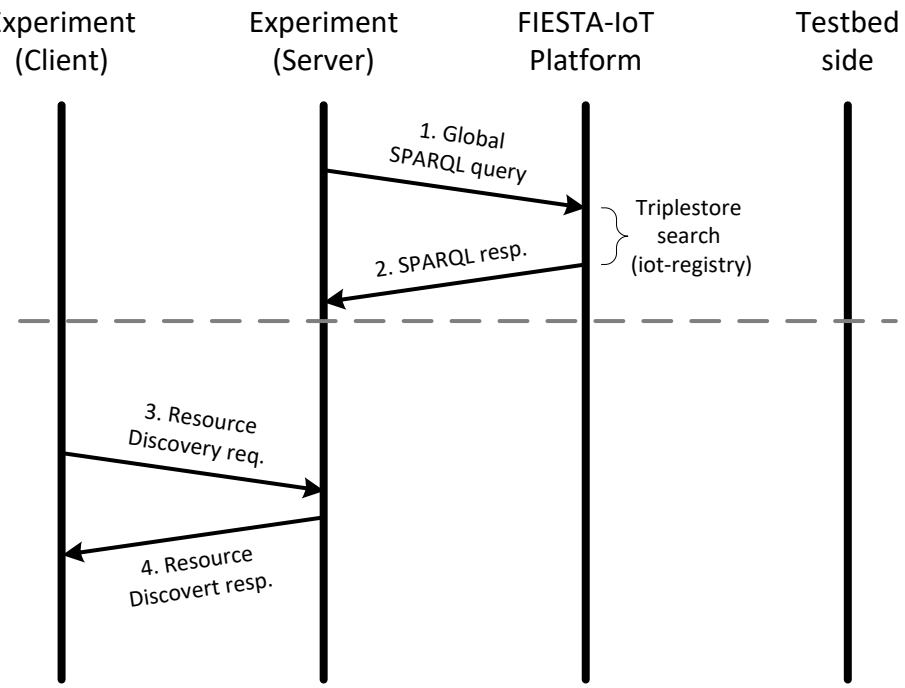

Figure 13. (Dynamic Discovery) Resource discovery sequence diagram

2.2.2.2 Data retrieval (through IoT Service endpoint)

Once we know the assets and where they are, the next step is to harvest real data (i.e. observations from the sensors). In the current version of the experiment, every time we click on a node's marker, we make use of the IoT Service endpoints (included in the resource description) in order to address a message and request the last observations measured by that particular node. We can see in Figure 14 as this leads to an end-to-end operation, whose sequence diagram is depicted below.

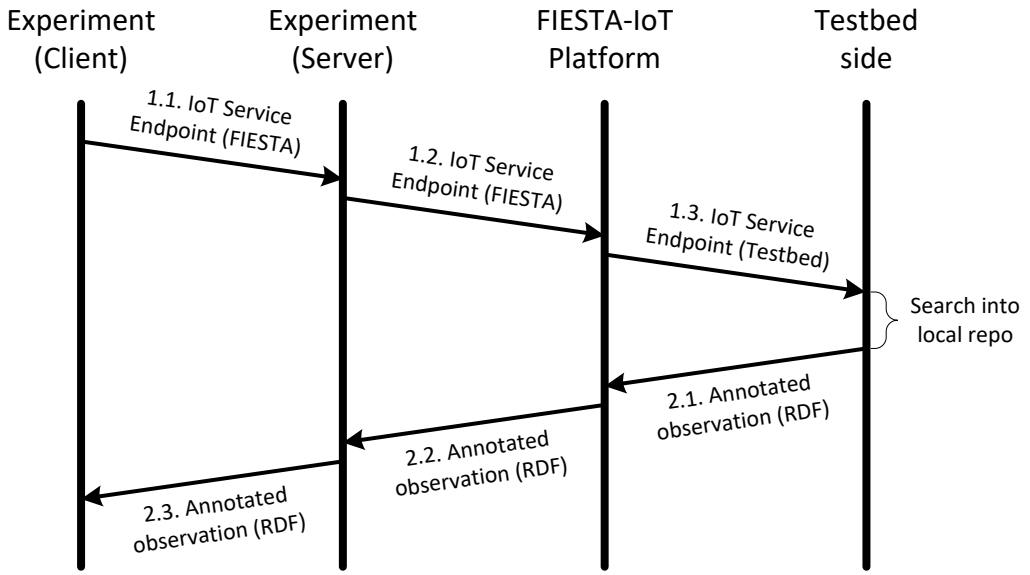

Figure 14. (Dynamic Discovery) Getting a last observation from an iot-service endpoint sequence diagram

1. As has been mentioned, the resource description contains the address of the IoT endpoint that actually expose that resource (i.e. a simple GET message). If we take Figure 14 as an example, the clicked node hosts five different sensors, that is five different endpoints. Thus, five messages addressed to five different endpoints will be sent.
2. Internally, the testbed retrieves the observation and sends it back in RDF format, fulfilling the FIESTA-IoT semantic model. Following the previous example, five annotated observations will be sent back to the client. It is worth highlighting that these messages' formats are actually RDF (Resource Description Format) documents, so the application has to parse them accordingly.

2.2.2.3 Phenomena-based filtering (resource discovery)

One of the features supported at the client side is the interactive discovery of resources based on their sensing capabilities, thus filtering out those that are not able to measure a particular subset of physical phenomena. As shown in Figure 11, when the option "Remote (SPARQL)" is enabled and we click on the "Send query" button, a SPARQL will be automatically generated by the client and delivered, across the server, to the FIESTA-IoT core. Regarding the query per se, it is basically alike that of Section 2.2.2.1. The main difference is the array of physical phenomena that is in the body; unlike the static list of the above case, we only append those QuantityKinds that are enabled in the interactive toggle group.

Figure 15 summarizes the sequence diagram observed in the system.

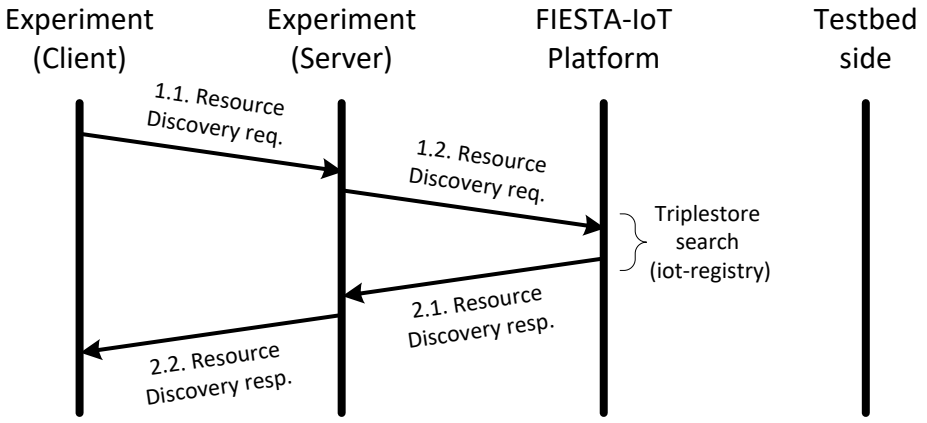

Figure 15. (Dynamic discovery) Phenomena-based filtering sequence diagram

NOTE: Even though in *(FIESTA-IoT D5.1, 2016)* we stated that we would support location-based discovery, now we believe that this feature would not bring about any insightful outcome but graphically showing resources that appear and disappear on the map.

2.2.2.4 Location-based clustering (data retrieval)

As was introduced before, we support the use of "graphical assets" to interactively cluster nodes by "drawing" simple objects on the map. Behind the association, a SPARQL query is generated by the client in order to get the last observations sensed by this new "cluster". Since this is a client-based request, the server does not need to record the results (this might lead to an unnecessary computational overhead, due to

the parsing and further filtering of all the duplicated information). All in all, the exchange of messages in this process is reflected in Figure 16.

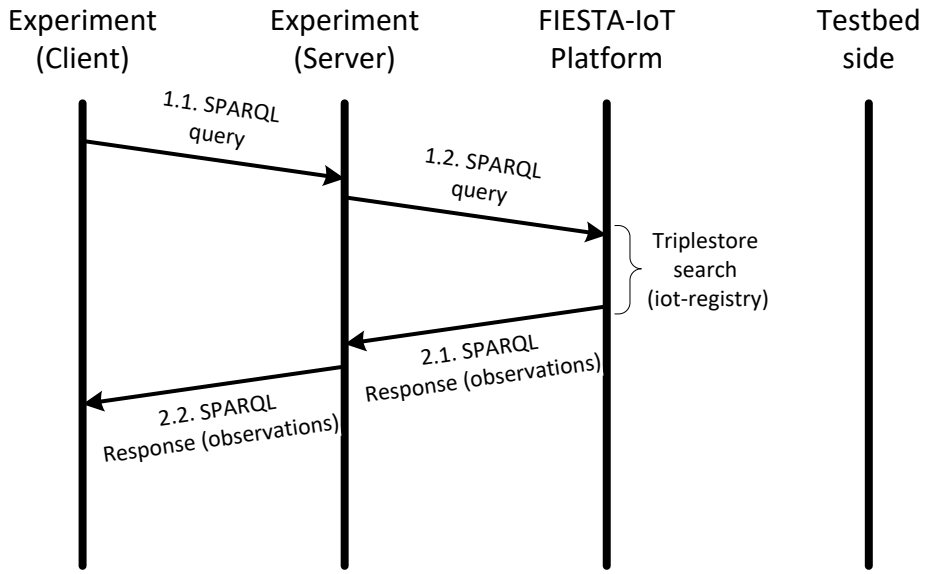

Figure 16. (Dynamic discovery) Location-based clustering data retrieval sequence diagram

In order to get the corresponding list of "last observations", we have used the following SPARQL sentence, where we manually specify the sensors that are inside the regions drawn on the map (or, in case of the polyline, closer than a predefined distance).

```
Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
Prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
Prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
Prefix time: <http://www.w3.org/2006/time#>
Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
select ?s (max(?ti) as ?tim) ?val ?lat ?long ?qk ?unit
where {
   ?o a ssn:Observation.
   ?o ssn:observedBy ?s.
   VALUES ?s {<sensorID_1> <sensorID_2> … <sensorID_N>}.
   ?s iot-lite:hasQuantityKind ?qk.
   ?s iot-lite:hasUnit ?unit.
   ?o ssn:observationSamplingTime ?t.
   ?o geo:location ?point.
   ?point geo:lat ?lat.
   ?point geo:long ?long.
   ?t time:inXSDDateTime ?ti.
   ?o ssn:observationResult ?or.
   ?or  ssn:hasValue ?v.
   ?v dul:hasDataValue ?val.
   {
      select  (max(?dt)as ?ti) ?s ?qk ?unit
      where {
```

```
            ?o a ssn:Observation.
            ?o ssn:observedBy ?s.
            ?s iot-lite:hasQuantityKind ?qk.
            ?s iot-lite:hasUnit ?unit.
            ?o ssn:observationSamplingTime ?t.
            ?t time:inXSDDateTime ?dt.
        }group by (?s)
    }
} group by (?s) ?tim ?val ?lat ?long ?qk ?unit
```

2.2.2.5 Historical data

Alongside the general discovery of resources generated at server's booting time, it also sends another query, focused on getting all the observations captured by the FIESTA-IoT platform. It is worth highlighting that this operation is only run once. Another thing that is worth a comment is that we can limit the time period during which we want to get data from, avoiding to process huge amounts of data. As at the time of writing the document there are not many observations stored, we have opted for getting everything, thus "dumping" all the meta-cloud to the server side. The process is resumed in Figure 17, where:

(1) the server send the raw SPARQL to get all the observations stored at FIESTA-IoT level (below we can see the query itself), and in
(2) the FIESTA-IoT platform sends the response back to the server. At last, this server parses this message and stores all the data in its own database (MongoDB).

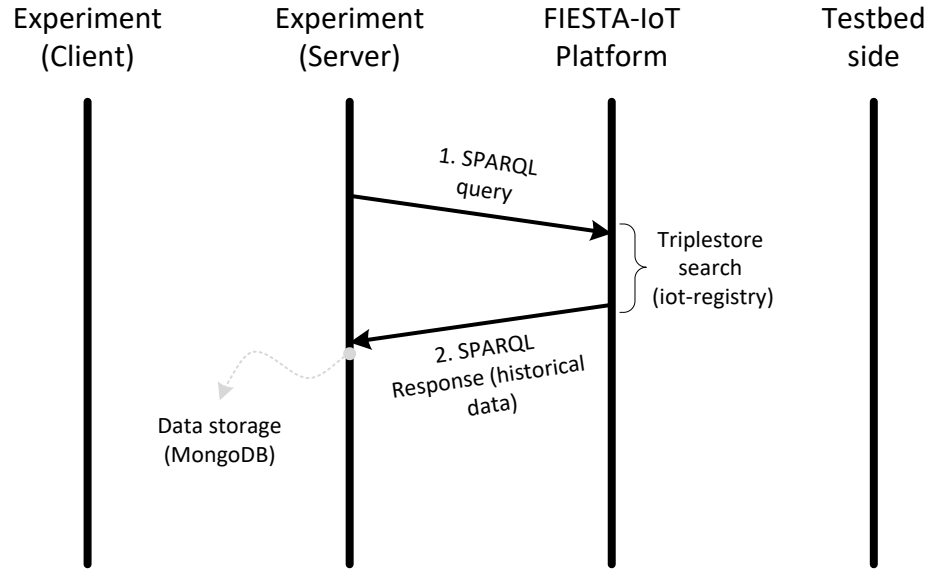

Figure 17 (Dynamic discovery) Historical data dump sequence diagram

Finally, this server parses this message and stores all the data in its own database (MongoDB).

```
Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
Prefix iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
Prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
Prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```



```
Prefix time: <http://www.w3.org/2006/time#>
select ?s  ?val ?lat ?long ?qk ?unit
where {
   ?o a ssn:Observation.
   ?o ssn:observedBy ?s.
   ?s iot-lite:hasQuantityKind ?qkr .
   ?qkr rdf:type ?qk .
   ?s iot-lite:hasUnit ?temp .
   ?temp rdf:type ?unit .
   ?o ssn:observationSamplingTime ?t.
   ?o geo:location ?point.
   ?point geo:lat ?lat.
   ?point geo:long ?long.
   ?t time:inXSDDateTime ?ti.
   ?o ssn:observationResult ?or.
   ?or  ssn:hasValue ?v.
   ?v dul:hasDataValue ?val.

} group by (?s) ?tim ?val ?lat ?long ?qk ?unit
```

2.2.2.6 Periodic polling for last observations

While the server is running, it periodically polls the FIESTA-IoT platform in order to get all the information. Such a process only implies the experiment's server and the FIESTA-IoT platform, where the server delivers a SPARQL like the one introduced in Section 2.2.2.4, albeit in this case we will not filter according to any sensor ID, but we will retrieve all the list of observations. Upon the reception of the SPARQL response, the server has to undertake the task of dropping out all the duplicated entries. This workflow is shown in Figure 18.

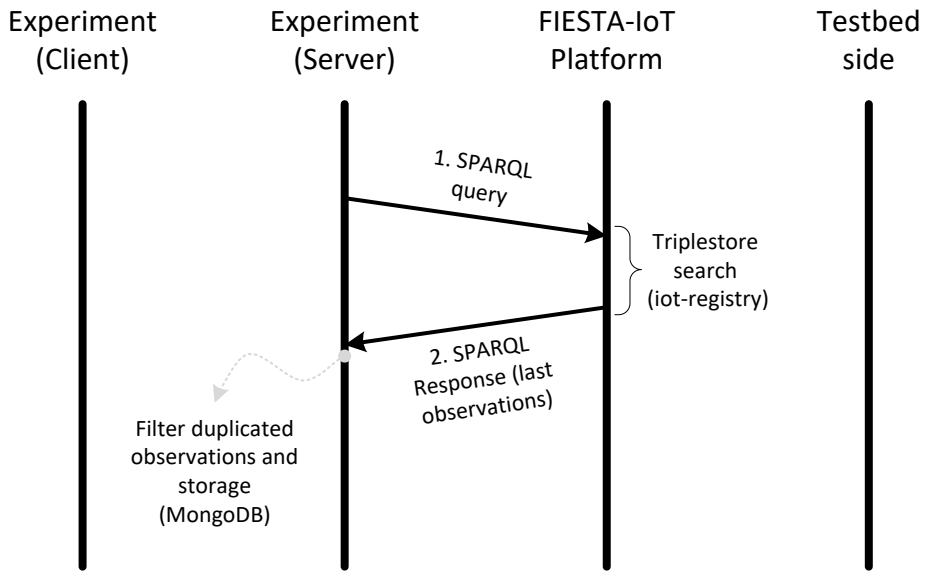

Figure 18 (Dynamic discovery). Storage of last observations from periodic polling sequence diagram.

We must remark that this particular service is the alternative to an asynchronous publish/subscribe operation (not ready at the time of writing the document).

2.2.2.7 Data visualization

One of the main outcomes of this experiment consists in the visualization of data coming for the sensors for further analysis. Indeed, this is the actual reason behind the storage of the observations. Hence, whenever a web client wants to load any of the graphical assets supported by the experiment (for instance, the evolution of the average data within a time frame), the client sends a request to the server's repository. Then, the response will be the input data of all this graphical elements.

2.2.2.8 Performance monitoring

Last, but not least, at the very same time we carry out the any of the aforementioned operations, the experiment's server keeps track of all of them. In other words, we record all the computational times consumed by each of this operations (regardless the location of the client executing the web application) so that we can extract information in the future about the overall performance of the platform.

Thanks to this, we can elaborate a set of good practices that might help us improve the quality of experience on not only experimenters, but also testbed providers (e.g. by optimizing the SPARQL queries to discover and handle resources).

2.2.3 Dataset used: FIESTA-IoT Ontology concepts used towards building the queries

As for the elements of the ontology that are used in this experiment, below we proceed to outline the purpose of each of them. Before starting, it is deemed necessary to differentiate between the two main phases that shape the experiment's lifetime: the resource discovery and the observation(s) retrieval.

In essence, when we talk about resources we need to extract the following information from the resource discovery:

- *Location of the sensor*, based on the *geo:lat* and *geo:long* classes
- *Sensor ID*, for displaying purposes (*ssn:Sensor*/*ssn:SensingDevice*)
- *Endpoint* that exposes the IoT Service; in this case, the gathering of the last observation (*iot-lite:endpoint*)
- *Physical phenomenon* observed by the particular sensor (*m3-lite:QuantityKind*)
- *Unit of measurement* bound to the data that will arrive as observations (*m3-lite:Unit*).

On the other hand, when it comes to the observation realm, apart from the core of the measurement per se (*ssn:Observation*), we have to answer the following questions:

(1) Who sensed the observation? – *ssn:Sensor/ssn:SensingDevice*
(2) Where? – *geo:location*
(3) When? – *Time:Instant*
(4) Type? – *iot-lite:QuantityKind* and *iot-lite:Unit*
(5) Value? – This corresponds to the actual values that are connected through the *dul:hasDataValue* data property

2.2.4 Outcomes

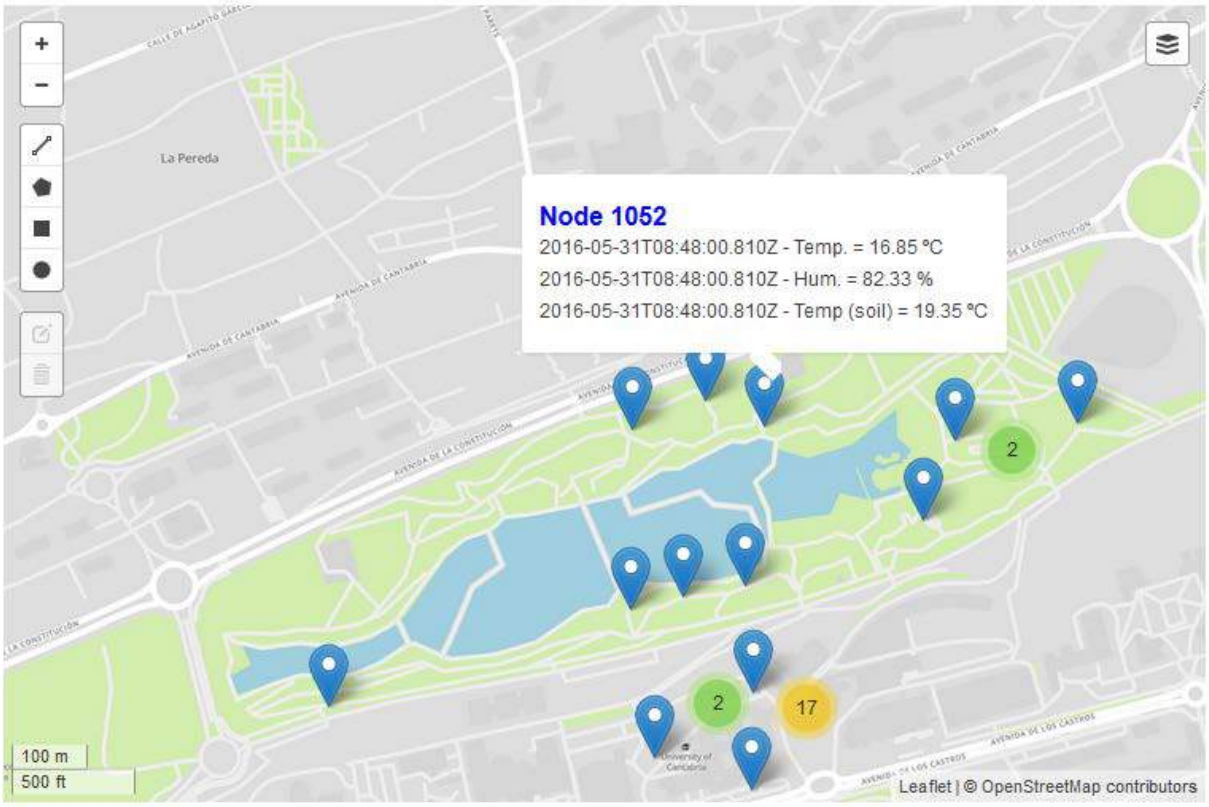

Figure 19 Visualization of the last observations measured by a particular node

In this section we sum up the main outcomes achieved in the current version of this experiment.

- Implementation of a client/server web application that focus on the dynamic discovery of IoT resources in a testbed agnostic manner.
- Visualization (on a map) of the resources registered in the FIESTA-IoT platform. This means that, at a first glance, users will not be able to distinguish whether a node belongs to FIESTA testbed A or FIESTA testbed B, since all of them are, in the end, FIESTA's resources.
- Local and remote discovery of resources.
- Interactive node clustering by means of graphical tools.
- Compilation of historical data, retrieved from the FIESTA-IoT platform.
- Representation of a weather station-like component, whose output is the result of the compilation of multiple data sources.
- Graphical representation of historical data, including composition of data from multiple resources (see Figure 20).
- Dynamic gathering of observations from the FIESTA-IoT platform.
- Monitoring of the messages exchanged between the different components of the whole system (experiment + FIESTA-IoT platform + testbeds).
- Performance analysis of both the experiment and the FIESTA-IoT platform based on a centralized logging system.

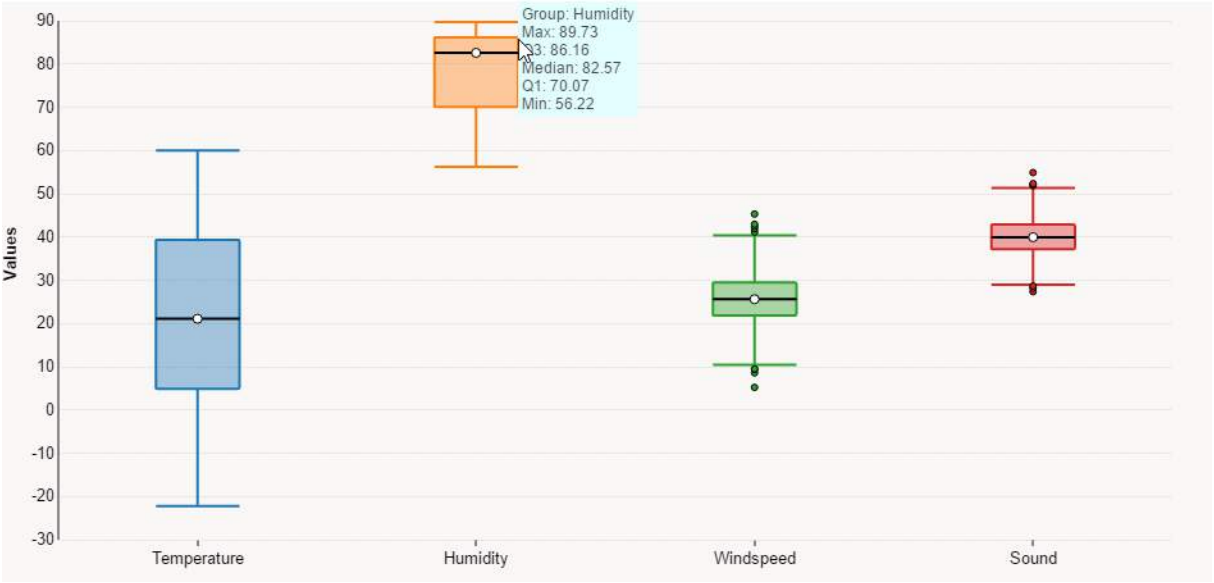

Figure 20. Sample of a graphical output of the processed data

2.2.5 Future work

Apart from all the functionalities that have been presented throughout this section, it is worth highlighting that the current version of the experiment is not definitive and there are still many open issues to be tackled in the third year. Hence, there is a number of features that are to be implemented and integrated before the project ends, as the ones listed below.

- Utilization of composed IoT services within the platform.
- So far, the graphical assets only yield a group of nodes. It would be interesting to let users define various subsets in order to compare among them.
- Subscription to data streams. At the time of writing, the asynchronous service is not an supported option in the FIESTA-IoT platform. As a consequence, we did have to seek an alternative to keep getting data from the meta-cloud. However, as soon as we can rely on this feature, we will shift from our periodic polling system to the subscription-based one.
- Creation of a FEDSPEC for the definition of the experiment. So far, playing the role of advanced experimenters has allowed us to manually interplay with the FIESTA-IoT registry API. Nevertheless, and in order to test another components of the platform, we might rely on the ERM and EEE functional components in order to:
    (1) Test its behaviour and performance,
    (2) Reduce the complexity of the experiment.
- Feedback from users. Even though we have not explicitly mentioned this feature in this deliverable, during the design phase we contemplated the possibility of enabling a place where experimenters might send information about potential misbehaviors in the platform.

- Foster a human language query processing. The potential of semantic technologies might be exploited by means of the use of natural language processor that can interpret questions written (the voice recognition is out of the scope of this project) in human language and give rise to e.g. regular SPARQL queries.
- Reutilization of components as external tools. The way this experiment has been implemented lets us straightforwardly extract and encapsulate the different features as standalone modules that can be used in other applications.

2.3 Large Scale Crowdsensing Experiments

The planned large-scale experiment enables the experimenter to understand the variations in the sound levels over the period of time and region. This experiment mainly focuses on the observations made available from the sound sensors available within a platform that are either mobile or static. Note that throughout this section we will use the terminology defined within the FIESTA-IoT ontology (FIESTA-IoT D3.2, 2016). The experiment follows the approach defined by FIESTA-IoT Meta-Cloud Architecture (FIESTA-IoT D2.4, 2015) where the experiment is submitted in the form of the DSL and then is executed by the Experiment Execution Engine (EEE).

As described in (FIESTA-IoT D5.1, 2016), various use cases are envisioned and will be implemented as a part of experiment FED-Spec (FIESTA-IoT Experiment Description). Within the FED-Spec the FIESTA-IoT Experiment Model Object (FEMO) consists of FIESTA-IoT Service Model Object (FISMO) to realise all the use cases. The complete FED-Spec for the experiment is made available as the Annex (see Annex II). The implemented use cases provide insights such as:

- What is the most recent sound level over a region?
- What is the variation of sound over time at a particular location?
- What is the variation of sound over time and over a region?
- What were the most noisy locations over time and over a region?
- What were the least noisy locations over time and over a region?

2.3.1 Use-case selection

We envision implementing all the use-cases that are described in (FIESTA-IoT D5.1, 2016). However, for this version we implement only one use-case that reports what are the most noisy places in the area. Thus we convert the requirement for this use-case to the FISMO and send the FISMO to the EEE for the execution.

Nevertheless, as different use-cases are complementing each other we like to build all uses-cases. The main use-case that is to be implemented and translated to the FISMO is the case where sound information is requested for the given region over a duration of time. Note that, as there is a “scheduling” attribute available in the FISMO, the duration of time can be specified within the “scheduling” attribute with the periodicity with which information is needed. The use-case query is defined such that the response of the query consists of only most recent observations. Thereby making

it essential for the experimenter to correctly configure the periodicity within the “scheduling” attribute.

The current support from the EEE component also makes it possible to poll this use case via “Polling” option. This aspect would make it possible to realise the use-case where only most recent observations are requested for the particular region.

The main use-case as described above, would provide most recent sound values coming from all the sound sensors within the specified region. Once such data is received at the experimenter end, while creating visualization selection of sound level values can be done to identify if the value is more/less than certain value. This makes it possible to realize the cases where the visual for most/least noisy locations within the region is to be shown. Nevertheless, the above cases can also be done using queries, for this version we implemented queries that are made available via experiment FISMOs. Note that we build all the queries and related FISMOs, however, at the experimenter side only most noisy location heat map is available.

2.3.2 Experiment architecture and workflow

The experiment workflow is same as described in Section 3.3.2 (FIESTA-IoT D4.1, 2015). Just focusing on the experimentation side and not on how observations from mobile and static devices are sent to FIESTA-IoT Meta-Cloud, the experiment workflow and architecture is same as that defined by FIESTA-IoT. Following list elaborates on the workflow:

1. The experimenter creates the FED-Spec (see Annex II) and specifies those attributes within FEMO and FISMO that are currently supported by EEE.
2. He then logs in into the FIESTA-IoT platform to get the token. In our case, we would use the FIESTA-IoT Portal to perform this step. All other steps that require interaction with the FIESTA-IoT Platform will also be using FIESTA-IoT Portal. The token is used in all the calls to the FIESTA-IoT platform.
3. He uses the ERM (Experiment Registry Module) user interface to send the FED-Spec and register it to the FIESTA-IoT platform.
4. Before proceeding to the next step, the experimenter makes sure that the service to receive data from the FIESTA-IoT platform is available. This service is used by the FIESTA-IoT platform’s EEE component to send the result of the query specified in the FISMO object.
5. He then opens the Experiment Management Console (EMC), selects the particular experiment to see the list of FISMOs attached to the experiment. Note that, the EMC interacts with ERM to get the required information regarding the experiment and interacts with EEE to get the status of execution of FISMO.
6. The experimenter then starts the execution of the FISMOs by toggling the start and stop button (for reference on Experiement Management console FISMO pane see Figure 21). This step would schedule the particular FISMO for specified duration and would provide a JobID to the FISMO. Upon a successful schedule, whenever the job ID is triggered, the query specified in the FISMO is executed on the FIESTA-IoT Meta-Cloud. The response

obtained is then forwarded to the location specified by the experimenter. In case the FISMO is already scheduled, the experimenters can pause/resume the triggering of the FISMO on the FIESTA-IoT platform.

7. The experimenter upon receiving the data from the FIESTA-IoT platform reads the data, performs the parsing of the data and visualizes the data.

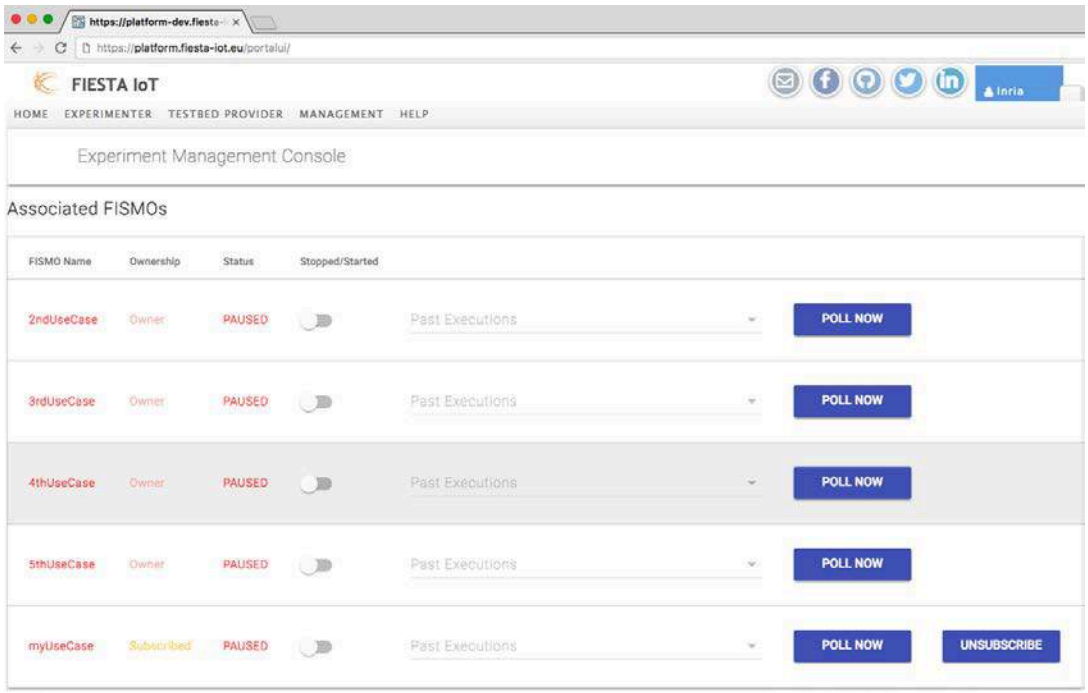

Figure 21 FISMO pane in the Experiment Management Console

Translating the above said workflow to the experiment architecture, there are various interactions among various components. These interactions are shown in the Figure 22. The sequential list of interactions is as follows:

1. Experimenter creates the experiment FED-Spec.
2. He authenticates himself.
3. An access-token is returned to the experimenter upon successful authentication.
4. The experimenter then sends the created FED-Spec to ERM using the provided User Interface (UI) along with the access-token.
5. He then opens the EMC.
6. The EMC requests the experiment details from the ERM.
7. Upon the response from ERM, EMC displays the content to the experimenter.
8. The experimenter then selects the experiment,
9. He enables the associated FISMOs to be executed on the FIESTA-IoT platform.
10. EMC then calls the EEE API to start the execution of the FISMO.

11. Upon the interval set in the scheduling object of the FISMO, the EEE queries IoT-Registry component to retrieve the desired data using the following query:

```
Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
Prefix iotlite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
Prefix dul: <http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
Prefix time: <http://www.w3.org/2006/time#>
Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
select ?sensorID (max(?ti) as ?time) ?value ?latitude ?longitude
where {
       ?o a ssn:Observation.
       ?o ssn:observedBy ?sensorID.
       ?o ssn:observedProperty ?qk.
       Values ?qk {m3-lite:Sound m3-lite:SoundPressureLevelAmbient}
       ?o ssn:observationSamplingTime ?t.
       ?o geo:location ?point.
       ?point geo:lat ?latitude.
       ?point geo:long ?longitude.
       ?t time:inXSDDateTime ?ti.
       ?o ssn:observationResult ?or.
       ?or  ssn:hasValue ?v.
       ?v dul:hasDataValue ?value.
       {
             select  (max(?dt)as ?ti) ?sensorID
             where {
                    ?o a ssn:Observation.
                    ?o ssn:observedBy ?sensorID.
                    ?o ssn:observedProperty ?qk.
                    Values ?qk {m3-lite:Sound m3-
             lite:SoundPressureLevelAmbient}
                    ?o ssn:observationSamplingTime ?t.
                    ?t time:inXSDDateTime ?dt.
             }group by (?sensorID)
        }
        FILTER (
             (xsd:double(?latitude) >= "-90"^^xsd:double)
             && (xsd:double(?latitude) <= "90"^^xsd:double)
             && ( xsd:double(?longitude) >= "-180"^^xsd:double)
             && ( xsd:double(?longitude) <= "180"^^xsd:double)
             )
        FILTER(?value>="75"^^xsd:double)
} group by ?sensorID ?time ?value ?latitude ?longitude
```

12. IoT-Registry executes the query and sends the response back to the EEE.
13. The EEE then forwards the response to the Experiment Data receiver component that executes on the experimenter side.

    13'. The EEE also notifies the EMC about the successful execution. The EMC upon the receipt of the response updates the UI related to the FISMO.

14. The Experimenter is presented with the updated UI (with most resent information about the FISMO) of the EMC
15. The Visualizer pull the information collected by Experiment Data Receiver and create the UI. Note that this UI is different from the EMC UI.

16. The Experimenter loads the Visualizations. In the current version only the most recent results are shown. Experimenter will have to refresh the UI to see the most recent information.

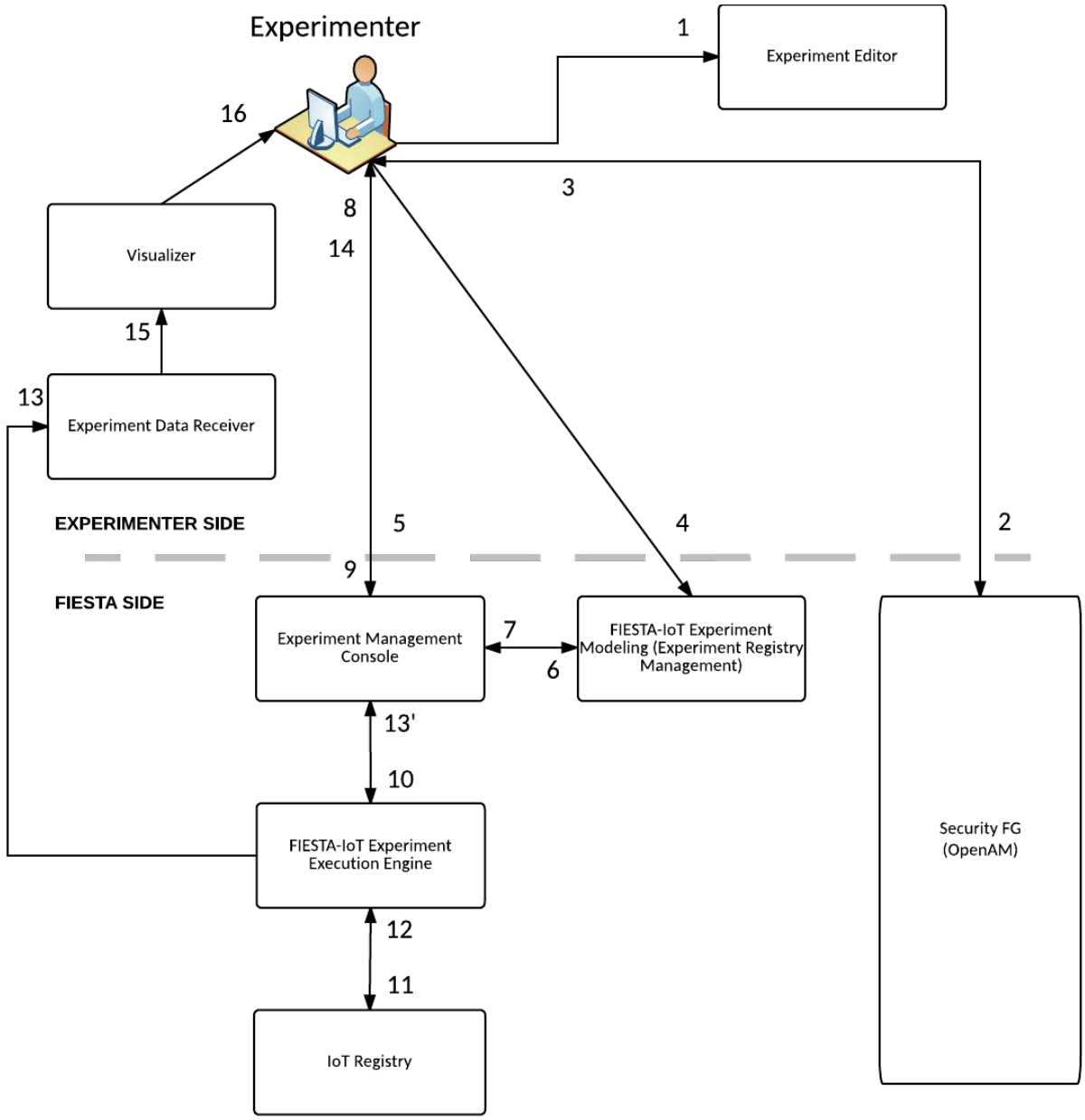

Figure 22 Large Scale Crowdsensing Experiment Architecture with interactions

2.2.6 FIESTA-IoT Ontology concepts used towards building the queries

Referring back to (FIESTA-IoT D5.1, 2016), the above use case needs following information: Sensor producing the sound level observations, location of the sensor, time the observation was taken, sound level values, and sound quantityKind. These information are realized in the ontology via ssn:sensor, geo:location, time:instant, ssn:ObservationValue and m3-lite:QuantityKind.

- *ssn:Sensor*: the sensor that observes a certain value;

- *geo:location:* that contain exact latitude and longitude at which the value has been observed;
- *time:instant:* that contains the timestamp of the observation;
- *ssn:observationValue:* the class that contains the observation value;
- *m3-lite:QuantityKind*: The phenomenon observed by the sensor.

We refer the readers to (FIESTA-IoT D3.1, 2016) and (FIESTA-IoT D3.2, 2016) for more knowledge about these concepts.

2.3.3 Outcomes

With our experiment[5] it is possible to know what are the most noisy areas within a specified region (although we consider the region as whole world for our current experiment). Using such information citizens could evade the noisy places and find more quite places to do their activities. As a part of this deliverable we are able to show above-mentioned aspect (see Figure 23).

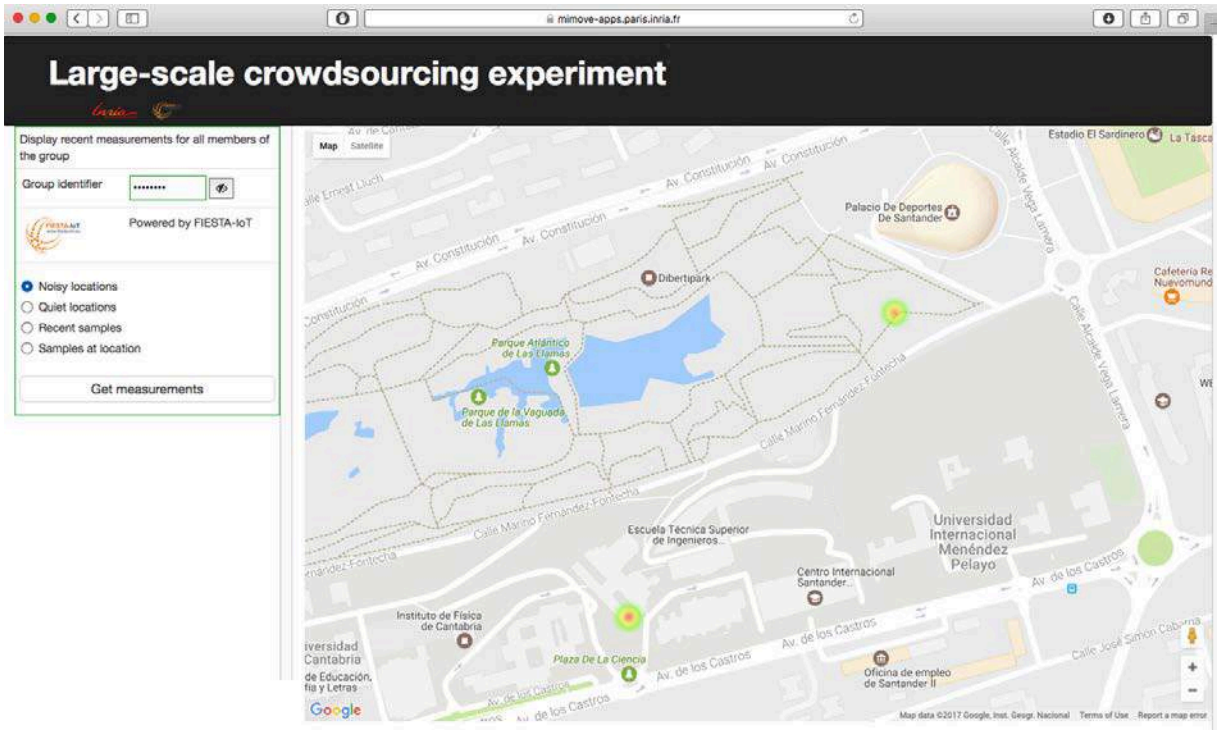

Figure 23: Noisy Locations heatmap

2.3.4 Future work

Currently, for this deliverable we have focused only on identifying most noisy locations. We however, intend to address all the use cases for the next version of the deliverable.

[5] Our Experiment is available at https://mimove-apps.paris.inria.fr/fiesta/index_fiesta.html

3 METHODOLOGIES FOR EXPERIMENT EVALUATION AND FIESTA-IOT VALIDATION

In this section, we present the different methodologies for evaluating the quality and the outcomes of the experiments, namely *experiment* evaluation, and for assessing the validation of the FIESTA-IoT concepts, platform and tools through the experimentation, namely *FIESTA-IoT validation*. For each of the two purposes, different aspects and point views have been identified:

- evaluation of experiment:
  - the evaluation of the achievements of the experiment objectives,
  - the evaluation of the process of design of experiment and the effective integration with the FIESTA-IoT tools,
  - the evaluation of the scientific value of the experiment outcomes;

- validation of FIESTA-IoT concepts, platform and tools:
  - the validation of the FIESTA-IoT concepts (like testbed agnostic access, portability of experiment) through the execution of the experiments,
  - the validation of the usability and convenience of FIESTA-IoT tools from the point of view of experimenters,
  - the validation of the accessibility of the FIESTA-IoT platform and tools through documentation and support from the point of view of experimenters.

The processes of evaluation and validation are composed of multiple steps performed at different stages of the experiment. We have distinguished two main macro-phases of the experiment life-cycle:

- *integration phase* which is comprehensive of:
  - *learning phase* which is the period during with the experimenter learns the capabilities of the FIESTA-IoT platform, the functions offered by each of the Functional Components (FCs) described in the architecture, and the utilities given by the FIESTA-IoT tools.
  - *design phase* is the period comprehensive of the definition of the architecture of the experiment and the actual use-case scenarios enacted by the experiment.
  - *development and integration phase* is the period spent for integrating the FIESTA-IoT tools (e.g. FEMO definition, ad-hoc connector with the FIESTA-IoT REST API) with already existent tools or other ad-hoc tools implemented for the sake of the experiment
- *execution phase* which is comprehensive of*:*
  - *run-time phase* of the experiments
  - *results interpretation phase* which takes place after the experiments finished or at intermediate stage when the experiment is still running and continuous outcomes are available.
  - *experiment maintenance phase* is meant to last right after the first implementation of the experiment in order to keep the experiment working with the updates of the FIESTA-IoT platforms and tools, to correct the behaviour if the results after the run-time phase are not the one expected, and to enhance the experiment with new feature.

All the described phases and sub-phases are iterative and connected each other with loop-back interaction. For that reason the evaluations of experiments and the validations of the FIESTA-IoT concepts, platform and tools, can also be considered an iterative process and the assessment might vary from this deliverable (which is an intermediate deliverable) to the final deliverable D5.3 due in M36.

The conception and implementation of the presented methodologies is an outcome of an iterative process of methodologies design and their application on the three in-house experiments. Therefore, the methodologies have been deeply analysed on both theoretical and practical aspects.

3.1 Evaluation of experiments

This part of the overall validation and evaluation methodology relates to the assessment of the experiment-specific milestones that each experimenter had before starting the experiment itself. In this sense, the evaluation subject is the experiment and the FIESTA-IoT platform is merely a tool for achieving the objectives that would lead to a successful evaluation.

The cross-reference of the two interaction phases (namely Integration and Execution phase as described in the introduction of the chapter) and the three evaluation subjects makes in total six points of evaluation of FIESTA-IoT experiments, which is summarized in the following table. The methodology of each evaluation point will be explained in the following subsections.

	Integration phase	**Execution phase**
Evaluate achievement of objectives	Identification of KPIs	Assessment of KPIs
Evaluate advance from SotA[6]	N/A	Objective assessment
Evaluate experiment integration	Subjective assessment	N/A

Table 1 Evaluation of experiment methodology

It is important to note that the experiment motivation, objectives and suitability themselves are not subject of the evaluation since this evaluation has been already carried out by external experts on a peer-review basis and the resulting selection of experiments is meant to be sound and pertinent. However, already at the proposal writing time, experimenters were asked to fill a questionnaire.

The questionnaire in Annex III Questionnaire of experiment evaluation from FIESTA-IoT point of viewis focused on getting a first glimpse of the feasibility of integrating the experiment and the potential feedback that the integration and execution of that experiment might provide:

- Feasibility. An experiment can only produce valuable outcomes if they are achievable.

[6] Note: The advance from SotA subject will not be evaluated at the integration phase because the results of the experiment which actually might support this advance should only be available after experiment execution. Similarly, experiment integration evaluation is not relevant at the execution phase where the interactions of experiment and the FIESTA-IoT is purely machine-to-machine.

- Feedback to the platform. It's a win-win that the experiment uses FIESTA-IoT resources and gives feedback to FIESTA-IoT to help the platform and the ecosystem to improve. FIESTA-IoT will privilege the experiments that are potentially capable of provide valuable feedback.

The questionnaire provided a score to each candidate experiments. This questionnaire did not produce any veto situation but was useful to have a reference for the experiments from high to low score.

The "Feedback" section of this questionnaire can be re-distributed to experimenters when they finish the experiments. The updated score will rely on the actual implementation and results of the experiment taking into account the additional insights that experimenters have learnt during the integration and execution of the experiment.

3.1.1 Evaluate achievement of experiment objectives

As part of the experiment definition, experimenter defines a set of objectives that the implementation of the experiment on top of the FIESTA-IoT Platform is aiming at. Objectives can be both specific to the experiment (e.g. Investigate the correlations between network topologies and associated data graphs in IoT-big network data environments.) as well as related to the FIESTA-IoT Platform (e.g. Include mechanisms in FIESTA-IoT to provide information on quality of data transmission and data outliers to data consumers).

In this respect, experimenters will define KPIs and measureable outcomes at the beginning of their experiments in order to make the assessment of the achievement of the objectives defined beforehand. Evaluation of this topic will be carried out over the successful completion and fulfilment of the previously identified KPIs.

3.1.2 Evaluate experiment advance over SotA

This evaluation topic refers to the advance to the State of the Art or the innovation that the experiment has achieved. In this sense, evaluation will be done through tangible impact KPIs that the experimenters have identified in terms of the research questions that could be answered with the execution of proposed experiment and the corresponding publications they can generate with these answers.

Additionally, since experiments selected through the FIESTA-IoT Open Calls can also focus on innovation rather than on research, analogous impact KPIs can be identified for them.

Although this is not a primary evaluation topic, it will be included within the experiment evaluation methodology as it will provide third-party assessment of the quality or innovative nature of the experimentation (i.e. peer-reviewed publications, market advantage, etc.).

3.1.3 Evaluate experiment integration and implementation

The final experiment evaluation topic refers to the steps and process followed by the experimenters during the integration phase of their experiment. In this sense, the methodology that will be followed to make the assessment of this point will be the specification of a checklist (Annex IV EvaluatION experiment integration and

implementation Checklist) that will be checked upon the completion of the integration phase for each experiment.

The aspects that have been identified relates to the best practices and support mechanisms that the FIESTA-IoT consortium has put in place. The attendance to these best practices is meant to ease the experimentation process and also to optimize the use of the FIESTA-IoT Platform resources.

3.2 Validation of FIESTA-IoT Platform and Tools

As discussed in the beginning of the section, the interactions between the experiments and the FIESTA-IoT platform can be grouped into two phases: integration phase and execution phase. Three validation subjects are considered regarding to the functionalities, services and support that FIESTA-IoT aims to provide to the experimenters: FIESTA-IoT concept, tools and resources (viz. support materials). More details will be described in the following subsections.

The cross-reference of the two interaction phases and the three validation subjects makes in total six points of validation of FIESTA-IoT platform and tools, which is summarized in the following table. The methodology of each validation point will be explained in the following subsections.

	Integration phase	**Execution phase**
Validate FIESTA-IoT concept	Identification of KPIs	Assessment of KPIs
Validate FIESTA-IoT tools	Subjective assessment	Objective assessment
Validate FIESTA-IoT resources	Subjective assessment	N/A

Table 2 Validation of the FIESTA-IoT platform methodology

Note: The FIESTA-IoT resources subject will not be evaluated at the execution phase because the supporting resources such as documentation, support service is not relevant at the execution phase where the interactions of experiment and the FIESTA-IoT is purely machine-to-machine.

3.2.1 Validate FIESTA-IoT concepts

The FIESTA-IoT concept mainly consists of a cloud-based platform that provides a unified method for experimenters to access to resources hosted on different testbeds through semantic technologies. Thus, two aspects need to be assessed by the experiments to validate the FIESTA-IoT concept:

1. Testbed-agnostic access to different resources, which means that the hosting testbed is irrelevant to the resource access method
2. Unique platform entry point, which means that all the resources of FIESTA-IoT platform are only accessible through the only entry point with a validated set of credentials (if requested)

FIESTA-IoT concept needs to be validated through M2M interactions between the experiments and the platform. However, during the integration phase where the interactions are H2M (human to machine), the assessment subjects and measurements methods should be identified:

- Integration phase: Identification of KPIs related to the FIESTA-IoT concept. Each experiment should define a set of KPIs at this stage to validate the

FIESTA-IoT concept during the next stage (experiment execution), for example, simutaneous data from 2 or more testbeds

- Execution phase: assessment of defined KPIs, either by a monitoring tool that retrieves relevant data value to validate the KPIs, or simply by manually checking the experiment result.

3.2.2 Validate FIESTA-IoT tools

FIESTA-IoT platform is delivered together with a set of tools for the experiment development, deployment and execution. The quality of the tools from the point of view of users, aka the experimenters, is also key to the quality of the FIESTA-IoT platform. Thus, it is indispensable to validate if the tools meet the expectations from the point of view of the users and provide the functionalities that the platform promises.

The validation of tools consists also of 2 phases:

- Integration phase. At the end of this phase, experimenters will be given a questionnaire with questions to evaluate the tools that served during their experiment development and deployment, including the easiness of learning the tools, the usefulness and the performance. The questionnaire will be shown in Annex V QuestionNaire: Validation of the FIESTA-IoT resources together with the questions from the following section.
- Execution phase. During the execution phase, monitoring tool of the FIESTA-IoT platform will continuously track the functions and performance of the tools which participate to the experiment execution, for example, if the API provides latest and historical observation of a sensor regarding to the request from an experiment, how much is the delay between the request and response, etc. The development of such monitoring tool is under the responsibility of experimenter.

3.2.3 Validate FIESTA-IoT resources

FIESTA-IoT resources refer to the support materials that the FIESTA-IoT platform made available for experiment development purpose. The completeness and clarity of these materials are essential for the experiment development efficiency. This aspect will be evaluated by the experiment developers by answering a specifically designed questionnaire at the end of the integration phase. This questionnaire, together with the one introduced in 3.2.2 will be presented in Annex V QuestionNaire: Validation of the FIESTA-IoT resources, and the result will help the FIESTA-IoT consortium to identify unsatisfactory part of the resources and to improve them in the future.

4 EVALUATION OF IN-HOUSE EXPERIMENTS AND VALIDATION OF FIESTA-IOT BY IN-HOUSE EXPERIMENTS

In this section we have applied the methodologies described in section 3 on the three in-house experiments which are the only available at the time of the creation of this document.

The evaluation of the achievements of the experiments has been performed as a self evaluation, based on the KPIs depicted in (FIESTA-IoT D5.1, 2016), by all the three in-house experiments owners, as exercise and example for the third-parties experiments, under the supervision of the other stakeholder involved in task T5.4. In a different manner was conducted the evaluation of the integration of the experiments integration and implementation where the questionnaire in Annex IV EvaluatION experiment integration and implementation Checklist, compiled within the task T5.4, has been answered by all the three in-house experiments owners, together with a description about the utilization of the FIESTA-IoT and what other tools have been necessary for the implementation of the experiments.

The validation of FIESTA-IoT has been conducted, by the three in-house experiments owners under the supervision of the other stakeholders of the task T5.4, on three different manners for three different targets: a validation of the FIESTA-IoT objectives, through the realization of the experiments; the validation of the FIESTA-IoT tools via the satisfaction of the KPIs defined by each experiments owner in (FIESTA-IoT D5.1, 2016); the validation of the FIESTA-IoT resources and documentation used for learning and understanding the projects and tools, through answering the questionnaire in Annex V QuestionNaire: Validation of the FIESTA-IoT resources.

4.1 Evaluation of experiments

4.1.1 Achievement of experiment KPIs evaluation

4.1.1.1 *Data Assembly and Services Portability Experiment*

KPI	Details	Status
Creation of more than 200 Virtual Entities	More than 250 Virtual Enties have been already created. For each of such Virtual Entities analytics functions are automatically instantiated and performed and augmented data created within the experiment.	Achieved
Data aggregated on more than 2 abstraction levels	The experiment is aggregating data over 4 different abstraction level (Building/Street, City, Region, Country).	Achieved
Have 1 or more indicators based on Observation-oriented analytics	The experiment is performing 2 anlaytics function based on observation: data statistics (average, minimum,	Achieved

	maximum) and sensor deployment quality (observation density per area, number of active sensors of a certain type per virtual entity).	
Have 1 or more indicators based on Resource-oriented analytics	Not yet matched at the current status of the Smart City Magnifier development. Since this deliverable refers to an intermediate status of the experiment with still the 3rd year to go, this typology of analytics is seen as future work for the last year of the FIESTA-IoT project.	Not Achieved
Leverage data from at least 3 testbeds	The experiment is acquiring and using data already from the SmartSantander testbed, the crowdsensing testbed SoundCity and the smart office testbed deployed within the KETI premises.	Achieved
Apply analytics functions on data coming from at least 2 testbeds	The dashboard, which is aggregating data based on the focused geographic scope, is able to show the situation of the entire European continent on the lateral gauge widget exploting data from the SmartSantander testbeds and the crowdsensing testbed of SoundCity.	Achieved

Table 3 Evaluation of the Data Assembly and Service Portability through KPIs

4.1.1.2 *Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access*

KPI	Details	Status
Leverage data from more than 1 testbed	At the time of writing this document, up to **four testbeds** do provide environmental information, i.e. SmartSantander, SmartICS, SoundCity and KETI	Achieved
Encapsulate, in a single response, the resource descriptions or from, at least, 4 different platforms	Through SPARQL requests, at time of this report, we could transparently receive data from 4 testbeds. As soon as more and more testbeds are registered, we will get data from more and more different sources of information.	Achieved
Filter resources upon location and phenomena based premises	The application provides the means to dynamically select a subset of resources, either basing our decision on the physical phenomenae measured by the	Achieved

	sensors or their current location	
Aggregate data in order to build higher level information	The application combines the raw measurements and yields more complex statistical data (e.g. average, variance, etc.).	Achieved
Have at least 50 different users running the experiment	This is a kind of off-topic performance indicator, meaning that other developers use this experiment as a guideline for theirs. With the publication of the experiment source code in GitHub[7] and the server running, we hope that external users take a look at it in order to get some acquaintance with the interaction with the FIESTA-IoT tools.	Partially achieved
Assess the performance of the FIESTA-IoT platform	One of the features included in the experiment focuses on the evaluation of the behaviour showed by the FIESTA-IoT platform. As long as the experiment is evolving, more and more measurements will be appended to this characterization assessment.	Achieved

Table 4 Evaluation of the Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access experiment through KPIs

4.1.1.3 *Large Scale Crowdsensing Experiments*

KPI	Details	Status
Leverage data from more than 1 testbed	The experiment in order to provide global perspective needs data from various types (mobile, static, participatory) of sensor available within the city environment. The experiment uses data made available from Soundcity, SmartSantander and SmartICS testbed.	Achieved
Large scale spatial Heatmap of noisy/quiet places	To build high quality heatmap, lot of data is needed. Currently, as within FIESTA-IoT there are not many sound sensors available, building large scale heatmap is something currently not available. We	Partially Achieved

[7] https://github.com/fiesta-iot/in-house-dynamic-discovery

	wish to have large scale quality heatmap in near future.	
City Independent nature of the experiment	Our experiment is city independent, this is achieved using a query that is not location dependent. Further, due to the nature of testbeds such as Soundcity that provides data from all over the world, we are able to get data from FIESTA-IoT which is global.	Achieved
Large scale view leveraging data from more than 100 sensors.	The experiment, in order to build large scale quality heatmap, needs data from lot of sensors. Currently, Soundcity Testbed provides 6 sound sensors. We envision seeing an increase in this number once more users join. Further, from SmartSantander there are currently 34 sensors from which the data is received while from smartICS approx 100 sound sensors provide data.	Achieved

Table 5 Evaluation of the Large Scale Crowdsensing experiments through KPIs

4.1.2 Experiment integration and implementation evaluation

4.1.2.1 *Data Assembly and Services Portability Experiment*

The questionnaire proposed for evaluating the integration and implementation phase has been answered by this experiment and put in the Annex IV EvaluatION experiment integration and implementation Checklist.

Furthermore, for the specific case of the Data Assembly and Services Portability Experiment, the implementation phase leveraged several components for achieving all the functionalities. The FIESTA-IoT tools have been used mainly for retrieving and interpreting the data. Instead, for implementing the backend analytics and context management, other tools have been used.

FIESTA-IoT tools

- *Resource Discovery*: used in order to discover resources within FIESTA by specifying certain parameters
- *FIESTA-IoT endpoints*: used for retrieving the latest observed value by the sensing devices.
- *Semantic Data Repository*: used for fetching the data of sensing devices deployed by testbeds which are not exposing IoT endpoints.
- *FIESTA ontology*: all the fetched data is interpreted with the annotation defined by the FIESTA ontology

- OpenAM: all the resource discoveries and historical data requests have been authenticated by a token acquired through the OpenAM server within FIESTA-IoT. A single set of credentials have been enough to access data from all the available testbeds.

Other tools

- *FIWARE IoT Broker GEri*[8]: used as Context Management. The configuration of the IoT Broker is the Standalone IoT Broker (IoT Broker + NEConfMan) with the Historical Agent feature enabled.

- *NGSI*[9]: as the data format and API used for the communications between: the backend components; the backend components and the frontend component.

- *OpenStreetMap*[10]: used as external resource for contextualizing the observations by their location

- *Freeboard*[11]: dashboard framework used to create the Dashboard component.

4.1.2.2 *Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access*

The questionnaire proposed for evaluating the integration and implementation phase has been answered by this experiment and put in the Annex IV EvaluatION experiment integration and implementation Checklist.

Furthermore, in a similar way to the former case, this experiment does make use of various key components that form the FIESTA-IoT platform core. We describe in the following list the main interaction with these elements.

FIESTA-IoT tools & components:

- Resource discovery: By means of an off-the-shelf SPARQL query, we can gather all the resources available (i.e. registered) at the FIESTA-IoT federation.
- FIESTA-IoT ontology: Playing the role of experimenters, we have to be aligned with the datasets generated by the platform. Due to the direct interplay between our application and FIESTA-IoT platform, we have to directly parse data that respect the rules imposed by this semantic model.
- IoT-Registry. The interaction with this component is essential to retrieve the resource descriptions (during the discovery phase) and the measurements that are being generated by the sensors throughout the time.

[8] https://catalogue.fiware.org/enablers/iot-broker

[9] https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_NGSI_Open_RESTful_API_Specification_%28PRELIMINARY%29

[10] http://wiki.openstreetmap.org/wiki/Nominatim

[11] https://freeboard.io/

- IoT-Service Endpoints. In order to get the information directly from the underlying testbeds, i.e. without having to seek into the semantic triplestore, some testbeds offer the possibility of retrieving data directly from a set of endpoints, through which they can provide information. Nonetheless, the interaction between the experiment and testbeds is always handled by the FIESTA-IoT platform, which “hides” the actual testbed operation to the experimenter(s). Specifically, we use these services to get the last observations measured by the sensors.
- OpenAM: As experiments come from outside FIESTA-IoT platform, every exchange of information has to be authenticated and authorized by this component. Following this protocol, we do obtain the necessary token to establish a secure channel between our server and e.g. FIESTA-IoT registry.

Other tools (external):

- Leaflet[12]: We use this library to display the resources on the map. Apart from the raw visualization of markers, it allows, for the sake of a better performance, the clustering of nodes, thus yielding a smoother performance.
- Turf[13]: To solve geometrical problems, e.g. nodes within a rectangle, polygon, circle, etc. we rely on this popular Javascript library, whose main value is the use of optimal algorithms to solve out this type of operations.
- D3[14]: Once we get the raw information from the FIESTA-IoT platform, we rely on this framework for the graphical (and more intuitive representation) of data.
- MongoDB: For the sake of storing historical data at the experimentation level, we leverage this database, we use a local database (allocated in the server) where we keep all the observations this server had been periodically polling to the FIESTA-IoT platform.

4.1.2.3 *Large Scale Crowdsensing Experiments*

The questionnaire proposed for evaluating the integration and implementation phase has been answered by this experiment and put in the Annex IV EvaluatION experiment integration and implementation Checklist.

Furthermore, this experiment uses FIESTA-IoT tools built to support Experimentation. The experiment uses following to execute the experiment, the “Experiment data receiver” created by FIESTA-IoT to enable experimenters receive data. The specific usage of the above-mentioned tools is explained in the Section 2.3.2:

- ERM: to store the FEDspec created.
- EEE: to execute the FISMOs in the FEDspec.
- EMC: to enable the execution of the needed FISMO.
- IoT-Registry (FIESTA-IoT Semantic Storage component): it is the component where the EEE periodically sends the query to be excuted for the results.
- Portal: is used to login to FIESTA-IoT and use ERM and EMC.

[12] http://leafletjs.com/

[13] http://turfjs.org/

[14] https://d3js.org/

- Security Component: to login to the portal and use the ERM and EMC, first necessary session cookie has to be generated. Security component is thus need to generate the session cookie (also known as access token) .

Other tools

On top of the tools provided by FIESTA-IoT, the experiment uses:

- A text editor to create the FED-Spec of the experiment. Once the “Experiment Editor” component is made available by FIESTA-IoT, we wish to use the “Experiment Editor” component to perform edits to the experiment FED-Spec.
- Tool to visualize the received result. We use node.js to build the user interface.

4.2 Validation of FIESTA-IoT concepts, platform and tools

As explained in 3.2, the validation of FIESTA-IoT concepts, platform and tools in the integration phase is mainly through the questionnaire which is presented in Annex V QuestionNaire: Validation of the FIESTA-IoT resources. In the execution phase, the validation is conducted by assessing the defined KPIs using subjective and objective methods.

In this section, we will present the validation of the three aspects and the conclusion from the validation questionnaire respectively in a subsection.

4.2.1 Validation of the FIESTA-IoT concepts

4.2.1.1 *Data Assembly and Services Portability Experiment*

The current implementation state of this experiment has accomplished already many FIESTA-IoT objectives:

Objective	Details	Status
Design and implement integrated IoT experiments/applications	We leverage the possibility of the FIESTA-IoT platform for creating a Smart City IoT application usable at the same time on multiple IoT deployments, all the ones integrated.	Matched
Testbed Agnostic Access to IoT Datasets	The discovery of resources and the retrieval performed by the Smart City Magnifier is completely unaware of the difference among the original IoT deployment.	Matched
Tools and Techniques for IoT Testbeds Interoperability and Portability	The Smart City Magnifier application runs smoothly over different zones of the globe, therefore leveraging data from different testbeds. In order to verify this statement the focus of the map in the dashboard was moved to different geographic areas overlapping different testbed deployments with the results that the indicators are still computed and displayed. Furthermore the datasets acquired are then seamlessly used as input of the analytics.	Matched
Proof-of-Concept Integrated Experiments	As a Proof-of-Concept for the data assembly and service portability experiment we have implemented the Smart City Magnifier thereby validating this objective.	Matched
Best Practices	The design and implementation of this experiment has brought to the definition of good practice for implementing large scale IoT experiments such as the lacking of subscription-notificaiton message bus and performance issues of historical query.	Matched

Table 6 Validation of FIESTA-IoT concepts by the Data Assembly and Service Portability experiment

4.2.1.2 *Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access*

This experiment was designed in order to accomplish various objectives that were originally defined as a list of main challenges to be tackled under the scope of the FIESTA-IoT project. Even though not all of them have to do with this application, it is true that we do cover some of them, summarized below.

Objective	Details	Status
Design and implement integrated IoT experiments/applications	We harness the possibilities offered by the FIESTA-IoT platform to place the experiment on top of it, thus establishing a single (and protected) communication point between them.	Matched
Testbed Agnostic Access to IoT Datasets	In the eyes of the application, all resources belong to FIESTA, regardless their actual owner. Indeed, there is no clue of the devices' ownership in any part of the experiment, as unique FIESTA-IoT identifiers are put on top of the legacy ones	Matched
Tools and Techniques for IoT Testbeds Interoperability and Portability	The IoT-Registry API provides a fully-fledged set of services that permits the transparent extraction of data from any of the underlying testbeds. Hence, the EaaS infrastructure satisfies the portability that bring about the obtention of data from different and heterogeneous testbeds.	Matched
Proof-of-Concept Integrated Experiments	The own nature of these in-house experiments comes to justify this objective.	Matched
Best practices	The experiment itself has followed a twofold path in order to fulfil the quality achievements pursued in this project.	Matched

Table 7 Validation of FIESTA-IoT concepts by Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access experiment

4.2.1.3 *Large Scale Crowdsensing Experiments*

This implementation of the experiments has validated the following objectives of the FIESTA-IoT project:

Objective	Details	Status
Design and implement integrated IoT experiments/applications	We are able to use the FIESTA-IoT Platform to design, implement and execute our experiment using various tools made available by FIESTA-IoT. We further used "single entry point and based on a single set of credentials" to perform our experimentation. This validates Objective 1	Matched

Testbed Agnostic Access to IoT Datasets	As the Experiment Execution Engine executes the queries on the Meta-Cloud infrastructure that provides Testbed agnostic access to IoT datasets, Objective 2 is also validated by the use and successful execution of the experiment	Matched
Tools and Techniques for IoT Testbeds Interoperability and Portability	As FIESTA-IoT tools (Portal, security, EEE, EMC, Meta-Cloud repository (internally by EEE)) are used to perform the experiment, a successful execution of the experiment validates Objective 3	Matched
Proof-of-Concept Integrated Experiments	As a proof-of-Concept for the crowdsensing Experiment we have implemented the large-scale crowdsensing experiment thereby validating Objective 5.	Matched

Table 8 Validation of FIESTA-IoT concepts by Large Scale Crowdsensing experiments

4.2.2 Validation of the FIESTA-IoT tools through KPIs

4.2.2.1 *Data Assembly and Services Portability Experiment*

We have created a list of KPIs (FIESTA-IoT D5.1, 2016), related to our experiment, for assessing the validation of FIESTA-IoT tools. The following table contains those KPIs with the status with the current situation of the platform and the experiment implementation.

KPI	Details	Status
1 or more measurements are to be notified to the data analytics algorithm after a data subscription.	Subscription system not yet implemented within FIESTA-IoT. As soon as the messages bus tool will be ready this point will be achieved and validated.	Not Achieved
Observations streams, due to a query or a subscription, from 2 or more testbeds.	The backend components get observations from the crowdsensing testbed SoundCity and the KETI smart office testbed with a single historical SPARQL query.	Achieved
A data analytics algorithm receives 1 or more value(s) computed by another data analytics algorithm.	At the moment no experiments is pushing back their results within FIESTA-IoT. With the integration of third-parties experiments this point will be most likely achieved and validated.	Not Achieved
2 or more measurements, observed from the same device at different time, are returned in the historical query response.	The backend of the Semantic Mediation Gateway is getting the full timeseries of the historical repository for each of the sensors producing observation within the KETI smart office.	Achieved
The data analytics algorithm is receiving 2 or more data messages after the authentication	The data requests within a certain period (in terms of day) are using the same token for authorization purpose.	Achieved

Table 9 Validation of FIESTA-IoT platform by the Data Assembly and Service Portability experiment through KPIs

We have also experienced the integration of different tools offered by FIESTA-IoT for easing the process of the experiment implementation:

Tools	Details	Status
OpenAM	We have easily implemented the usage of the API offered of the OpenAM server deployed by FIESTA-IoT for accessing the FIESTA-IoT functionalities	Validated
IoT-Registry	Our experiment can already easily get the list of the wanted IoT resources within one single request.	Validated
IoT service endpoints	The connector of the Smart City Magnifier is able to retrieve the last observation from sensors directly by making a request to the IoT service endpoints transparently proxied by FIESTA-IoT.	Validated
Semantic Data Repository	Our experiment is able already to request historical data, within a geographic scope, with a single request	Validated
Message Bus	The subscription-notify system has not being validated since the Message Bus component is not yet ready. The implementation of this component is anyway on the roadmap.	Not validated

Table 10 Validation of FIESTA-IoT tools by the Data Assembly and Service Portability experiment

Problems encountered:

The first problem encountered during the implementation of the experiment is the lacking of the asynchronous notification system due to a prioritization of the effort on finalizing the core of the FIESTA-IoT platform. This problem has not blocked the development of the experiment which implements query polling instead. In future an asynchronous notification system would nevertheless be adopted and bring advantages like better performances.

A second problem encountered was the performance of the Semantic Data Repository. Data SPARQL queries were replied with a too high response time or even, in some extreme cases, ended with a connection timeout. This issue was solved by optimizing the SPARQL and reformulating it by changing the order of the clause.

4.2.2.2 *Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access*

We have given rise to the following list to assess the FIESTA-IoT platform, where we concentrate the most relevant achievements to be accomplished:

KPI	Details	Status
Display information of a minimum of 5,000 resources, coming from	The achievement of this indicator depends on the platform and the integrated testbeds. At the time of	Partially achieved

at least four different data sources	writing, we discover approximately 1,000 resources, but we believe that, by the end of the evaluation process, the platform will have more than these goal of 5000.	
Encapsulate, in a single response, the resource descriptions from,at least, 4 different platforms	Through SPARQL requests, at time of this report, we could transparently receive data 4 testbeds. As soon as more and more testbeds are registered, we will get data from more and more different sources of information.	Achieved
Get data from the invocation of the corresponding IoT Service endpoints	Instead of directly querying to the FIESTA-IoT triplestore, the platform offers the possibility of "bypassing" the central repository and indirectly access testbeds' databases so that we can get the latest measurements captured by the different sensors.	Achieved
Get historical data through explicit SPARQLs	By means of the IoT-Registry API, we directly query for historical data, allowing us to perform statistical analysis.	Achieved
Use the asynchronous service to subscribe to future events	When it comes to refer to future events, the most elegant solution is to asynchronously receive the data in a seamless way. Unfortunately, the service is not ready yet. Of course, we do believe that this will be ready soon and we will integrate this service in the next iteration of the experiment.	Not achieved

Table 11 Validation of FIESTA-IoT platform by the Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access experiment through KPIs

Technically speaking, at the very same time we have tested our own application, we have assessed the intrinsic components that shape the FIESTA-IoT core platform. Below we summarize those ones that have been touched:

Tools	Details	Status
OpenAM	We have stuck to the security means proposed by this framework.	Validated
IoT-Registry	The IoT-registry was successfully able to execute the query and provide result in desired format in desired time.	Validated
IoT service endpoints (linked to IoT-Registry resource broker* (FIESTA-IoT D4.2, 2017)*)	Even being an optional part of the resource description, we have assessed that the FIESTA-IoT platform is able to seamlessly proxy between the experiment and the testbeds without hinting any information about the real devices' identification.	Validated
Message Bus	As soon as the asynchronous system is settled down, FIESTA-IoT platform (and thus, this experiment) will fully support subscription-like mechanisms.	Not validated (component not ready)

Table 12 Validation of FIESTA-IoT tools by the Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access experiment

Problems encountered

One of the features that has not been ready by the initial phase of this experiment is the *asynchronous service*, where we could have harnessed the potential of a fully-fledged subscription system to the observations. As soon as it is available, we will integrate this mechanism into the experiment, thus replacing the legacy polling service.

At the moment of writing this document, the primary way to retrieve data from relying on the utilization of IoT service endpoints instead of the storage of observations in the own platform. Therefore, we have had to rely on the invocation of these endpoints, found in the resource descriptions, instead of directly querying to the meta-cloud repository. Alike, this is a temporal solution that will be replaced as soon as testbed gradually store their data in the FIESTA-IoT platform.

4.2.2.3 *Large Scale Crowdsensing Experiments*

The KPI defined help us validate FIESTA-IoT tools. We list these KPIs below.

KPI	Details	Status
Leaverage data from as many testbeds as possible (to quantify: more than 1 testbed)	Currently, based on the experiment requirement of observations from sound sensors, the experiment gets data from SmartSantander and Soundcity testbed. Further, as the query used is not testbed agnostic, in furture, if more testbed associated to FIESTA-IoT provide sound sensor related observations, the experiment will get data from them as	Achieved

	well.	
Number of sensors that provide data to the experiment (to quantify: more than 100 sensors)	As SoundCity Testbed is crowdsensing Testbed and the integration with FIESTA-IoT was done recently (month 24), there are less than 4 users that have authenticated Soundcity Testbed to send their information to FIESTA-IoT. We envision seeing an increase in this number once more and more users join. Further, from SmartSantander there are currently 34 sensors and more than 100 sound sensors from Smart ICS from which the data is received.	Achieved
Large Number of samples needed for high quality results	Currently, as the triple store is young not lot of observations are available. This also partially depends on previous KPI. As soon as the triple store grows wrt to number of observations related to sound, this KPI will be achieved.	Not Achieved

Table 13 Validation of FIESTA-IoT platform by the Large Scale Crowdsensing Experiments experiments through KPIs

Further, based on our experience, we also list if used FIESTA-IoT tools were easy to use, integrate, provide needed functionality with respect to our experiment:

Tools	Details	Status
Portal	We were able to successfully login to the portal and use the different tools. This also validates the Security Component (OpenAM).	Validated
ERM	We were able to successfully able to register our experiment using ERM	Validated
EMC	We were able to successfully schedule the experiment using the EMC and use provided functionality.	Validated
EEE	The experiment services were successfully scheduled and executed.	Validated
IoT-Registry	The IoT-registry was successfully able to execute the query and provide result in desired format in desired time.	Validated
Experiment Data receiver	The experiment data receiver was successfully able to receive data sent by EEE.	Validated

Table 14 Validation of FIESTA-IoT tools by the Large Scale Crowdsensing Experiments experiments

Problems Encountered

There are a number of issues currently faced by us while the preparing for our experiment. These issues are:

- Missing/incorrect triples: some Testbeds missed essential concepts made available via ontology. Thus, effort had to be put in to identify issues using SPARQL queries and were reported to the Testbed owners which then modified their annotators to correctly match the ontology. Further, as this was reported to the Testbed owners, they gave priority to solve this issue.
- Unavailability of a large number of sound sensors: Currently there are few sound sensors available within FIESTA-IoT platform and due to this the results quality is not very high.

Besides, above-mentioned list, we also, would like to state that we also faced some of the issues that were stated before (by the two other experiments).

4.2.3 Validation of the FIESTA-IoT resources

4.2.3.1 *Data Assembly and Services Portability Experiment*

Resources	Details	Status
Documentation on various tools, Training session and related material.	The abundant and well structured material about the API, the authentication procedures, the ontology and the example provided of SPARQL have been very helpful for the realization of the Smart City Magnifier prototype.	Validated
Issue tracker and email support	Direct communication with the responsible of the components, or, more generatically, with the FIESTA-IoT support, has hugely helped and solved all the issues in very short time.	Validated

Table 15 Validation of the FIESTA-IoT resources by the Data Assembly and Services Portability experiment

4.2.3.2 *Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access*

Resources	Details	Status
Documentation on various tools, Training session and related material.	Mainly, the guidance included in the handbook and the IoT-registry and OpenAM API documentation sites have been enough to clear out all the different issues that have been looming during the implementation phase. Moreover, the development of this experiment has been used as feedback to improve and complete the documentation itself.	Validated
Issue tracker and email support	Internal communication has been essential to fix all the issues that have arisen during the implementation of the experiment. The feedback achieved in this "private" support has been leveraged to improve the service offered to externals.	Validated

Table 16 Validation of the FIESTA-IoT resources by the Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access experiment

4.2.3.3 *Large Scale Crowdsensing Experiments*

Below we validate the various resources provided by FIESTA-IoT project:

Resources	Details	Status
Documentation on various tools, Training session and related material.	The high quality documentation enabled us to clearly integrate and understand the workflow for experimenters. The documentation helped us clearly create queries and FEDSpec using the best practices specified by the FIESTA-IoT Platform. The training material specially: workshops, presentations and handbook on top of documentation on various tools helped us successfully build our experiment.	Validated
Issue tracker and email support	The support provided was quality support. The experiment specific issues were clear delt with spontaneously.	Validated

Table 17 Validation of the FIESTA-IoT resources by the Large Scale Crowdsensing experiments

4.2.4 Conclusion from the validation questionnaire

All in-house experimenters have filled the validation questionnaire and their answers are available in Annex V QuestionNaire: Validation of the FIESTA-IoT resources. This questionnaire assesses the quality of experience of FIESTA-IoT experimenters. From their answers, we can get the following conclusion:

- Documentation of FIESTA-IoT is consulted often and appreciated by the experimenters that it provides rich, comprehensive and useful information for experiment development. Experimenters declared that they always found needed information in the documentation, and the quality of the documentation is satisfactory.
- For setting up and deploying an experiment, the processes are easy to follow and implement, and the integration and deployment on the FIESTA-IoT platform is relatively straight forward without much complication. The time spent to fully integrate the experiment with FIESTA-IoT is, in general, not more than 2 weeks.
- The FIESTA-IoT APIs are simple and useful. 2 experimenters over 3 declared that they preferred the API-based solution for interaction with the platform rather than using the experiment portal for the flexibility. However, the experiment portal stays the favorite of 1 in-house experimenter.
- The experiment results have met the experimenters' expectation according to their answers.
- All the experimenters declared to have an excellent interaction with the FIESTA-IoT team, and would recommend FIESTA-IoT platform to other experimenters.

From their answers, we can also identify some aspects that FIESTA-IoT need to improve in the future:

- The only documentation that the experimenters declared lower rate is the one about the SPARQL query. This subject can be enhanced in the handbook.
- The performance and availability of the platform is not totally satisfactory.

5 CONCLUSIONS

This deliverable has described the implementations and outcomes of the three in-house experiments. The report contained in this document is twofold for each experiment: report of the actual experiment architecture and implementation, and the interaction with FIESTA-IoT platform and the concept exploited of the FIESTA-IoT project.

It is worth to notice that even if all the experiments are at their first versions, many achievements and outcomes have been reached on both perspective: experimentation and FIESTA-IoT platform validation.

From the experiments perspective all of the three are effectively working and operative and ready to use the FIESTA-IoT platform as testbeds interoperability platform. All of them have implemented the first version of both the backend system, for getting and analyze data, and the frontend components for showing the results. Furthermore all the three in-house experiments have been evaluated with a methodology in order to have the most objective view of the results.

From the platform validation perspective, the experiments have been able to access data from different testbeds in an agnostic manner. Furthermore the experiments have shown their portability among testbeds, since, for all the three in-house experiments, the applications can be used in every region of the globe without the necessity to re-configure it and seamlessly exploiting data from very different IoT systems. In addition the experiments have been capable of retrieving more than one observations from the same sensor, with a single query, hence showing the capability of historical query. One of the experiments has also successfully leveraged the FIESTA-IoT tools for design the experiment and running it directly into the FIESTA-IoT platform and harvesting the results asynchronously. Finally all the three experiments have successfully integrated and used the security functions used to ensure the access control to data. All the interactions with the FIESTA-IoT platform have been preceded by only a single authentication request, used to retrieve the necessary token. Many FIESTA-IoT concepts have been leveraged into the three in-house experiment: the usage of the FIESTA-IoT ontology for understanding data coming from different IoT deployment in a seamless manner and the automatic execution of backend analytics (e.g. statical data aggregation); dynamic discovery of the resources regardless of testbed deployments; the Virtual Entities concept for adding abstraction layer to the data starting from the pure observations; the automatic execution of the experiment and the asynchronous harvesting of the results.
Also for the validation of the FIESTA-IoT platform, concepts and tools has been address with a well defined methodology and applied on all the experiments.

The outcome of the this document is bringing at the same time: hints to third-parties experimenters and FIESTA-IoT platform users on how to use the powerful FIESTA-IoT tools and testbeds interoperability for IoT applications; feedback to the FIESTA-IoT project on which aspect is to be considered weakness points to be enhanced in the future.

The work executed till now brought to attention some weakness of the FIESTA-IoT platform like: the triple store performance which can be a big bottle-neck if not wisely handled in the future; the scarcity of the data which can be easily overcome with the integration of FIESTA-IoT extensions from Open Calls; the necessity of an

asynchronous notification system for data, with the aim of lowering the bandwidth used, which is already in the roadmap of the FIESTA-IoT platform for the third year of the project.

6 REFERENCES

FIESTA-IoT D2.1. (2015). *FIESTA-IoT Project Deliverable D2.1 - Stakeholders requirements.*

FIESTA-IoT D2.4. (2015). *FIESTA-IoT Project Deliverable D2.4 - FIESTA Meta-Cloud Architecture and Technical Specifications.*

FIESTA-IoT D3.1. (2016). *FIESTA-IoT Project Deliverable D3.1 - Semantic models for testbeds, interoperability and mobility support and best practices.*

FIESTA-IoT D3.2. (2016). *FIESTA-IoT Project Deliverable D3.2 - Semantic Models for Testbeds, Interoperability and Mobility Support, and Best Practices.*

FIESTA-IoT D4.1. (2015). *FIESTA-IoT Project Deliverable D4.1 - EaaS Model Specification and Implementation.*

FIESTA-IoT D4.2. (2017). *FIESTA-IoT Project Deliverable D4.2 - EaaS Model Specification and Implementation V2.*

FIESTA-IoT D4.5. (2016). *FIESTA-IoT Project Deliverable D4.5 - Tools and Techniques for Managing Interoperable Data sets.*

FIESTA-IoT D5.1. (2016). *FIESTA-IoT Project Deliverable D5.1 - Experiment Design and Specification.*

ANNEX I FIESTA-IOT HOSTING INFRASTRUCTURE

Com4Innov owns a powerful data center with big capacity and resources in terms of CPU, RAM and disk storage. For this reason, as well, it plays an important role in the FIESTA-IoT project. We have agreed within Consortium to host the web services that are needed for the project, the tools that are developed and in general the FIESTA-IoT Metacloud architecture and the data provided by the testbeds that are participating in the project.

As far as the data center infrastructure of Com4Innov is concerned we show here under the main elements sizing the performance :

- ***3 Controller, Telemetry – MongoDB (16cpu, 32Go Ram, 500Go HDD)***
  - *Nova API*
  - *Neutron*
  - *Glance API*
  - *Cinder API*
  - *Pacemaker, Corosync*
  - *Swift Proxy*
  - *Ceilometer*
- ***8 Compute, Storage – Cinder (72cpu, 128Go Ram, 2To HDD)***
  - Libvirt, KVM
  - Neutron OpenVSwitch Agent
  - Cinder Volume
  - ISCSI, TGT

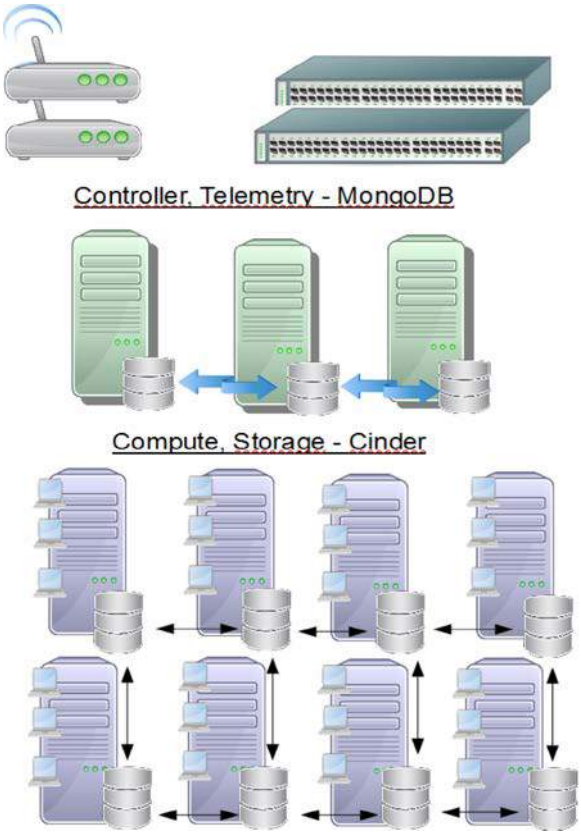

Figure 24 Working Elements

The different working elements are shown in Figure 24.

Installation and implementation

First, we assigned three Virtual Machines that are hosted in Com4Innov data center. These three VMs are of different types, each one serves different scope in the project development.

The first VM is called Utilities and contains the Gitlab, the Moodle and the Ontology platform. The second is the VM Testing and serves testing pre-production purposes of the tools that are being developed. Finally, the third one is the Production VM in which will be hosted the final version of the platform and the Metacloud Architecture, services and data. In , please find a table containing all the available information for the VMs, their capacity and the services running on them.

Virtual Machines	vCPU	RAM	Disk Drive (GB)	Context/Comments	OS
Utilities	vCPU2	4GB	40	Gitlab, Moodle, Ticketing system	Ubuntu server 14.04
Testing	vCPU8	16GB	160	Testing machine before production (tools, platform, some of data)	Ubuntu server 14.04
Procution	vCPU12	32GB	1615 (can be changed dynamically) <= 2000	Deployment of all the tools, platform, data, etc. of the whole project	Ubuntu server 14.04

Table 18 Virtual Machines

As a result, the Com4Innov administrator is responsible for giving access and privileges to the users accessing the VMs in order to develop new modules. The web services that are installed in the VM Utilities are covering the Gitlab, the Moodle and the Ontology.

ANNEX II FED-SPEC FOR LARGE-SCALE EXPERIMENT DEPICTING ALL USE CASES

```
<?xml version="1.0" encoding="UTF-8"?>
<fed:FEDSpec xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:fed="http://www.fiesta-iot.eu/fedspec"
 xmlns:prt="http://www.w3.org/2007/SPARQL/protocol-types#"
 xmlns:vbr="http://www.w3.org/2007/SPARQL/results#"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.fiesta-iot.eu/fedspec
file:/C:/Ext_SSD/AIT/FIESTA/FIESTA-SVN/WP4/Task%204.1/Objects/XSD/FEDSpec.xsd"
userID="USER">
    <fed:FEMO name="Experiment">
        <fed:description>LargeScale crowdsensing experiment</fed:description>
        <fed:domainOfInterest>http://purl.org/iot/vocab/m3-lite#Transportation
http://purl.org/iot/vocab/m3-lite#Pollution http://purl.org/iot/vocab/m3-lite#City
http://purl.org/iot/vocab/m3-lite#Health</fed:domainOfInterest>
        <fed:FISMO name="2ndUseCase">
            <fed:description>Over time all noise observations for a given
location</fed:description>
            <fed:discoverable>true</fed:discoverable>
            <fed:experimentControl>
                <fed:scheduling>
                    <fed:startTime>2016-11-08T18:50:00.0Z</fed:startTime>
                    <fed:Periodicity>250</fed:Periodicity>
                    <fed:stopTime>2017-11-08T18:49:59.0Z</fed:stopTime>
                </fed:scheduling>
            </fed:experimentControl>
            <fed:experimentOutput
location="https://experimentserver.org/store/"></fed:experimentOutput>
            <fed:queryControl>
                <prt:query-request>
                    <query><![CDATA[
                        # [1 / 1] visualization type: 'Gauge' and sensors
                        Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
                        Prefix iotlite: <http://purl.oclc.org/NET/UNIS/fiware/iot-
lite#>
                        Prefix dul:
<http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
                        Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
                        Prefix time: <http://www.w3.org/2006/time#>
                        Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
                        Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
                        select ?sensorID ?time ?value ?latitude ?longitude
                        where {
                            ?o a ssn:Observation.
                            ?o ssn:observedBy ?s.
                            ?o ssn:observedProperty ?qk.
                            Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                            ?o ssn:observationSamplingTime ?t.
                            ?o geo:location ?point.
                            ?point geo:lat ?latitude .
                            ?point geo:long ?longitude.
                            ?t time:inXSDDateTime ?ti.
                            ?o ssn:observationResult ?or.
                            ?or  ssn:hasValue ?v.
                            ?v dul:hasDataValue ?value.
```



```
                            FILTER (
                             (xsd:double(?latitude) >= "4.34"^^xsd:double)
                           && ( xsd:double(?longitude) >= "3.806"^^xsd:double)
                            )
                        } group by ?sensorID ?time ?value ?latitude ?longitude
                    ]]></query>
                </prt:query-request>
             </fed:queryControl>
        </fed:FISMO>
        <fed:FISMO name="3rdUseCase">
            <fed:description>Over time noise observations for a given bounding box
(time period in scheduling)</fed:description>
            <fed:discoverable>true</fed:discoverable>
            <fed:experimentControl>
                <fed:scheduling>
                    <fed:startTime>2016-11-08T18:50:00.0Z</fed:startTime>
                    <fed:Periodicity>250</fed:Periodicity>
                    <fed:stopTime>2017-11-08T18:49:59.0Z</fed:stopTime>
                </fed:scheduling>
            </fed:experimentControl>
            <fed:experimentOutput location="="https://experimentserver.org/store
"></fed:experimentOutput>
            <fed:queryControl>
                <prt:query-request>
                    <query><![CDATA[
                        # [1 / 1] visualization type: 'Gauge' and sensors
                        Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
                        Prefix iotlite: <http://purl.oclc.org/NET/UNIS/fiware/iot-
lite#>
                        Prefix dul:
<http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
                        Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
                        Prefix time: <http://www.w3.org/2006/time#>
                        Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
                        Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
                        select ?sensorID (max(?ti)
as ?time) ?value ?latitude ?longitude
                        where {
                            ?o a ssn:Observation.
                            ?o ssn:observedBy ?sensorID.
                            ?o ssn:observedProperty ?qk.
                            Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                            ?o ssn:observationSamplingTime ?t.
                            ?o geo:location ?point.
                            ?point geo:lat ?latitude.
                            ?point geo:long ?longitude.
                            ?t time:inXSDDateTime ?ti.
                            ?o ssn:observationResult ?or.
                            ?or  ssn:hasValue ?v.
                            ?v dul:hasDataValue ?value.
                            {
                                select  (max(?dt)as ?ti) ?sensorID
                                where {
                                    ?o a ssn:Observation.
                                    ?o ssn:observedBy ?sensorID.
                                    ?o ssn:observedProperty ?qk.
                                    Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
```



```
                                    ?o ssn:observationSamplingTime ?t.
                                    ?t time:inXSDDateTime ?dt.
                                }group by (?sensorID)
                            }
                            FILTER (
                               (xsd:double(?latitude) >= "-90"^^xsd:double)
                            && (xsd:double(?latitude) <= "90"^^xsd:double)
                            && ( xsd:double(?longitude) >= "-180"^^xsd:double)
                            && ( xsd:double(?longitude) <= "180"^^xsd:double)
                            )
                        } group by ?sensorID ?time ?value ?latitude ?longitude
                    ]]></query>
                </prt:query-request>
             </fed:queryControl>
        </fed:FISMO>
        <fed:FISMO name="4thUseCase">
            <fed:description>3rd usecase with noise more than x
dB(A)</fed:description>
            <fed:discoverable>true</fed:discoverable>
            <fed:experimentControl>
                <fed:scheduling>
                    <fed:startTime>2016-11-08T18:50:00.0Z</fed:startTime>
                    <fed:Periodicity>250</fed:Periodicity>
                    <fed:stopTime>2017-11-08T18:49:59.0Z</fed:stopTime>
                </fed:scheduling>
            </fed:experimentControl>
            <fed:experimentOutput location="="https://experimentserver.org/store
/"></fed:experimentOutput>
            <fed:queryControl>
                <prt:query-request>
                    <query><![CDATA[
                        # [1 / 1] visualization type: 'Gauge' and sensors
                        Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
                        Prefix iotlite: <http://purl.oclc.org/NET/UNIS/fiware/iot-
lite#>
                        Prefix dul:
<http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
                        Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
                        Prefix time: <http://www.w3.org/2006/time#>
                        Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
                        Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
                        select ?sensorID (max(?ti)
as ?time) ?value ?latitude ?longitude
                        where {
                            ?o a ssn:Observation.
                            ?o ssn:observedBy ?sensorID.
                            ?o ssn:observedProperty ?qk.
                            Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                            ?o ssn:observationSamplingTime ?t.
                            ?o geo:location ?point.
                            ?point geo:lat ?latitude.
                            ?point geo:long ?longitude.
                            ?t time:inXSDDateTime ?ti.
                            ?o ssn:observationResult ?or.
                            ?or  ssn:hasValue ?v.
                            ?v dul:hasDataValue ?value.
                            {
                                select  (max(?dt)as ?ti) ?sensorID
```



```
                                where {
                                    ?o a ssn:Observation.
                                    ?o ssn:observedBy ?sensorID.
                                    ?o ssn:observedProperty ?qk.
                                    Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                                    ?o ssn:observationSamplingTime ?t.
                                    ?t time:inXSDDateTime ?dt.
                                }group by (?sensorID)
                            }
                            FILTER (
                               (xsd:double(?latitude) >= "-90"^^xsd:double)
                            && (xsd:double(?latitude) <= "90"^^xsd:double)
                            && ( xsd:double(?longitude) >= "-180"^^xsd:double)
                            && ( xsd:double(?longitude) <= "180"^^xsd:double)
                            )
                            FILTER(?value>="75"^^xsd:double)
                        } group by ?sensorID ?time ?value ?latitude ?longitude
                    ]]></query>
                </prt:query-request>
             </fed:queryControl>
        </fed:FISMO>
        <fed:FISMO name="5thUseCase">
            <fed:description>3rd usecase with noise less than x
dB(A)</fed:description>
            <fed:discoverable>true</fed:discoverable>
            <fed:experimentControl>
                <fed:scheduling>
                    <fed:startTime>2016-11-08T18:50:00.0Z</fed:startTime>
                    <fed:Periodicity>250</fed:Periodicity>
                    <fed:stopTime>2017-11-08T18:49:59.0Z</fed:stopTime>
                </fed:scheduling>
            </fed:experimentControl>
            <fed:experimentOutput location="="https://experimentserver.org/store
/"></fed:experimentOutput>
            <fed:queryControl>
                <prt:query-request>
                    <query><![CDATA[
                        # [1 / 1] visualization type: 'Gauge' and sensors
                        Prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
                        Prefix iotlite: <http://purl.oclc.org/NET/UNIS/fiware/iot-
lite#>
                        Prefix dul:
<http://www.loa.istc.cnr.it/ontologies/DUL.owl#>
                        Prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
                        Prefix time: <http://www.w3.org/2006/time#>
                        Prefix m3-lite: <http://purl.org/iot/vocab/m3-lite#>
                        Prefix xsd: <http://www.w3.org/2001/XMLSchema#>
                        select ?sensorID (max(?ti)
as ?time) ?value ?latitude ?longitude
                        where {
                            ?o a ssn:Observation.
                            ?o ssn:observedBy ?sensorID.
                            ?o ssn:observedProperty ?qk.
                            Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                            ?o ssn:observationSamplingTime ?t.
                            ?o geo:location ?point.
                            ?point geo:lat ?latitude.
```



```
                            ?point geo:long ?longitude.
                            ?t time:inXSDDateTime ?ti.
                            ?o ssn:observationResult ?or.
                            ?or  ssn:hasValue ?v.
                            ?v dul:hasDataValue ?value.
                            {
                                select  (max(?dt)as ?ti) ?sensorID
                                where {
                                    ?o a ssn:Observation.
                                    ?o ssn:observedBy ?sensorID.
                                    ?o ssn:observedProperty ?qk.
                                    Values ?qk {m3-lite:Sound m3-
lite:SoundPressureLevelAmbient}
                                    ?o ssn:observationSamplingTime ?t.
                                    ?t time:inXSDDateTime ?dt.
                                }group by (?sensorID)
                            }
                            FILTER (
                               (xsd:double(?latitude) >= "-90"^^xsd:double)
                            && (xsd:double(?latitude) <= "90"^^xsd:double)
                            && ( xsd:double(?longitude) >= "-180"^^xsd:double)
                            && ( xsd:double(?longitude) <= "180"^^xsd:double)
                            )
                            FILTER(?value<="45"^^xsd:double)
                        } group by ?sensorID ?time ?value ?latitude ?longitude
                    ]]></query>
                </prt:query-request>
            </fed:queryControl>
        </fed:FISMO>
    </fed:FEMO>
</fed:FEDSpec>
```


ANNEX III QUESTIONNAIRE OF EXPERIMENT EVALUATION FROM FIESTA-IOT POINT OF VIEW

Evaluation questionnaire:Template

Questionnaire		Suggestion for evaluation
Feasibility	What kinds of sensor data do you need? (e.g. temperature, humidity, etc.)	-1 point for each kind of sensor that are not available in FIESTA-IoT resources
	Does the experiment need data from a specific place? (e.g. Paris, Tokyo, etc.). If so, specify the place(s)	+2 point if no -1 point for each requested location that is not available in FIESTA-IoT
	Do you need to filter the data during discovery/retrieve? If so, what are the criteria? (location, phenomena, time, etc.)	-1 point for each criteria that is not implemented in FIESTA-IoT
	Does the experiment need external data to accomplish the goal? If so, which external data do you need?	-1 point if the answer is yes
	How the experiment will consume data? (request-based or subscription-based)	-1 point if subscription-based for the moment as the function is not yet stable. Will be neutral in the future
	In case of request-based consumption, what is the expected request rate?	-1 point if the rate<10s
	In case of subscription-based data consumption, what is the expected notification rate?	-1 point if the rate<10s
	Do you need third party tools to accomplish the experiment?	Neutral question
Feedback	What tools do you need among the ones provided by FIESTA-IoT? (refer to the FIESTA-IoT tool list)	+ 1 point for each FIESTA-IoT tool
	How many FIESTA-IoT testbeds do you need to accomplish the experiment?	+ 1 point for each testbed involved in the experiment
	To what extend will the experiment use semantic data? (e.g. only for discovery, produce semantic data, etc.)	+1 point for each basic use (i.e. discovery, retrieve, store), +2 points for knowledge producing operations (i.e. reasoning, cross-field operations)
	Will the experiment generate new knowledge from the requested data and provide it back to	+ 4 if provide knowledge back to FIESTA-IoT

	FIESTA-IoT knowledge base?	

Evaluation questionnaire answered: Data Assembly and Services Portability Experiment

Questionnaire		Answer
Feasibility	What kinds of sensor data do you need? (e.g. temperature, humidity, etc.)	Our experiment is automatically instantiating a set of analytics functions (data statistics such as average, minimum, maximum, but also sensor deployment metrics like observation density per area, number of active sensors of a certain type per virtual entity) on every available kind of sensor data.
	Does the experiment need data from a specific place? (e.g. Paris, Tokyo, etc.). If so, specify the place(s)	No
	Do you need to filter the data during discovery/retrieve? If so, what are the criteria? (location, phenomena, time, etc.)	No.
	Does the experiment need external data to accomplish the goal? If so, which external data do you need?	No, all the inputs can be taken from the FIESTA-IoT platform
	How the experiment will consume data? (request-based or subscription-based)	Can work in both modes,
	In case of request-based consumption, what is the expected request rate?	~3 minutes
	In case of subscription-based data consumption, what is the expected notification rate?	~1 minute
	Do you need third party tools to accomplish the experiment?	Yes.
Feedback	What tools do you need among the ones provided by FIESTA-IoT? (refer to the FIESTA-IoT tool list)	OpenAM security, Semantic Data Repository, IoT-Registry API and IoT Service endpoints
	How many FIESTA-IoT testbeds do you need to accomplish the experiment?	At least two, but the more the better.
	To what extend will the experiment use semantic data? (e.g. only for discovery, produce semantic data, etc.)	For the process of data acquisition and analytics: historical query, resource discovery and data

		analytics execution.
	Will the experiment generate new knowledge from the requested data and provide it back to FIESTA-IoT knowledge base?	New knowledge will be produced but at the moment there is not plan to push it back to FIESTA-IoT.

Evaluation questionnaire answered: Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access

Questionnaire		Answer
Feasibility	What kinds of sensor data do you need? (e.g. temperature, humidity, etc.)	Environmental data (temperature, illuminance, atmospheric pressure, relative humidity, windspeed, solar radation, etc.)
	Does the experiment need data from a specific place? (e.g. Paris, Tokyo, etc.). If so, specify the place(s)	No
	Do you need to filter the data during discovery/retrieve? If so, what are the criteria? (location, phenomena, time, etc.)	Yes. At the time of writing, location, phenomena and time queries will be necessary
	Does the experiment need external data to accomplish the goal? If so, which external data do you need?	No, all the inputs can be taken from the FIESTA-IoT platform
	How the experiment will consume data? (request-based or subscription-based)	Can work in both modes, but preferably through a subscription-based operation
	In case of request-based consumption, what is the expected request rate?	~5 minutes
	In case of subscription-based data consumption, what is the expected notification rate?	~1 minute
	Do you need third party tools to accomplish the experiment?	Yes.
Feedback	What tools do you need among the ones provided by FIESTA-IoT? (refer to the FIESTA-IoT tool list)	OpenAM security, IoT-Registry API and IoT Service endpoints
	How many FIESTA-IoT testbeds do you need to accomplish the experiment?	Every testbed providing environmental data is welcome. We do not have a particular goal on this metric
	To what extend will the experiment use semantic data? (e.g. only for discovery, produce semantic data, etc.)	Resource discovery and extraction of data (observations)
	Will the experiment generate new knowledge from the requested data and provide it back to	No (in principle)

	FIESTA-IoT knowledge base?	

Evaluation questionnaire answered: Large Scale Crowdsensing Experiment

Questionnaire		Answer
Feasibility	What kinds of sensor data do you need? (e.g. temperature, humidity, etc.)	Sound Sensor. This is already available in the taxonomy and the testbeds are already providing data
	Does the experiment need data from a specific place? (e.g. Paris, Tokyo, etc.). If so, specify the place(s)	No, it is large scale and does not depend on the location.
	Do you need to filter the data during discovery/retrieve? If so, what are the criteria? (location, phenomena, time, etc.)	Yes, we need most recent observations. This can be done while querying the registry.
	Does the experiment need external data to accomplish the goal? If so, which external data do you need?	No.
	How the experiment will consume data? (request-based or subscription-based)	It will be request based. We have a portal that will allow the end users or the citizens to view the map of the noisy/quite places. Thus once the data is sent to the experimenter (us) by the EEE, we will store the data and will consume it if there is a request from a citizen or end user.
	In case of request-based consumption, what is the expected request rate?	As soon as possible. We expect it to be fast
	In case of subscription-based data consumption, what is the expected notification rate?	NA
	Do you need third party tools to accomplish the experiment?	Yes.
Feedback	What tools do you need among the ones provided by FIESTA-IoT? (refer to the FIESTA-IoT tool list)	We need: Portal, OpenAM security, EEE, EMC, ERM and IoT-Registry
	How many FIESTA-IoT testbeds do you need to accomplish the experiment?	All those have the sound sensor. Currently there are 3 testbeds: Smart Santander, Smart ICS and SoundCity.
	To what extend will the experiment use semantic	Only to get observations.

	data? (e.g. only for discovery, produce semantic data, etc.)	
	Will the experiment generate new knowledge from the requested data and provide it back to FIESTA-IoT knowledge base?	No.

ANNEX IV EVALUATION EXPERIMENT INTEGRATION AND IMPLEMENTATION CHECKLIST

Checklist template

Topic name			Topic Description	Answer
Learning phase	Attendance to training WS		Have the experimenter attended the training workshop?	Y/N
	Use of support channels	e-mail	Have the experimenter used the helpdesk support tools?	Y/N
		ticket system		Y/N
		live chat		Y/N
	Consult documentation		Have the experimenter consulted the on-line documentation? *If yes, which part(s) of it?*	Y/N *which one(s)*
	Re-use sample material		Have the experimenter used the available sample material? *If yes, which one(s)?*	Y/N *which one(s)*
	Follow suggested best-practices		Have the experimenter followed the suggested best-practices? *If yes, which one(s)?*	Y/N *which one(s)*
Design and development phase	Use of FIESTA-IoT tools	Experiment Related tools	Which tools from the FIESTA-IoT Platform portfolio does the experiment use?	Y/N
		SPARQL endpoint		Y/N
		Resource browser		Y/N
		REST access to datasets		Y/N
	Suggest additional functionalities based on experience		Have the experimenter proposed additional functionalities that could be benefitial for future experiments? *If yes, which one(s)?*	Y/N *which one(s)*
	Provide code / enhancements / modules / tools		Have the experimenter provided code/enhancements/modules/tools that could be benefitial for future experiments? *If yes, which one(s)?*	Y/N *which one(s)*
	Support objective assessment of platform non-functional requirements		Does the experiment allow objective assessment of the FIESTA-IoT platform non-functional requirements? *If yes, which one(s)?*	Y/N *which one(s)*

Checklist: Data Assembly and Services Portability Experiment

Topic name			Topic Description	Answer
Learning phase	Attendance to training WS		Have the experimenter attended the training workshop?	NA
	Use of support channels	e-mail	Have the experimenter used the helpdesk support tools?	NA
		ticket system		NA
		live chat		NA
	Consult documentation		Have the experimenter consulted the on-line documentation? *If yes, which part(s) of it?*	NA
	Re-use sample material		Have the experimenter used the available sample material? *If yes, which one(s)?*	NA
	Follow suggested best-practices		Have the experimenter followed the suggested best-practices? *If yes, which one(s)?*	NA
ADesign and development phase	Use of FIESTA-IoT tools	Experiment Related tools	Which tools from the FIESTA-IoT Platform portfolio does the experiment use?	N
		SPARQL endpoint		Y
		Resource browser		N
		REST access to datasets		Y
	Suggest additional functionalities based on experience		Have the experimenter proposed additional functionalities that could be benefitial for future experiments? *If yes, which one(s)?*	N
	Provide code / enhancements / modules / tools		Have the experimenter provided code/enhancements/modules/tools that could be benefitial for future experiments? *If yes, which one(s)?*	N
	Support objective assessment of platform non-functional requirements		Does the experiment allow objective assessment of the FIESTA-IoT platform non-functional requirements? *If yes, which one(s)?*	N *Not systematically*

Checklist: Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access

Topic name			Topic Description	Answer
Learning phase	Attendance to training WS		Have the experimenter attended the training workshop?	NA
	Use of support channels	e-mail	Have the experimenter used the helpdesk support tools?	NA
		ticket system		NA
		live chat		NA
	Consult documentation		Have the experimenter consulted the on-line documentation? *If yes, which part(s) of it?*	NA
	Re-use sample material		Have the experimenter used the available sample material? *If yes, which one(s)?*	NA
	Follow suggested best-practices		Have the experimenter followed the suggested best-practices? *If yes, which one(s)?*	NA
ADesign and development phase	Use of FIESTA-IoT tools	Experiment Related tools	Which tools from the FIESTA-IoT Platform portfolio does the experiment use?	Y
		SPARQL endpoint		Y
		Resource browser		Y
		REST access to datasets		Y
	Suggest additional functionalities based on experience		Have the experimenter proposed additional functionalities that could be benefitial for future experiments? *If yes, which one(s)?*	Y *Performance monitoring tool* *Feeback module*
	Provide code / enhancements / modules / tools		Have the experimenter provided code/enhancements/modules/tools that could be benefitial for future experiments? *If yes, which one(s)?*	Y *Application source code which can be easily broken down into independent module*
	Support objective assessment of platform non-functional requirements		Does the experiment allow objective assessment of the FIESTA-IoT platform non-functional requirements? *If yes, which one(s)?*	Y *Continuous performance assessment of the platform*

Checklist: Large scale crowdsensing experiment

Topic name			Topic Description	Answer
Learning phase	Attendance to training WS		Have the experimenter attended the training workshop?	NA
	Use of support channels	e-mail	Have the experimenter used the helpdesk support tools?	NA
		ticket system		NA
		live chat		NA
	Consult documentation		Have the experimenter consulted the on-line documentation? *If yes, which part(s) of it?*	NA *Being the creator of EEE and other experimenter related tools, we knew beforehand.*
	Re-use sample material		Have the experimenter used the available sample material? *If yes, which one(s)?*	NA *Same as above*
	Follow suggested best-practices		Have the experimenter followed the suggested best-practices? *If yes, which one(s)?*	Y *All those are suggested for the best working of the EEE and experiment related tools*
ADesign and development phase	Use of FIESTA-IoT tools	Experiment Related tools	Which tools from the FIESTA-IoT Platform portfolio does the experiment use?	Y
		SPARQL endpoint		NA
		Resource browser		NA
		REST access to datasets		NA
	Suggest additional functionalities based on experience		Have the experimenter proposed additional functionalities that could be benefitial for future experiments? *If yes, which one(s)?*	N *Not as of now*
	Provide code / enhancements / modules / tools		Have the experimenter provided code/enhancements/modules/tools that could be benefitial for future experiments? *If yes, which one(s)?*	N *No not as of now.*
	Support objective assessment of platform non-functional requirements		Does the experiment allow objective assessment of the FIESTA-IoT platform non-functional requirements? *If yes, which one(s)?*	N *Not as of now*

ANNEX V QUESTIONNAIRE: VALIDATION OF THE FIESTA-IOT RESOURCES

Questionnaire template

Starting the experimentation

Part I: documentation

Q1.Did you use the documentation for experimenters provided on the moodle ?

- Yes, I consulted almost all the documents
  - Please, specify the ones you mainly used……..
- Yes, but only some documents
  - Please, specify the ones you mainly used……..
- No, I didn't

Q2.Were you able to find the needed information?

- Always
- Most of the time
- Sometimes
- Never

Q3.Do you believe that some documentation is missing?

- Yes
  - Please specify……
- No

Q4.How would you rate the quality of the documentation provided to discover the platform?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
➢ **Documentation about FEDSPEC**	☐	☐	☐	☐	☐	☐
➢ **Documentation about APIs**	☐	☐	☐	☐	☐	☐
➢ **Documentation about Ontology**	☐	☐	☐	☐	☐	☐
➢ **Documentation about SPARQL queries**	☐	☐	☐	☐	☐	☐
➢ **Documentation about installing Experiment Data Receiver**	☐	☐	☐	☐	☐	☐
➢ **Experiment Execution process and guidelines**	☐	☐	☐	☐	☐	☒

Q5.How would you rate the relevance of the documentation to support you to set up your experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
➢ **Documentation about FEDSPEC**	☐	☐	☐	☐	☐	☐
➢ **Documentation about APIs**	☐	☐	☐	☐	☐	☐
➢ **Documentation about Ontology**	☐	☐	☐	☐	☐	☐
➢ **Documentation about SPARQL queries**	☐	☐	☐	☐	☐	☐
➢ **Documentation about installing Experiment Data Receiver**	☐	☐	☐	☐	☐	☐
➢ **Experiment Execution process and guidelines**	☐	☐	☐	☐	☐	☐

Part II: ease of setting up, ease of deployment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Q6.How would you rate the FEDSPEC creation process?	☐	☐	☐	☐	☐	☐
Q7.How would you rate the SPARQL Queries creation process?	☐	☐	☐	☐	☐	☐
Q8.How would you rate the integration and deployment process?	☐	☐	☐	☐	☐	☐
Q9.How would you rate the quality and quantity of available data?	☐	☐	☐	☐	☐	☐
Q10.How would you rate the performance of EEE module?	☐	☐	☐	☐	☐	☐
Q11.How would you qualify the quality and relevance of tools which have been made available to you?	☐	☐	☐	☐	☐	☐
Q12.How would you qualify the quality of FIESTA-IoT APIs?	☐	☐	☐	☐	☐	☐
Q13.How would you qualify the easy of installing Experiment Data Receiver (Excellent being very easy and Poor being very hard)	☐	☐	☐	☐	☐	☐

Q14.Do you prefer to move to API-based solution rather than using the experiment portal?

- Yes
- No

If Yes, Please specify the reason……

Q15. How much time have you spent in total to integrate the FIESTA-IoT tools in your experiment for having the first experiment prototype working (it counts only the time used to setup the FIESTA-IoT tools such as APIs connector, EMC, Data Receiver setup and so on, without counting effort for visualization tools or set up of external tools):

	LESS THAN 1 WEEK	LESS THAN 2 WEEKS	LESS THAN 1 MONTH	LESS THAN 2 MONTHS	MORE THAN 2 MONTHS
➢ **Get Started Level***	☐	☐	☐	☐	☐
➢ **Basic Integration Level**	☐	☐	☐	☐	☐
➢ **Full Integration Level**	☐	☐	☐	☐	☐

* "Get Started Level" corresponds to following the the instructions in the handbook, "Basic Integration level" corresponds to the first integration of your experiment to Fiesta-IoT, and "Full Integration level" refers to a final integration after necessary fine-tuning of your experiment

During the experimentation

Q16.How would you rate your experience of the FIESTA-IoT platform during the experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
➢ **Availability of the platform**	☐	☐	☐	☐	☐
➢ **Performance of the platform**	☐	☐	☐	☐	☐
➢ **Interaction with FIESTA-IoT team**	☐	☐	☐	☐	☐

Q17.Please give us all comments you may have about your experience during the experimentation?

……

Ending the experiment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
Q18.Overall, how do you qualify your experience on	☐	☐	☐	☐	☐

FIESTA-IoT platform?	

Q19. Are you satisfied with the results you obtained?

- Yes, I'm very satisfied
- Yes, but only partially
  - Explain why……
- No, I'm not
  - Explain why……

Q20. Would you recommend FIESTA-IoT platform to other experimenters?

- Yes
- No

Questionnaire answered: Data Assembly and Services Portability Experiment

Starting the experimentation

Part I: documentation

Q1.Did you use the documentation for experimenters provided on the moodle ?

- Yes, but only some documents
  - I have mostly consulted the Handbook for getting information about the security system and getting access to the FIESTA-IoT platform. Furthermore I have very often consulted the FIESTA-IoT ontology page and the FIESTA-IoT API documentation. Finally I have leverage the information contained in the handbook for building my SPARQL queries

Q2.Were you able to find the needed information?

- Always

Q3. Do you believe that some documentation is missing?

- No, I found the needed information

Q4. How would you rate the quality of the documentation provided to discover the platform?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☐	☐	☐	☐	☐	☒
Documentation about APIs	☒	☐	☐	☐	☐	☐
Documentation about Ontology	☒	☐	☐	☐	☐	☐
Documentation about SPARQL queries	☐	☐	☒	☐	☐	☐
Documentation about installing Experiment Data Receiver	☐	☐	☐	☐	☐	☒
Experiment Execution process and guidelines	☐	☐	☐	☐	☐	☒

Q5. How would you rate the relevance of the documentation to support you to set up your experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☐	☐	☐	☐	☐	☒

➢ **Documentation about APIs**	☒	☐	☐	☐	☐	☐
➢ **Documentation about Ontology**	☒	☐	☐	☐	☐	☐
➢ **Documentation about SPARQL queries**	☐	☐	☒	☐	☐	☐
➢ **Documentation about installing Experiment Data Receiver**	☐	☐	☐	☐	☐	☒
➢ **Experiment Execution process and guidelines**	☐	☐	☐	☐	☐	☒

Part II: ease of setting up, ease of deployment

	EXCELLENT	**VERY GOOD**	**GOOD**	**FAIR**	**POOR**	**N/A**
Q6. How would you rate the FEDSPEC creation process?	☐	☐	☐	☐	☐	☒
Q7. How would you rate the SPARQL Queries creation process?	☒	☐	☐	☐	☐	☐
Q8. How would you rate the integration and deployment process?	☐	☒	☐	☐	☐	☐
Q9. How would you rate the quality and quantity of available data?	☐	☒	☐	☐	☐	☐
Q10. How would you rate the performance of EEE module?	☐	☐	☐	☐	☐	☒
Q11. How would you qualify the quality and relevance of tools which have been made available to you?	☒	☐	☐	☐	☐	☐
Q12. How would you qualify the quality of FIESTA-IoT APIs?	☒	☐	☐	☐	☐	☐
Q13. How would you qualify the easy of installing Experiment Data Receiver (Excellent being very easy and Poor being very hard)	☐	☐	☐	☐	☐	☒

Q14. Do you prefer to move to API-based solution rather than using the experiment portal?

- Yes

If Yes, Please specify the reason

I went to the API-based solution from the first phase since we are more accustomed to connector creation.

Q15. How much time have you spent in total to integrate the FIESTA-IoT tools in your experiment for having the first experiment prototype working (it counts only the time used to setup the FIESTA-IoT tools such as APIs connector, EMC, Data Receiver setup and so on, without counting effort for visualization tools or set up of external tools):

	LESS THAN 1 WEEK	LESS THAN 2 WEEKS	LESS THAN 1 MONTH	LESS THAN 2 MONTHS	MORE THAN 2 MONTHS
➢ **Get Started Level***	☒	☐	☐	☐	☐
➢ **Basic Integration Level**	☒	☐	☐	☐	☐
➢ **Full Integration Level**	☐	☒	☐	☐	☐

* “Get Started Level” corresponds to following the instructions in the handbook, “Basic Integration level” corresponds to the first integration of your experiment to Fiesta-IoT, and “Full Integration level” refers to a final integration after necessary fine-tuning of your experiment

During the experimentation

Q16. How would you rate your experience of the FIESTA-IoT platform during the experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
➢ **Availability of the platform**	☐	☒	☐	☐	☐
➢ **Performance of the platform**	☒	☐	☐	☐	☐
➢ **Interaction with FIESTA-IoT team**	☒	☐	☐	☐	☐

Q17. Please give us all comments you may have about your experience during the experimentation?

We have found the FIESTA-IoT platform very reliable and, in case of any small issue, the direct communication with the FIESTA-IoT support helped us on have it quickly solved (either on our side or on their side).

Ending the experiment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
Q18. Overall, how do you qualify your experience on FIESTA-IoT platform?	☒	☐	☐	☐	☐

Q19. Are you satisfied with the results you obtained?

- Yes, I'm very satisfied

We have already reached a very good quality of the experiment and a big step of innovation toward Smart City applications. The results were already shown in different events like big fair (CeBIT 2017) or to many industrial partners. The amount of data handled is already of a good size but with the integration of third parties extension already planned, we are confident to reach much higher results.

Q20. Would you recommend FIESTA-IoT platform to other experimenters?

- Yes

Questionnaire answered: Dynamic Discovery of IoT Resources for Testbed Agnostic Data Access

Starting the experimentation

Part I: documentation

Q1.Did you use the documentation for experimenters provided on the moodle ?

- Yes, but only some documents
  - We have mainly paid attention to the IoT-Registry API and OpenAM documentation sections, as they are our main interplay points with the FIESTA-IoT platform.

Q2.Were you able to find the needed information?

- Always

Q3.Do you believe that some documentation is missing?

- No

Q4.How would you rate the quality of the documentation provided to discover the platform?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☐	☐	☐	☐	☐	☒
Documentation about APIs	☒	☐	☐	☐	☐	☐
Documentation about Ontology	☒	☐	☐	☐	☐	☐
Documentation about SPARQL queries	☐	☒	☐	☐	☐	☐
Documentation about installing Experiment Data Receiver	☐	☐	☐	☐	☐	☒
Experiment Execution process and guidelines	☐	☐	☐	☐	☐	☒

Q5.How would you rate the relevance of the documentation to support you to set up your experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☐	☐	☐	☐	☐	☒
Documentation about APIs	☒	☐	☐	☐	☐	☐
Documentation about Ontology	☒	☐	☐	☐	☐	☐
Documentation about SPARQL queries	☐	☒	☐	☐	☐	☐
Documentation about installing	☐	☐	☐	☐	☐	☒

Experiment Data Receiver						
➢ **Experiment Execution process and guidelines**	☐	☐	☐	☐	☐	☒

Part II: ease of setting up, ease of deployment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Q6.How would you rate the FEDSPEC creation process?	☐	☐	☐	☐	☐	☒
Q7.How would you rate the SPARQL Queries creation process?	☐	☒	☐	☐	☐	☐
Q8.How would you rate the integration and deployment process?	☐	☒	☐	☐	☐	☐
Q9.How would you rate the quality and quantity of available data?	☐	☐	☒	☐	☐	☐
Q10.How would you rate the performance of EEE module?	☐	☐	☐	☐	☐	☒
Q11.How would you qualify the quality and relevance of tools which have been made available to you?	☒	☐	☐	☐	☐	☐
Q12.How would you qualify the quality of FIESTA-IoT APIs?	☒	☐	☐	☐	☐	☐
Q13.How would you qualify the easy of installing Experiment Data Receiver (Excellent being very easy and Poor being very hard)	☐	☐	☐	☐	☐	☒

Q14.Do you prefer to move to API-based solution rather than using the experiment portal?

- Yes

I went to the API-based solution from the first phase since we are more accustomed to connector creation.

We opted for the direct use of the API because we think it offers more flexibility for skilled experimenters (or application developers).

Q15. How much time have you spent in total to integrate the FIESTA-IoT tools in your experiment for having the first experiment prototype working (it counts only the time used to setup the FIESTA-IoT tools such as APIs connector, EMC, Data Receiver setup and so on, without counting effort for visualization tools or set up of external tools):

	LESS THAN 1 WEEK	LESS THAN 2 WEEKS	LESS THAN 1 MONTH	LESS THAN 2 MONTHS	MORE THAN 2 MONTHS
➢ **Get Started Level***	☒	☐	☐	☐	☐
➢ **Basic Integration Level**	☒	☐	☐	☐	☐
➢ **Full Integration Level**	☐	☒	☐	☐	☐

* "Get Started Level" corresponds to following the instructions in the handbook, "Basic Integration level" corresponds to the first integration of your experiment to Fiesta-IoT, and "Full Integration level" refers to a final integration after necessary fine-tuning of your experiment

During the experimentation

Q16.How would you rate your experience of the FIESTA-IoT platform during the experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
➢ **Availability of the platform**	☐	☒	☐	☐	☐
➢ **Performance of the platform**	☐	☐	☒	☐	☐
➢ **Interaction with FIESTA-IoT team**	☒	☐	☐	☐	☐

Q17.Please give us all comments you may have about your experience during the experimentation?

Thanks to the clear documentation we have only had to follow the instructions in order to get the information needed from the FIESTA-IoT platform. The toughest part was on our own court, being the implementation of the experiment the real tricky thing.

Ending the experiment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
Q18.Overall, how do you qualify your experience on FIESTA-IoT platform?	☐	☒	☐	☐	☐

Q19. Are you satisfied with the results you obtained?

- Yes, I'm very satisfied

Q20. Would you recommend FIESTA-IoT platform to other experimenters?

- Yes

Questionnaire answered: Large scale crowdsensing experiment

Starting the experimentation

Part I: documentation

Q1. Did you use the documentation for experimenters provided on the moodle ?

- Yes, but only some documents
  - Handbook. Other than courses and documentation on Moodle I also consulted ontology documentation and deliverables.

Q2. Were you able to find the needed information?

- Always

Q3. Do you believe that some documentation is missing?

- No, the features that were provided were clearly documented.

Q4. How would you rate the quality of the documentation provided to discover the platform?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☑	☐	☐	☐	☐	☐
Documentation about APIs	☐	☐	☐	☐	☐	☑
Documentation about Ontology	☑	☐	☐	☐	☐	☐
Documentation about SPARQL queries	☐	☐	☐	☐	☐	☑
Documentation about installing Experiment Data Receiver	☑	☐	☐	☐	☐	☐
Experiment Execution process and guidelines	☑	☐	☐	☐	☐	☐

Q5. How would you rate the relevance of the documentation to support you to set up your experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Documentation about FEDSPEC	☑	☐	☐	☐	☐	☐
Documentation about APIs	☐	☐	☐	☐	☐	☑
Documentation about Ontology	☑	☐	☐	☐	☐	☐

➢ **Documentation about SPARQL queries**	☐	☐	☐	☐	☐	☑
➢ **Documentation about installing Experiment Data Receiver**	☑	☐	☐	☐	☐	☐
➢ **Experiment Execution process and guidelines**	☑	☐	☐	☐	☐	☐

Part II: ease of setting up, ease of deployment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR	N/A
Q6. How would you rate the FEDSPEC creation process?	☑	☐	☐	☐	☐	☐
Q7. How would you rate the SPARQL Queries creation process?	☐	☐	☐	☐	☐	☑
Q8. How would you rate the integration and deployment process?	☑	☐	☐	☐	☐	☐
Q9. How would you rate the quality and quantity of available data?	☐	☐	☐	☑	☐	☐
Q10. How would you rate the performance of EEE module?	☑	☐	☐	☐	☐	☐
Q11. How would you qualify the quality and relevance of tools which have been made available to you?	☑	☐	☐	☐	☐	☐
Q12. How would you qualify the quality of FIESTA-IoT APIs?	☐	☐	☐	☐	☐	☑
Q13. How would you qualify the easy of installing Experiment Data Receiver (Excellent being very easy and Poor being very hard)	☑	☐	☐	☐	☐	☐

Q14. Do you prefer to move to API-based solution rather than using the experiment portal?

- No

Q15. How much time have you spent in total to integrate the FIESTA-IoT tools in your experiment for having the first experiment prototype working (it counts only the time used to setup the FIESTA-IoT tools such as APIs connector, EMC, Data Receiver setup and so on, without counting effort for visualization tools or set up of external tools):

	LESS THAN 1 WEEK	LESS THAN 2 WEEKS	LESS THAN 1 MONTH	LESS THAN 2 MONTHS	MORE THAN 2 MONTHS
➢ **Get Started Level***	☒	☐	☐	☐	☐
➢ **Basic Integration Level**	☒	☐	☐	☐	☐
➢ **Full Integration Level**	☒	☐	☐	☐	☐

* “Get Started Level” corresponds to following the instructions in the handbook, “Basic Integration level” corresponds to the first integration of your experiment to Fiesta-IoT, and “Full Integration level” refers to a final integration after necessary fine-tuning of your experiment

During the experimentation

Q16. How would you rate your experience of the FIESTA-IoT platform during the experimentation?

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
➢ **Availability of the platform**	☐	☑	☐	☐	☐
➢ **Performance of the platform**	☐	☑	☐	☐	☐
➢ **Interaction with FIESTA-IoT team**	☑	☐	☐	☐	☐

Q17. Please give us all comments you may have about your experience during the experimentation?

Easy to use solution enabled us configure our experiment on FIESTA-IoT platform with ease.

Ending the experiment

	EXCELLENT	VERY GOOD	GOOD	FAIR	POOR
Q18. Overall, how do you qualify your experience on FIESTA-IoT platform?	☑	☐	☐	☐	☐

Q19. Are you satisfied with the results you obtained?

- Yes, but only partially
  - As the data is recently fed in to the system, volume of data needed for our experiment is currently missing. We would like to continuously monitor the data volume provided to us. We believe it will grow in near future.

Q20. Would you recommend FIESTA-IoT platform to other experimenters?

- Yes

