A Reference Architecture for Federating IoT Infrastructures Supporting Semantic Interoperability

Francois Carrez, Tarek Elsaleh
Institute for Communication Systems
University of Surrey
Guilford, UK
f.carrez@surrey.ac.uk
t.elsaleh@surrey.ac.uk

David Gómez, Luis Sánchez, Jorge Lanza
Network Planning and Mobile Communications Laboratory
Universidad de Cantabria
Santander, Spain
dgomez@tlmat.unican.es
lsanchez@tlmat.unican.es
jlanza@tlmat.unican.es

Paul Grace
IT Innovation
University of Southampton
Southampton, UK
pjg@it-innovation.soton.ac.uk

Abstract

The Internet-of-Things (IoT) is unanimously identified as one of the main pillars of future smart scenarios. However, despite the growing number of IoT deployments, the majority of IoT applications tend to be self-contained, thereby forming vertical silos. Indeed, the ability to combine and synthesize data streams and services from diverse IoT platforms and testbeds, holds the promise to increase the potential of smart applications in terms of size, scope and targeted business context. This paper describes the system architecture for the FIESTA-IoT platform, whose main aim is to federate a large number of testbeds across the planet, in order to offer experimenters the unique experience of dealing with a large number of semantically interoperable data sources. This system architecture was developed by following the Architectural Reference Model (ARM) methodology promoted by the IoT-A project (FP7 "light house" project on Architecture for the Internet of Things). Through this process, the FIESTA-IoT architecture is composed of a set of Views that deals with a "logical" functional decomposition (Functional View, FV) and data structuring and annotation, data flows and inter-functional component interactions (Information View, IV).

Index Terms

Internet-of-Things; Architecture; Testbed; Federation

I. Introduction

Recent advances in the Internet of Things (IoT) have progressively moved in different directions (e.g. designing technology, increasing the number of inter-connected entities and not less important the security aspects of IoT). IoT advances have drawn a common big challenge that focuses on the integration of the IoT generated data. The key challenge is to provide a common sharing model (or models) organizing the information coming from the connected IoT services and systems and more importantly able to offer them as services in order to optimize the design of new IoT systems and facilitate the generation of solutions more rapidly. This vision of integrating IoT platforms within cloud infrastructures is related to several scientific challenges, such as the need to aggregate and ensure the interoperability of data streams stemming from different IoT testbeds, as well as the need to provide tools for building applications that horizontally integrate diverse IoT Solutions.

The focus of the FIESTA-IoT project is on the paradigm of formulating and managing IoT data from heterogeneous systems and environments and their underlying resources (such as smart devices, sensors, actuators, etc.). This paper describes the system architecture for the FIESTA-IoT platform, aiming at federating a large number of testbeds across the planet in order to offer experimenters with a unique experience of dealing and experimenting with a large number of semantically interoperable data sources.

The rest of the paper is structured as follows. Section 2 summarizes the main features that have been considered in the process of defining the FIESTA-IoT Platform architecture. A brief summary of the IoT-ARM (Architectural Reference Model) structure and methodology is provide in Section 3 as a baseline of related reference work. The Functional and Information Views of the FIESTA-IoT Platform are provided in Section 4. This is the core specification of the FIESTA-IoT Platform architecture. Finally, we conclude the document in Section 5.

II. Related Work

A primary decision of the FIESTA-IoT project was to follow the IoT ARM [1]. This means, in particular, to stick to the Reference Model as defined in the ARM (especially the Domain Model [2] DM - and the Information Model IM-, which are both considered as fixed). The main parts of the ARM consist of the Reference Model (RM), Reference Architecture (RA) and a side part consisting of the associated methodology:

The main parts of the ARM consist of the RM, RA and a side part that undertakes the associated methodology:

- **The IoT RM** consists of a set of models (namely the Domain, Information, Functional, Communication and Security/Trust/Privacy Models). The FIESTA-IoT architecture must comply with them, especially to DM and IM.
- **The IoT RA** consists of a set of Views and Perspectives - as defined by Rozansky and Woods [3] - that actually define the FIESTA-IoT Architecture. The main contribution of this paper is ultimately to sketch the key aspects of these Views and Perspectives.
- **Guidance** that defines the overall process used to derive a concrete architecture out of the ARM. The requirement mapping exercise following the requirement collection phase made as part of the FIESTA-IoT project allowed to derive a preliminary Functional View (FV).

However, the description of an architecture following the methodology defined by the IoT ARM consists in the specification of the different views that shape the IoT RA. The Functional View focuses on the decomposition into Functional Components (FCs), while the Information View describes information flows, interaction between components and structure of information, in compliance with the Information Model.

When elaborating the mapping from the necessary functionalities addressing the already identified FIESTA-IoT Platform set of requirements [4] towards the IoT Functional View, it is important to try to stick, as much as possible, to the list of components already identified in the so-called IoT Native Functional View [1]. However, it is perfectly possible and allowed to introduce new ones.

Along with a description of the FCs within Functional Groups (FGs) it is equally important to get a concise and still precise description of the different FCs implemented and to understand also very clearly how they interact with and position w.r.t. the other components. FIESTA-IoT Architecture description concentrates on a clear textual description of these FCs and FGs but also in the inter-component interplay that will be described as system use-cases.

III. FIESTA-IoT Reference Architecture. Main design considerations

Some aspects and constraints, directly derived from the project objectives and requirements, have been considered. They are briefly below:

- **Compliance to the Architectural Reference Model (ARM) from IoT-A**. While full compliance was not strictly required, the architecture tries to follow as rigorously as possible the whole architectural methodology released by the FP7 “light house” project about Architecture for the IoT.
- **Full support of semantics**. The FIESTA-IoT platform is semantic-enabled, so it is necessary to put in place all mechanisms needed to support semantics (languages, ontologies and tools). A related consideration is that, in order not to deter testbeds which are not semantic-ready from becoming part of the FIESTA-IoT federation, it is necessary to come up with an architecture that needs also to enhance those testbeds (capability-wise) in order to pull them to the level of FIESTA-IoT standard.
- **Compliance to FIESTA-IoT set of ontologies**. Testbeds which are not semantics-ready will have to comply with the ontology defined in FIESTA-IoT [5] so as to ensure full semantic interoperability.
- **Logical Functional Decomposition**. The decomposition into components is a logical one; meaning that when the platform physical components are implemented, there might not be a direct mapping.
- **Technology Agnostic**. Architecture is agnostic to any implementation/design choices.
- **Support multiple stakeholders’ roles**. Different FIESTA-IoT end-user roles and the way they will interact with the FIESTA-IoT platform are defined.
- **Accommodate different levels of technology skills**. While most of interfacing between end-users and the testbeds can be handled by a set of IoT Services, it is considered that providing ”direct” access to data using complex but powerful data-centric languages could be a convenient choice for certain kinds of actors with high semantic skills.
- **Security**. Access to the offered IoT services must only be allowed to authorised persons. Data privacy must be maintained and guaranteed.
- **Message Bus**. Communication channel between testbeds and the FIESTA-IoT platform.

IV. Architecture Functional View

Before describing the different components that are part of the Functional View, it is important to define, on the one hand, the roles and associated duties that actors involved with the FIESTA-IoT platform should endorse and, on the other hand, the different kinds of platform configurations and capabilities that can co-exist under the umbrella of the FIESTA-IoT federation.

A. Roles and IoT Infrastructure Taxonomy

This short section introduces a taxonomy of actors dealing with FIESTA-IoT Federation Platform.

- **Raw-data producers** are responsible for producing the raw data with a low level of metadata. In addition, they are also responsible for describing and publishing IoT Service and Resource semantic description either locally or at the FIESTA-IoT level.

- **(Added-value) Service providers** are providing added-value services (e.g. reasoners, data analytics or other generic enablers) that, in turn, could be combined and used in order to create knowledge (by knowledge producers).
- **Knowledge producers** are involved in leveraging the basic IoT services/Resources provided by the rawdata producers and services provided by the service providers in order to create and store higher-order knowledge.
- **Experimenters** are using the aforementioned services and consuming data provided by the FIESTA-IoT Platform for the sake of their own business.

The Experiment-as-a-Service (EaaS) concept is captured by the service provider role. Experimenters are FIESTA-IoT platform users, while other roles are platform contributors.

Testbeds are typically involved at least in raw-data production and may be as well Knowledge producers if they are willing to provide added-value IoT Services on top of their basic activities. However, three different kind of testbeds have been identified according to its semantic capabilities:

- **Class-I testbed**: These testbeds are fully aligned with FIESTA-IoT. They store locally semantically annotated data (FIESTA-IoT compliant). They also manage locally IoT Service/Resource descriptions and endpoints. They provide a data endpoint for direct data queries. All descriptions are semantically described and compliant with FIESTA-IoT's schemas (i.e. semantic data model).
- **Class-II testbed**: These testbeds were initially not semantic-ready; still they store their data locally. In order to comply with the FIESTA-IoT rules, they will have to implement some functional components. Class-II testbeds will replicate their data, after it has been semantically annotated according to the FIESTA-IoT ontology, to the FIESTA-IoT data repository. Consequently, they do not offer a data endpoint locally; queries to data originating from that testbed will be answered by the central FIESTA-IoT data repository directly.
- **Class-III testbed**: These testbeds were initially neither semantic-ready nor stored any data locally. In order to be part of the FIESTA-IoT federation they will have to integrate few additional FCs.

B. Functional Groups and Component Descriptions

Fig.1 provides a complete picture of the FIESTA-IoT RA, whose FCs are displayed over IoT-A's FGs.

It is important to note that the Functional View at this stage is a logical view. Actually, any component may endorse more than one role, spanning even more than one FG, for instance if a decision is taken to implement only one registry dealing, at the same time, with resource descriptions and data (i.e. observations).

1) Management FG:

- User Management FC. This component is responsible for registering new users within the FIESTA-IoT management database. FIESTA-IoT users can sign up to use the FIESTA-IoT services via a Graphical User Interface (GUI); they can also use the GUI to update their personal information. The registration process includes the issuing of security credentials (despite the management of keys and authentication/access enforcement points are at the Security FG side).
- Web Browsing & Configuration FC: This FC is a web application that builds and provides the FIESTA-IoT actors with a graphical interface for interactively discovering, manipulating and configuring (Create, Read, Update and Delete - CRUD operations) Resources and Services. It relies on the IoT Service/Resource-centric, Web Front-end Sub-FCs.

2) Service Organization FG: This FG and the IoT Process Management FG embrace the tools for modelling, creating and supporting the execution of experiments that are used by experimenters to access and make use respectively of data available at the FIESTA-IoT platform (and federated testbeds) and the myriad of IoT services also available. On the other hand, they can be used by added-value service providers for the creation of more complex experiments.

- **IoT Service Composer**: The IoT Service Composer FC is used to compose IoT Services or added-value services (like reasoners, aggregators etc.) into higher-level (still IoT) services. Such services can be used, for instance, for building abstract sensors (like combining different kind of sensors e.g. particle sensor / CO_2 / CO sensor in order to infer an air quality sensor).
- **IoT Composite Service Execution Engine**: This component is responsible for executing the Composite IoT Services, which are described at the IoT Service/Resource Registry FC side but actually stored locally in this component. It offers a REST interface that triggers the retrieval and execution of the Composite Service so that the REST request can be answered.

3) IoT Process Management FG:

- **Experiment modelling**: This component allows the modelling, either through graphical interface or scripting, of an experiment. It relies in particular on off-the-shelf interfaces provided by the other FCs for querying data, searching and invoking IoT Services (exposing resources), etc.
- **Experiment Execution Engine**: This component is responsible for executing the experiment (see above). IoT Services referred to within the experiment are therefore invoked from this component.

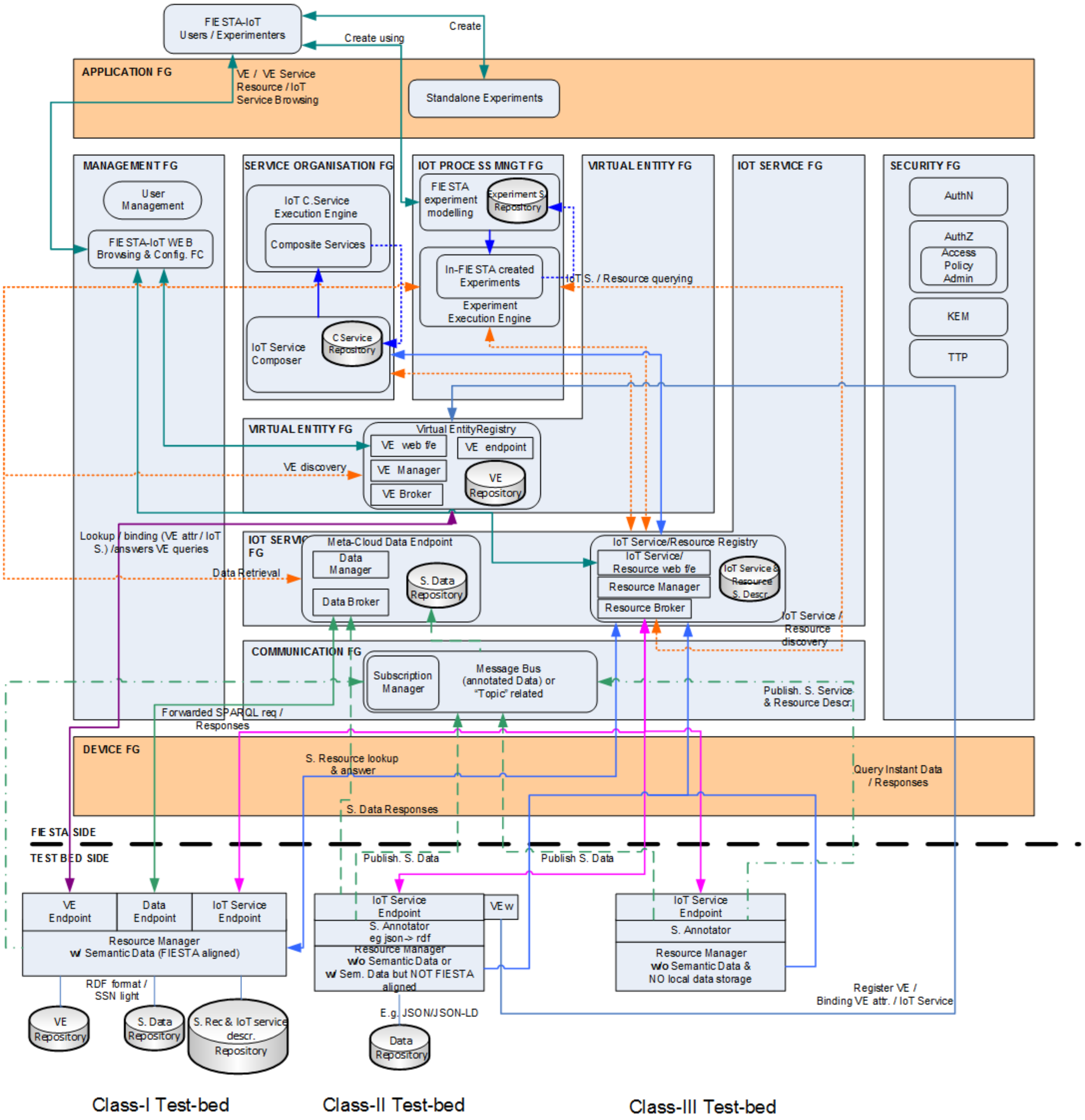
FIESTA-IoT Users / Experimenters
Create
Create using
APPLICATION FG
VE / VE Service Resource / IoT Service Browsing
Standalone Experiments
MANAGEMENT FG
User Management
FIESTA-IoT WEB Browsing & Config. FC
SERVICE ORGANISATION FG
IoT C.Service Execution Engine
Composite Services
IoT Service Composer
C.Service Repository
IOT PROCESS MNGT FG
FIESTA experiment modelling
Experiment S. Repository
In-FIESTA created Experiments
Experiment Execution Engine
VIRTUAL ENTITY FG
IOT SERVICE FG
SECURITY FG
AuthN
AuthZ
Access Policy Admin
KEM
TTP
IoT S. / Resource querying
VIRTUAL ENTITY FG
Virtual EntityRegistry
VE web i/f
VE endpoint
VE Manager
VE Repository
VE Broker
VE discovery
Lookup / binding (VE attr / IoT S.) / answers VE queries
IOT SERVICE FG
Meta-Cloud Data Endpoint
Data Manager
S. Data Repository
Data Broker
Data Retrieval
IoT Service/Resource Registry
IoT Service/ Resource web i/f
Resource Manager
Resource Broker
IoT Service & Resource S. Descr.
IoT Service / Resource discovery
COMMUNICATION FG
Subscription Manager
Message Bus (annotated Data) or "Topic" related
Publish. S. Service & Resource Descr.
Forwarded SPARQL req / Responses
DEVICE FG
S. Resource lookup & answer
S. Data Responses
Query Instant Data / Responses
FIESTA SIDE
TESTBED SIDE
Publish. S. Data
Publish S. Data
VE Endpoint
Data Endpoint
IoT Service Endpoint
Resource Manager w/ Semantic Data (FIESTA aligned)
RDF format / SSN light
VE Repository
S. Data Repository
S. Rec & IoT service descr. Repository
IoT Service Endpoint
VE w
S. Annotator eg json-> rdf
Resource manager w/o Semantic Data or w/ Sem. Data but NOT FIESTA aligned
E.g. JSON/JSON-LD
Data Repository
IoT Service Endpoint
S. Annotator
Resource Manager w/o Semantic Data & NO local data storage
Register VE / Binding VE attr. / IoT Service
Class-I Test-bed
Class-II Test-bed
Class-III Test-bed

Fig. 1. FIESTA-IoT System Architecture

4) IoT Service FG:

- **IoT Service/Resource Registry**: It provides an API for registering resources and their surrounding metadata (e.g. location, sensing capabilities, associated IoT services, etc.). This particular API can be used either for registering composite IoT services defined by the IoT Service Composer FC (Added-value Services in particular) or by testbeds which do not handle locally the definition of the IoT Services that expose their resources (See Class-II & -III Testbeds scenarios). This registry allows also to look-up IoT Services exposing resources based on various criteria (based on metadata). This registry is a federating one, meaning that it will forward requests to the testbed registries and compile/aggregate answers coming back from them, with additional constraints like for instance restricting the volume of answers.
- **Meta-Cloud Data Endpoint**: This component offers user interfaces to query data managed by FIESTA-IoT (either it

is locally stored and managed at the FIESTA-IoT side or remotely stored and managed by Class-I testbeds). It is about exclusive interaction between FIESTA-IoT user and stored data, meaning here annotated data coming from raw-data producers but also from knowledge producers. The Meta-Cloud Data Endpoint FC aims at managing and storing data published by the Class-II & -III testbeds and is the central point where data queries are resolved. When a data request comes, it will resolve it locally and also propagate the request to the Class-I testbed data endpoint (Class-I testbed data-endpoints are registered to the FIESTA-IoT platform beforehand).

5) Communication FG:

- **Message Bus**: This component brings about a communication channel following the Pub/Sub paradigm to the FIESTA-IoT ecosystem.

6) Security FG:

- **Authentication (AuthN)**. It is responsible for enforcing the authentication of registered FIESTA-IoT users.
- **Authorization (AuthZ)**. This component makes decisions about access control requests (intercepted at access decision points) based upon Access Control Policies (ACPs).

V. Architecture Information View

Based on the IM the Information View aims at providing details about how the information is actually coded, serialized and handled within the target IoT system.

Before starting with the explanation of the system use cases, it is worth highlighting a couple of assumptions that have been made to keep the figures as simple as possible.

- **Focus on core IoT Service FG components**: Experimenters might have different connection points to the FIESTA-IoT infrastructure; however, the Meta-Cloud Data Endpoint FC and the IoT Service Registry FC are the actual cornerstone of the whole platform. For the sake of simplicity, we will focus on the system use cases from these two FCs, disregarding the potential previous connections. In other words, the use cases in this section will only represent the sequence of messages between experimenters and these two core FCs, assuming that the intermediate elements would just forward the messages.
- **Security filter pre-condition**: The use cases described below will need a previous step for authentication and authorization of the requests described.

A. Security-based use-cases

1) Experiment registration/Identity management: In order for an experimenter to use the FIESTA-IoT services and the FIESTA-IoT testbeds, they must be known to FIESTA-IoT. FIESTA-IoT is the sole identity provider in the testbed federation i.e. experimenters register with FIESTA-IoT and provide FIESTA-IoT with the credentials (username, password) that will be used for authentication.

The experimenter signs-up through the FIESTA-IoT portal. In the logical FIESTA-IoT architecture, the AuthN component exposes the identity management functionality via the web portal. Based upon the information, that the experimenter fills in his/her registration form, FIESTA-IoT decides whether to allow the experimenter to register. Finally, the experimenter information is stored in the Member Database (part of the User Management FC).

2) Protected Resource Access: Resources in the FIESTA-IoT federation are protected and require authorization in order to be accessed by only FIESTA IoT experimenters. Each request to use one of the protected resources is checked in order to ensure that the request is from an authenticated experimenter, and that they are authorized to perform the request. Resource access in FIESTA-IoT follows a traditional Policy Enforcement Point (PEP) Pattern requests are intercepted by the PEP (at FIESTA-IoT endpoints) and these are sent to a Policy Decision Point (PDP) component, which forms part of the logical AuthZ component). The PDP implements the grant/deny decision when evaluating the request against the access policy.

B. Resource/IoT Service oriented use-cases

1) Testbed registers an IoT service/resource: It is worth highlighting that every Resource/IoT service must be associated with a semantic description aligned with FIESTA-IoT's ontology [5].

On the first hand, resources/IoT Services pertaining to Class-I testbeds are stored locally (testbed level) but the corresponding IoT Service/Resource endpoint needs to be registered in the FIESTA-IoT IoT Service Registry, indicating that subsequent resource searches will be brokered towards the Class-I testbed Resource Manager. Moreover, semantic descriptions of Resources/IoT Services pertaining to Class-II & -III testbeds need to be stored directly at the FIESTA-IoT side, thus replicating the info and fulfilling the semantic annotations according to the FIESTA-IoT ontology.

Testbeds will perform a Resource/IoT Service Registration request that will be addressed to the Resource/IoT Service Registry FC. Two possibilities arise here: as for Class-I testbeds, as they store their information locally, they will become an extension of the FIESTA-IoT meta-cloud (working as a distributed system). Thus, they will only register the IoT Service Endpoints. On the other hand, Class-II & -III testbeds, will have to semantically annotate their resources/IoT services descriptions and replicate them at the FIESTA-IoT level. This way, it is deemed necessary to include a Semantic Annotator entity at the testbed level. Once the registration is performed, the IoT Service Registry FC sends back an acknowledgment to the testbeds, confirming that everything has gone well or, otherwise, informing about any potential fault.

2) Experiment looks up resources/IoT Services (Discover): Probably, the first step an experimenter might take is to search the resources/services available in the FIESTA-IoT meta-testbed. If this search is resource-oriented, they will target the IoT Service/Resource Registry (directly or indirectly) which will be finally facilitating the discovery process. The experimenter generates a query that aims at retrieving a list of resources/IoT Services that comply with the requirements that shape such request. It is addressed to the Resource Broker sub-FC that will, in turn, forward it to the corresponding testbeds' Resource Managers, in case of having Class-I testbeds; besides, it will also look for matching resources/IoT Services on the central Triple-Store (for Class-II & -III testbeds). From the result set of the query execution in the corresponding repositories, a response is generated and sent back to the experimenter.

3) Experiment invokes IoT Services: From the results gathered in the previous use case (i.e. resource discovery), experimenters have now the information they need to start retrieving data from the resources that are exposed by their IoT Services. Assuming that the IoT Service endpoints are known, experimenters can invoke these services through these addresses and wait until the data is received.

For Class-I testbeds, the IoT Service Endpoint does deal with the incoming request locally at testbed level and sends back the response with all the metadata inside. Otherwise, for Class-II & -III testbeds the response is retrieved from the underlying resources but the format of the measurements needs to be then translated to the FIESTA-IoT semantic format, using the Semantic Annotator. The FIESTA-IoT Platform acts as an intermediary between experimenters and testbeds so, after receiving the request from the end-user at the IoT Service/Resource Registry's Broker sub-FC it will forward the query downwards to the corresponding testbed's IoT Service.

C. Data oriented use-cases

1) Testbed publishes semantically annotated Data to the FIESTA-IoT Message Bus: As has been already stated, Class-II & -III testbeds do not store semantically annotated data by themselves, so they do not provide a local Data Endpoint. Consequently, they will need to publish semantically enhanced data to the central repository so that it can be accessed by any third-parties, like experimenters, Knowledge Producers or (added-value) Service Providers. To achieve this, the Message Bus will play an essential role, acting as the intermediate entity between testbed and both FIESTA-IoT's Meta-Cloud Data Endpoint and experimenters.

Every time a physical resource generates an observation/measurement, a testbed Resource Manager sends a message (containing the annotated description of the measurement) towards FIESTA-IoT's Message Bus. Once the Message Bus gets the information, it stores it at the Meta Cloud Data Endpoint.

2) Experiment queries/retrieves Data: FIESTA-IoT platform enables testbed-agnostic access to data. However, while the queries will not depend on the testbed that is originating the data (in fact, data can come from many of them after the same request), the way in which the platform retrieves the data depends on the testbed class.

a) Data originating from Class-I testbed. Class-I testbeds can be seen as an extension of the FIESTA-IoT platform, their joint operation will work like a distributed storage system. FIESTA-IoT platform will play the role of a broker, hiding the underlying operation to end-users and forwarding the queries/responses. A data query is sent by the experimenter towards the Meta-Cloud Data Endpoint's Resource Broker. The query is forwarded to underlying Class-I testbeds' Data Endpoint. Then, the query against the testbed's Semantic Data Repository is performed. If the information is there, the Resource Manager will gather the data and compose a response message, which will be addressed back to the experimenter, using again FIESTA-IoT as a relaying actor.

b) Data originating from Class-II or Class-III testbed: As already discussed, all data generated by Class-II & -III testbeds is semantically annotated and replicated at the FIESTA-IoT level (namely in the Meta-Cloud Data Endpoint's Semantic Data Repository). The experimenter generates a data query and sends it to the Meta-Cloud Data Endpoint's Resource Broker. Then, the message will get the Data Manager, which extracts the raw query snippet from the body of the message and the query is carried out in the Semantic Data Repository (SDR). The response of this query will be sent back to the experimenter.

VI. Conclusions

The FIESTA-IoT architecture described in this paper aims mainly at presenting the basic foundations for the FIESTA-IoT platform, which should be afterwards complemented with concrete interfaces, information about ontologies, deployment views, etc. However, it still offers a one-stop-shop document for whoever wants to delve quickly into the FIESTA-IoT topic; as a consequence, it prevents people from looking up several more specialized documents in order to fetch essential information.

The architecture presented in this paper offers a synthetic and abstract view of the FIESTA-IoT platform. It is inclusive in the sense that it can accommodate under its federation a large number of testbeds with heterogeneous capabilities (some being semantic-enabled already, some not). It offers full semantic interoperability: all assets of the testbed (resources and IoT Services) are semantically annotated and described; they are discoverable using either powerful data query languages or simpler APIs. FIESTA-IoT is therefore able to offer the greatest testbed agnostic experience to both expert users (semantically skilled) and more basic experimenters as well.

Acknowledgement

This work is partially funded by the European project Federated Interoperable Semantic IoT/cloud Testbeds and Applications (FIESTA-IoT) from the European Union's Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943. The authors would like to thank the FIESTA-IoT consortium for the fruitful discussions.

References

[1] IoT-A Consortium, "Deliverable D1.5 Final architectural reference model for the IoT v3.0," Tech. Rep., 2013.

[2] S. Haller, A. Serbanati, M. Bauer, and F. Carrez, "A Domain Model for the Internet of Things," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 411–417.

[3] N. Rozanski and E. Woods, "Applying viewpoints and views to software architecture," *Open University White Paper*, 2005.

[4] FIESTA-IoT Consortium, "Deliverable D2.1 - Stakeholders Requirements." Tech. Rep., 2015.

[5] R. Agarwal, D. Gómez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sánchez, N. Georgantas, and V. Issarny, "Unified IoT ontology to enable interoperability and federation of testbeds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016, pp. 70–75.