CONTEXT / DATA MANAGEMENT,
CLOUD HOSTING, SECURITY & I2ND

Gradiant

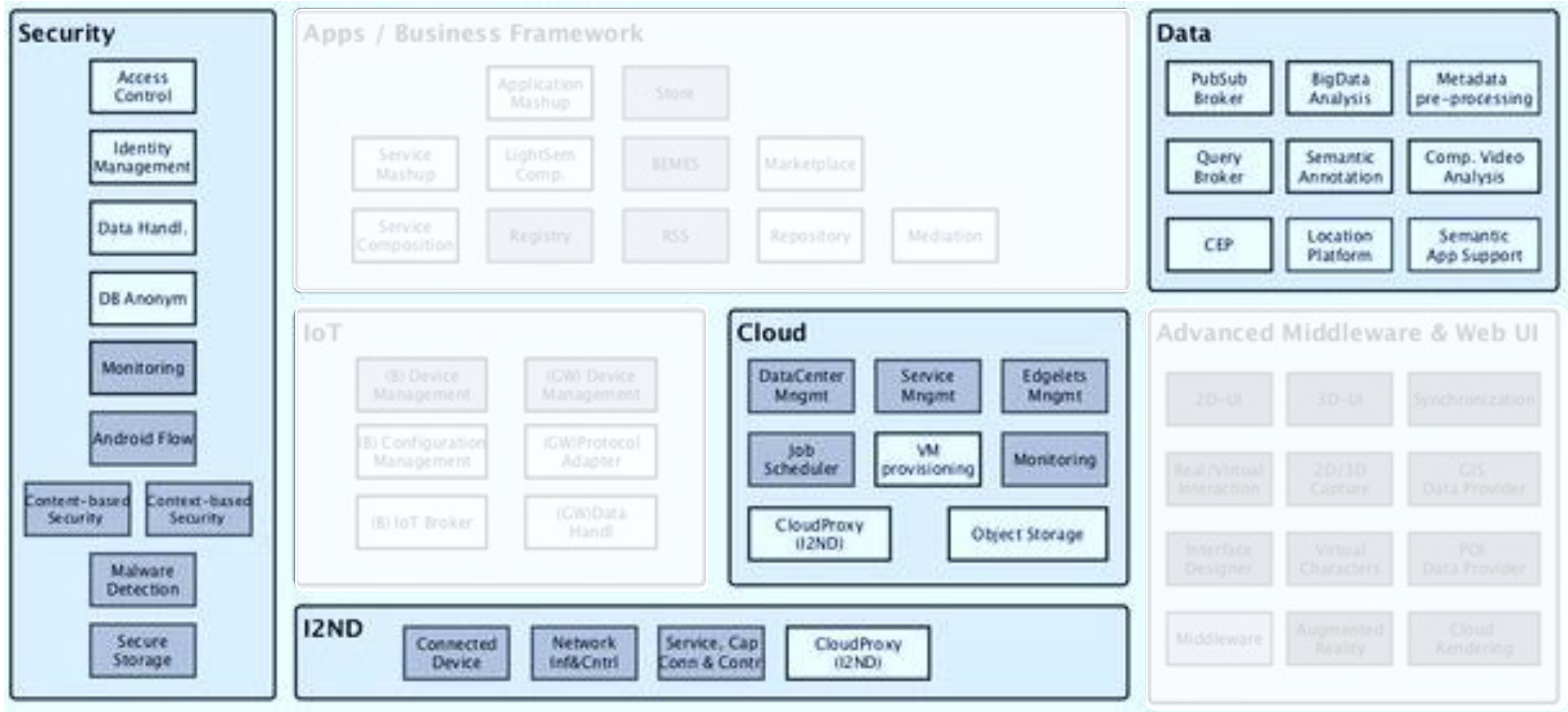
Galician research and development center
in advanced telecommunications

2015

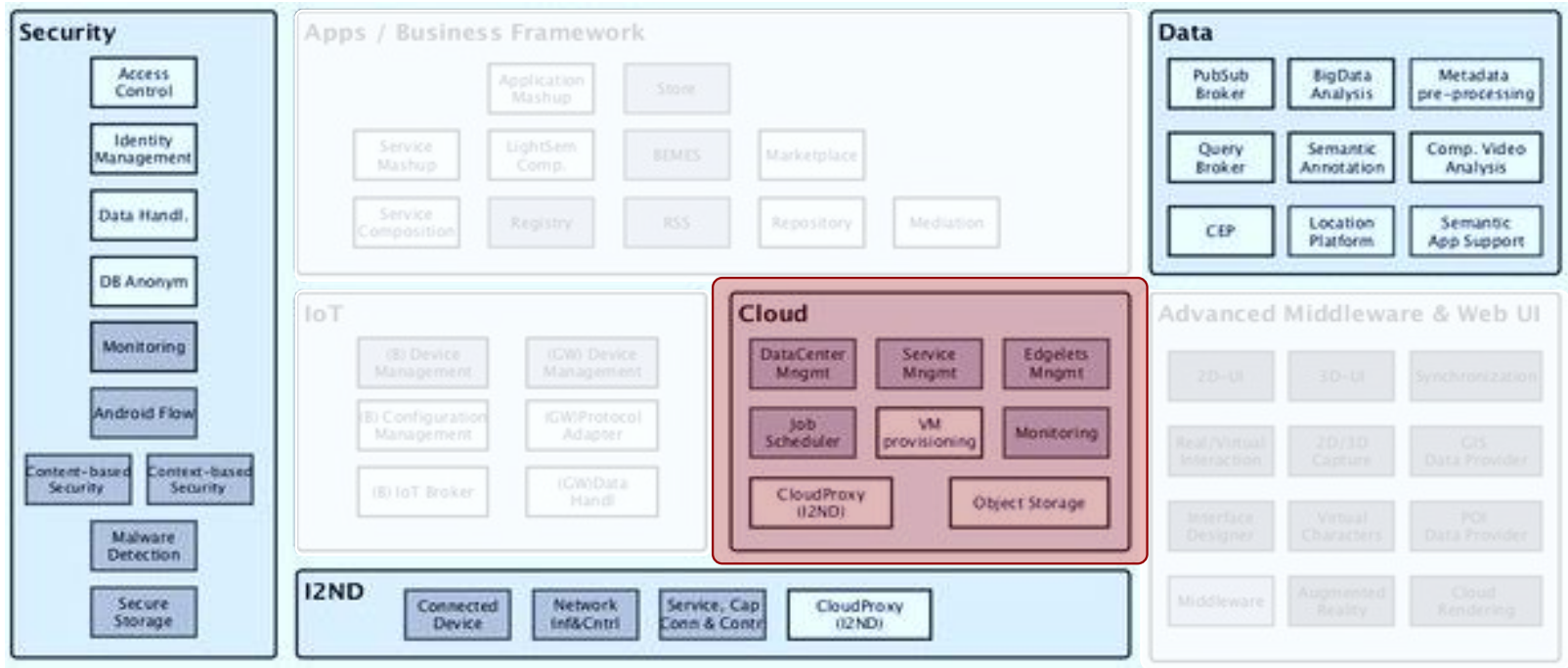
www.gradiant.org

Gradiant

Architecture Overview



CLOUD HOSTING



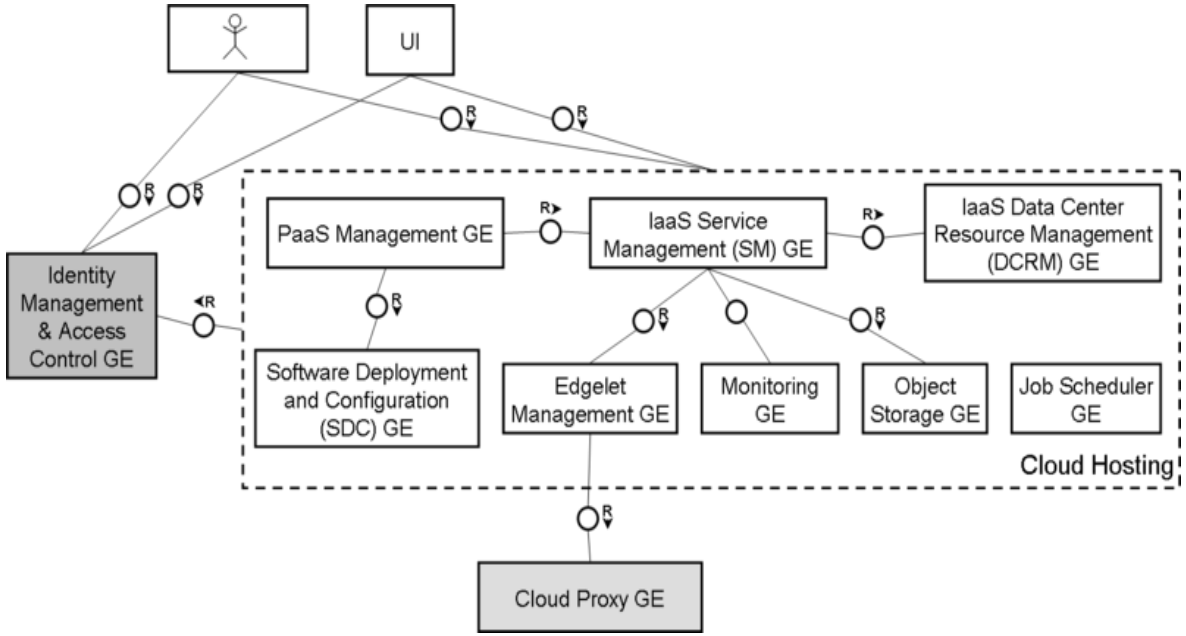
Architecture Overview

The **Cloud Hosting** module offers Generic Enablers (GEs) for designing a cloud infrastructure that can be used to develop, deploy and manage Future Internet applications and services.

FI-WARE Cloud module contains the following GEs:

- **IaaS Data Center Resource Management (DCRM) GE:** offers provisioning and life cycle management of virtualized resources (compute, storage, network) associated with virtual machines.
- **Object Storage GE:** offers provisioning and management of object-based storage containers and elements.
- **Job Scheduler GE:** offers the application to submit and manage computational jobs in a unified and scalable manner.

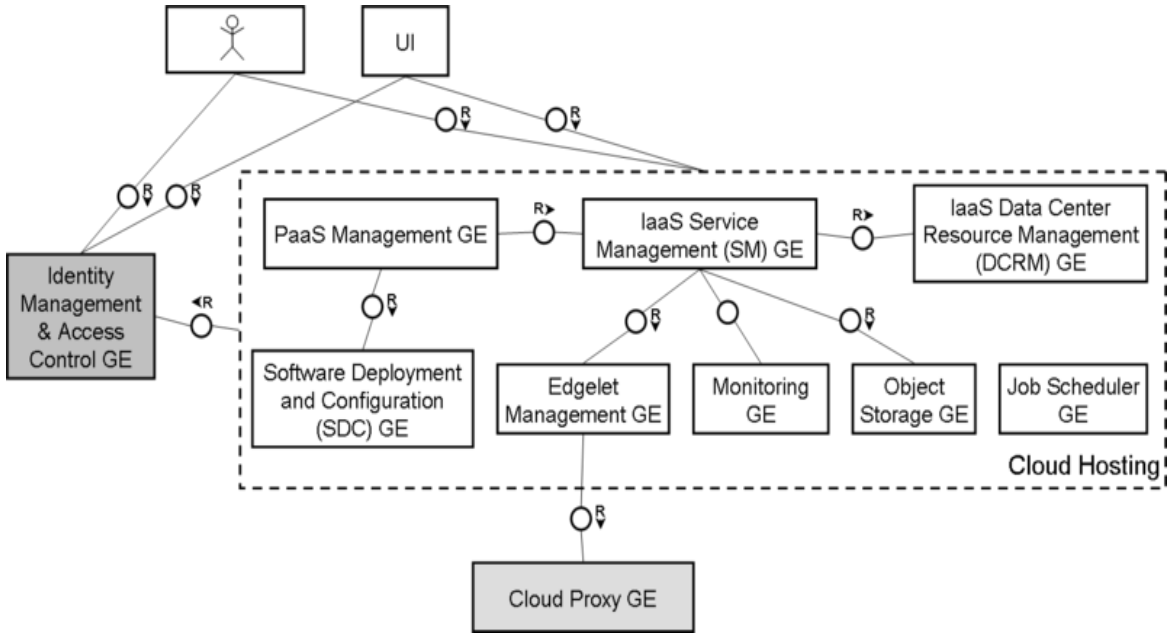
(continue …)



Architecture Overview - II

(… continue)

- **Edgelet Management GE:** offers the capability to host lightweight application components, called edgelets.
- **IaaS Service Management (SM) GE:** provides the means to host complex applications.
- **PaaS Management GE:** offers provisioning and management of complete PaaS environments.
- **Software Deployment and Configuration (SDC) GE:** offers a flexible framework for installation and customization of software products (via Chef recipes) within individual virtual machines.



IaaS Resource Management GE - DCRM

FIWARE
IaaS Resource Management

The **IaaS Resource Management GE** or **DataCenter Resource Management (DCRM) GE** provides the basic Virtual Machine (VM) hosting capabilities, as well as management of the corresponding cloud resources within a particular FI-WARE Cloud Instance.

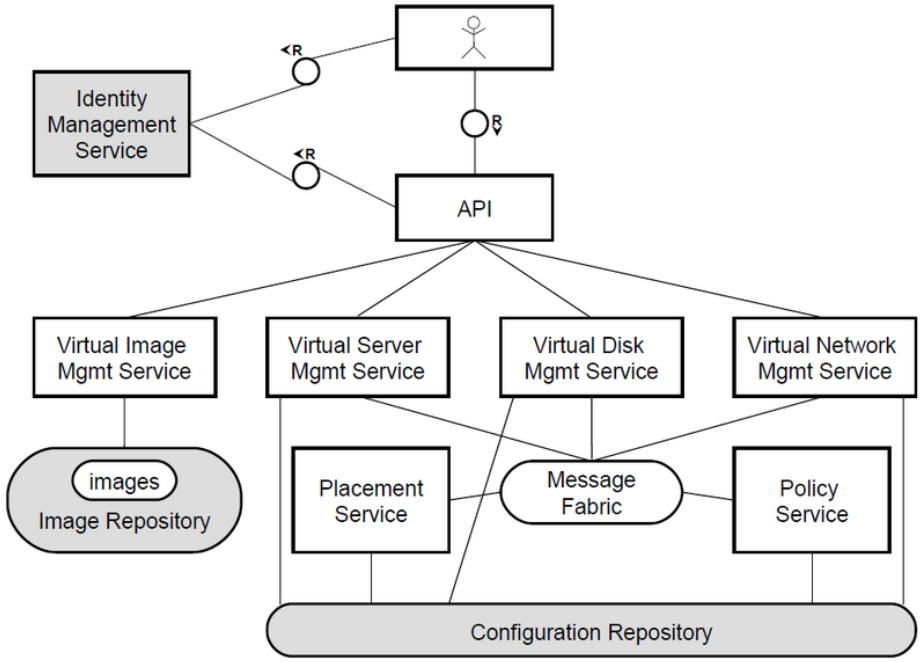
The main capabilities of this GE are:

- Provision of a VM with a specified VM image
- Manage network and storage of the VM
- Resource monitoring of the VM
- Resiliency of the persistent data associated with the VM
- Manage resource allocation (with guarantees)
- Secure access to the VM
- Resource optimization
- Capacity management and admission control
- Multi-tenancy (isolation between VMs of different accounts)
- Automation of typical admin tasks

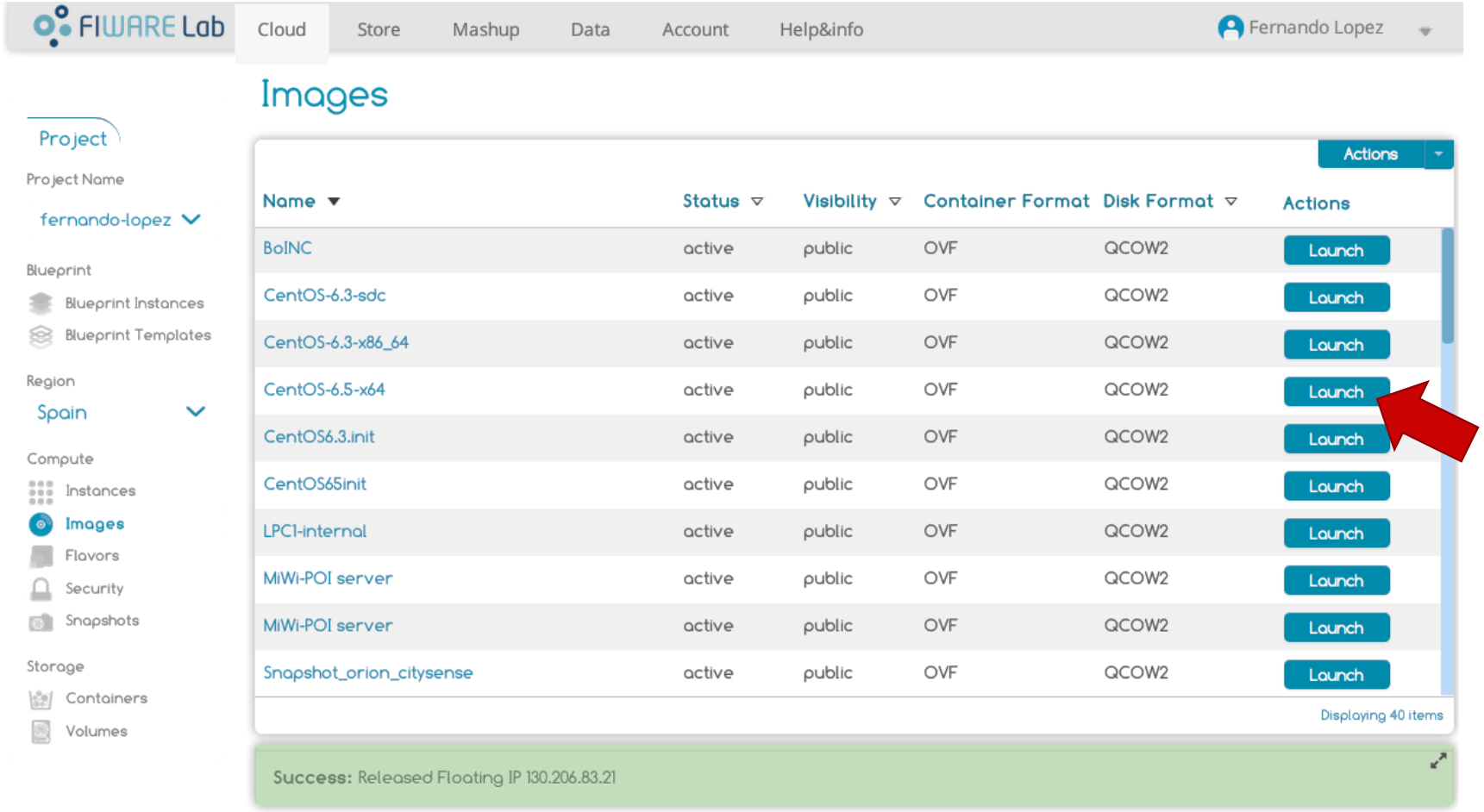
IaaS Resource Management GE – DCRM Provisioning

DCRM GE supports provisioning and management of:

- **Virtual servers**: Virtualized containers that can host an arbitrary Operating System and arbitrary software stack on top, installed within the virtual server.
- **Virtual disks:** Persistent virtual disks that can be potentially attached to an arbitrary virtual server.
- **Virtual networks:** Logical networks abstraction that would typically represent a network segment at layer 2 of the OSI model.
- **Virtual images :** An image is a collection of packaged files used to create or rebuild a virtual server. Basically, a virtual image is a snapshot of a virtual server.



IaaS GE – Launch instance

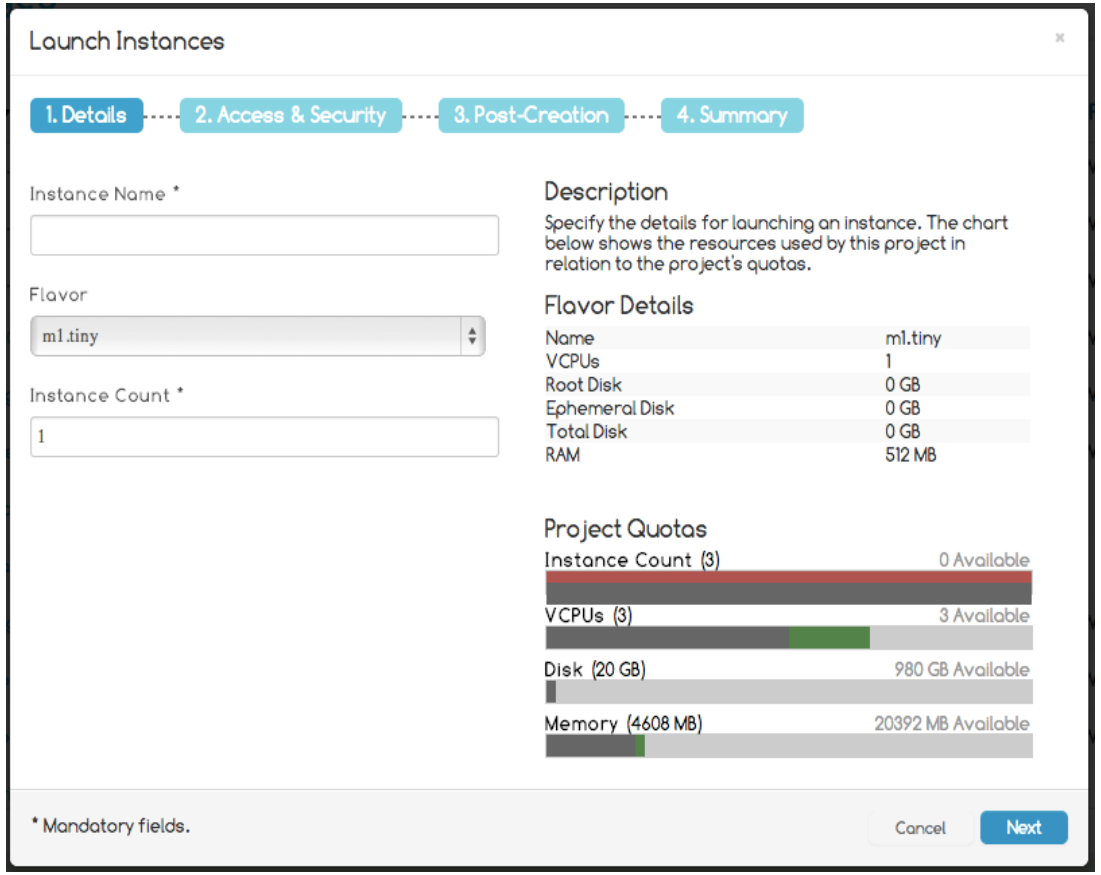


IaaS GE – Launch instance II

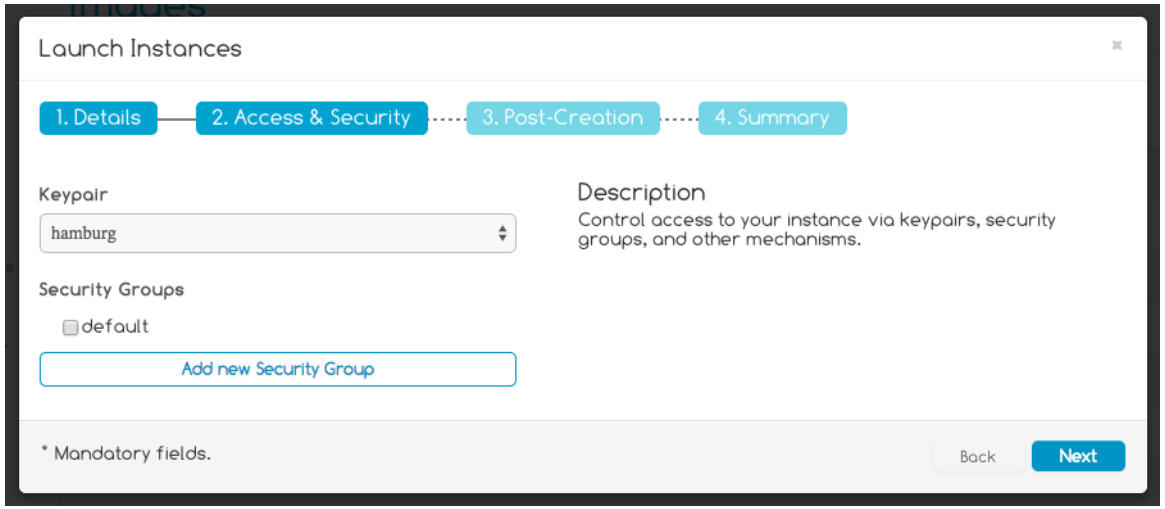
Instance definition

The first step of the launching process is the instance definition.

The user must give a name to the instance and select a flavor with the number of VCPUs, disk capacity and memory requirements.



IaaS GE – Launch instance III

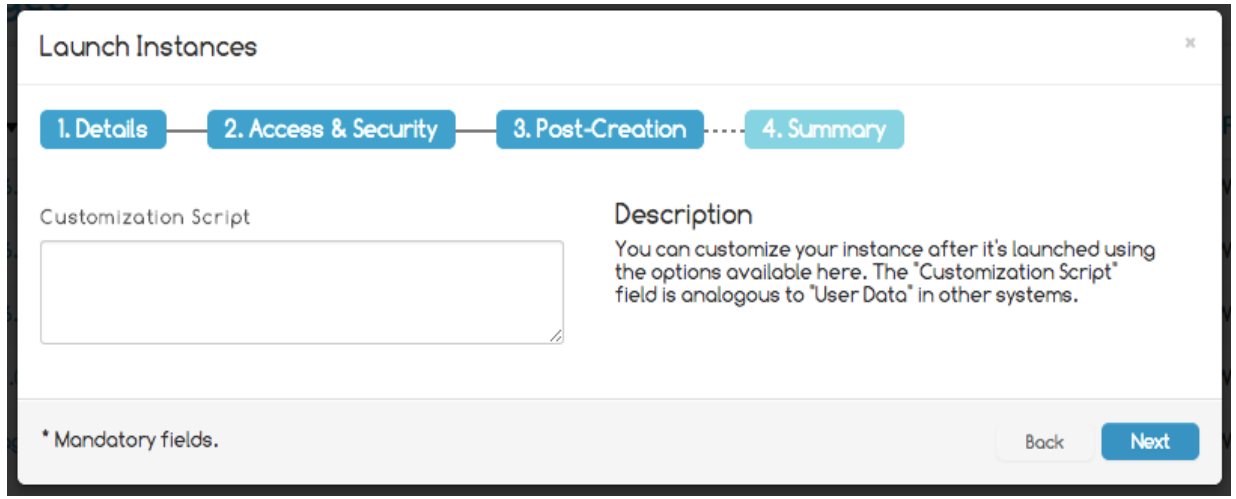


Instance security

Then the user must establish the access control and security requirements of the instance by selecting a ***keypair*** and the ***security groups*** allowed.

Instance customization

The users can customize their instances after they are launched using a "***Customization Script***"

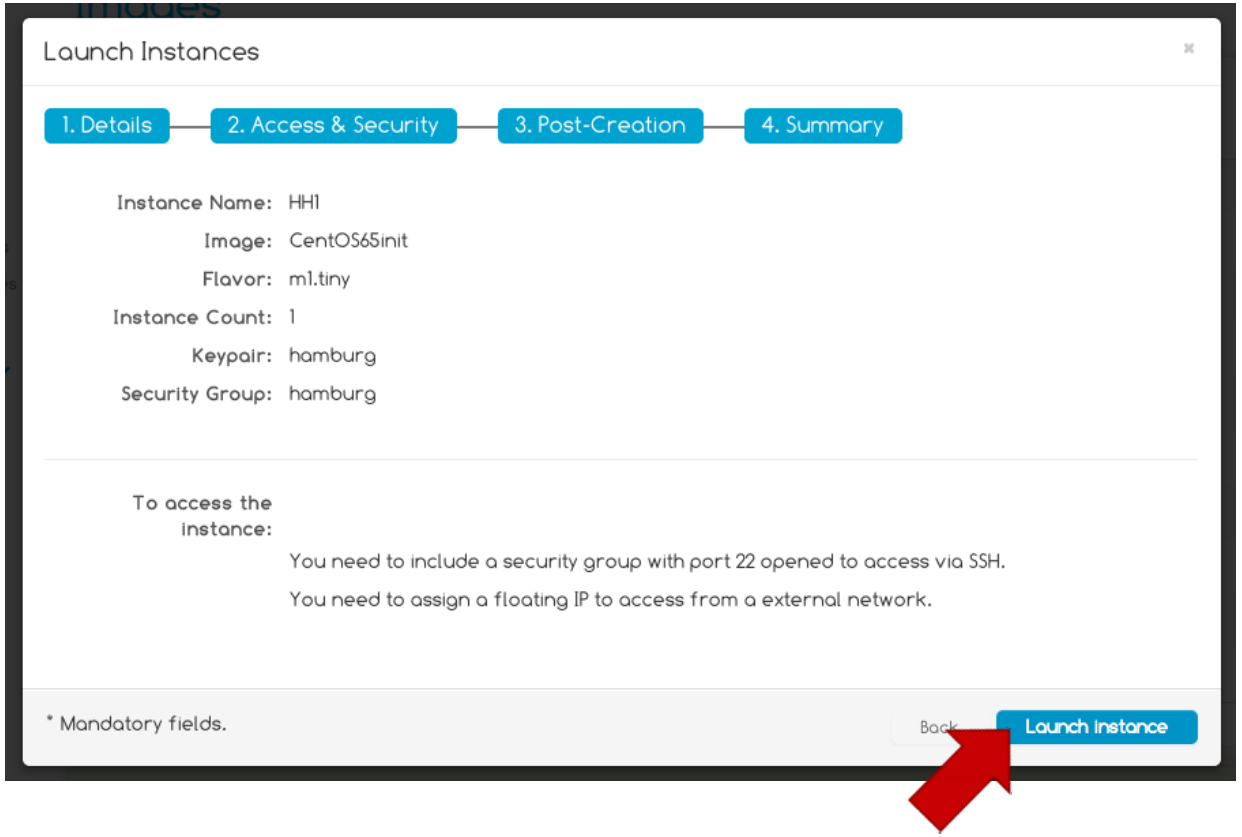


IaaS GE – Launch instance IV

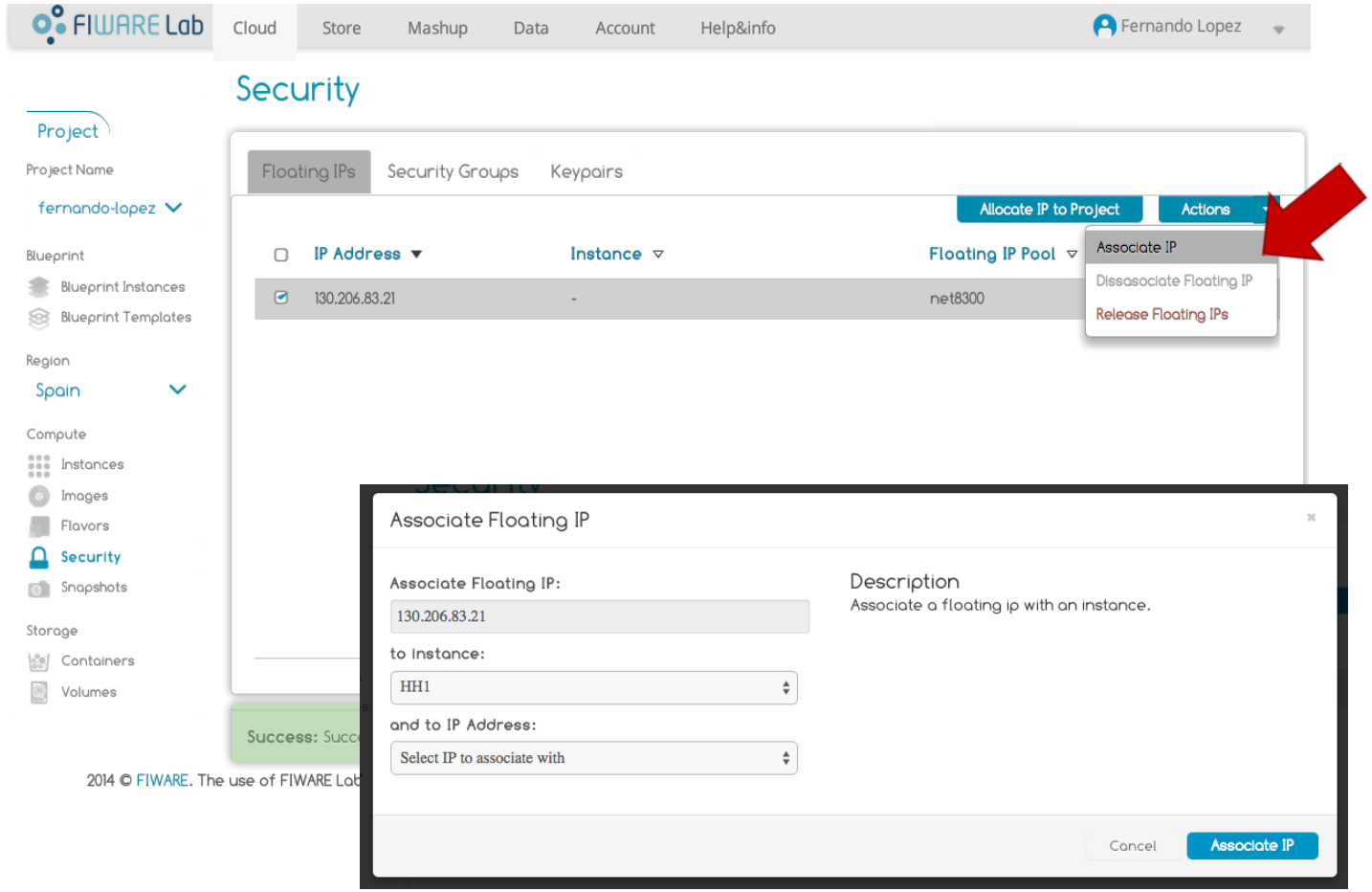
Instance launching

When the instance creation is complete, a summary page will show its main characteristics and features.

Finally, to launch the instance, just click on “Launch instance”



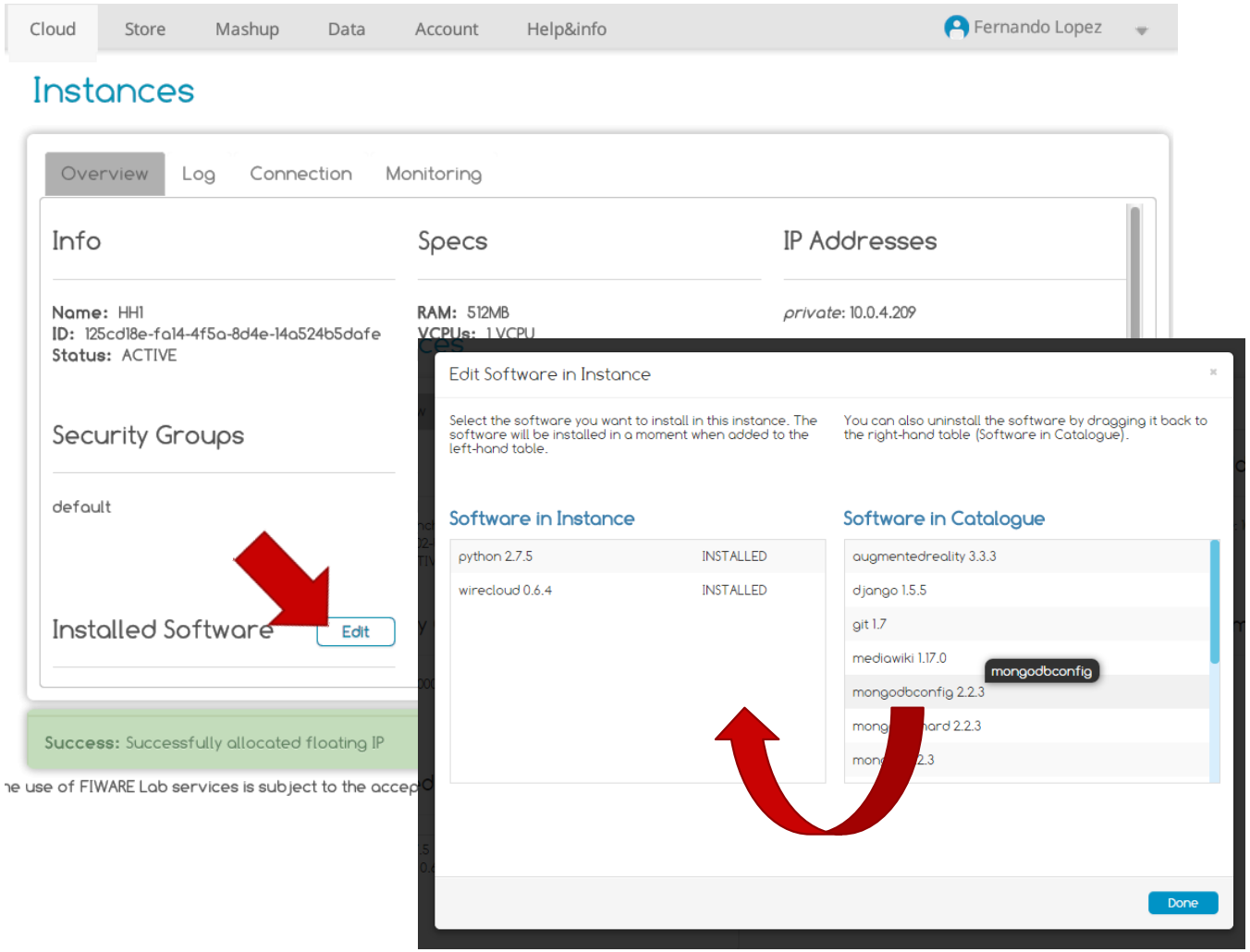
IaaS GE – Allocate Floating IP



Floating IP allocation is needed to access the instance from an external network.

A floating IP is a public IP that can be attached to virtual servers in the Cloud infrastructure.

IaaS GE – Instance Software



An instance may required some concrete pieces of software to run properly: servers, repositories, data bases, user interfaces, etc.

To install new software on the created instance, the users must select the “edit” button of the “Installed software” section at the instance overview.

Software is added to the instance simply by dragging it from the Catalogue into the user’s instance.

IaaS Resource Management GE – CLI

Virtual Images

- **listVirtualImages** -- Returns a list of all available virtual images
- **queryVirtualImages** -- Returns a filtered list of available virtual images
- **getVirtualImageDetails** -- Returns details of a virtual image
- **uploadVirtualImage** -- Uploads a new virtual image into the repository

Virtual Servers

- **createVirtualServer** -- Provisions a new virtual server with the given properties (virtual hardware, policy parameters, access, etc).
- **destroyVirtualServer** -- Removes a virtual server
- **powerOnVirtualServer** -- Powers on a virtual server
- **powerOffVirtualServer** -- Powers off a virtual server
- **restartVirtualServer** -- Restarts a virtual server
- **shutdownVirtualServer** -- Shuts down a virtual server
- **resizeVirtualServer** -- Changes the virtual hardware allocation for a virtual server, e.g., allocated RAM or number of CPUs
- **getVirtualServerDetails** -- Returns details of a virtual server (virtual hardware specification, state, associated policy parameters, access details, etc.)

Virtual disks

- **createVirtualDisk** -- Provisions a new virtual disk with the given properties (size, capabilities, etc.).
- **destroyVirtualDisk** -- Removes a virtual disk
- **attachVirtualDisk** -- Attaches a given virtual disk to a given virtual server
- **detachVirtualDisk** -- Detaches a given virtual disk from a given virtual server
- **getVirtualDiskDetails** -- Returns details of a given virtual disk (capabilities, attachment details, etc.)

Virtual networks

- **createVirtualNetwork** -- Provisions a new virtual network with the given properties (e.g., VLAN ID, capabilities, etc.).
- **destroyVirtualNetwork** -- Removes a virtual network
- **attachVirtualServerToNetwork** -- Attaches a virtual network interface of a given virtual server to a given virtual network
- **detachVirtualServerFromNetwork** -- Detaches a virtual network interface of a given virtual server from a given virtual network
- **getVirtualNetworkDetails** -- Returns details of a given virtual network (ID, capabilities, attachment details, etc.)

PaaS Manager - Pegasus

Pegasus
PaaS Manager

The **PaaS Manager GE** provides a new layer to reduce the task of deploying applications on a Cloud infrastructure.

The PaaS Manager manages the provisioning of the required virtual resources, and the installation and configuration of elastic architectures based on load balancers and software tiers.

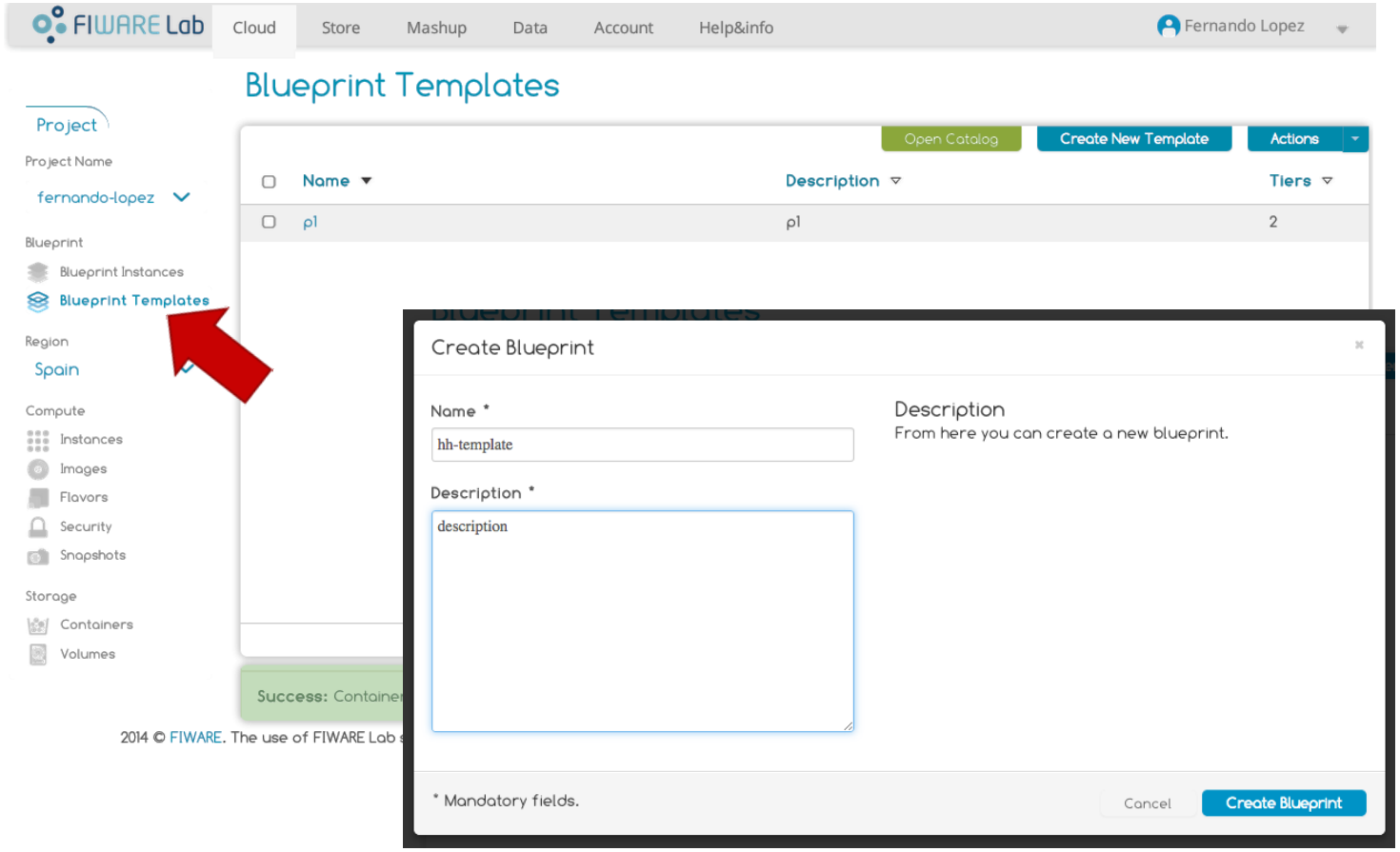
The PaaS Manager GE interacts with the IaaS SM GE for the deployment of the hardware infrastructure (virtual machines or servers and networks) and with the SDC GE for the installation and configuration of the software.

The main functionalities that the PaaS Manager provides are:

- **Management of Application Environments**, which involves the provisioning and configuration of IaaS resources, and installation, configuration and management of the Products Instances required for the application components to be deployed.

- **Management of Application Components (ACs)** (lifecycle and configuration) with the help of SDC GE for the installation and configuration of ACs.

PaaS GE – Blueprint templates



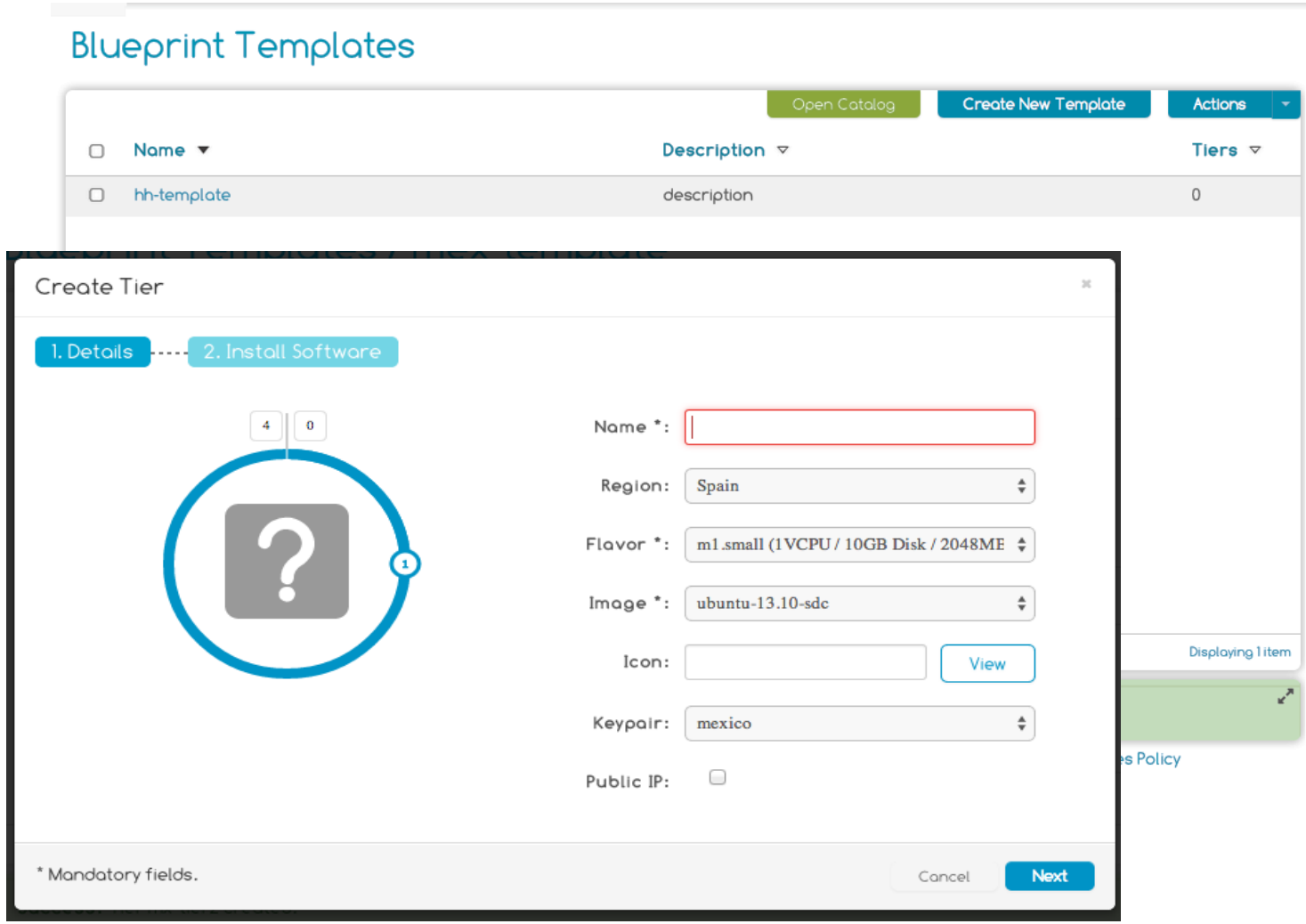
Applications normally are not deployed just in a unique VM but they are composed by a set of layers (DB, web server..). A **blueprint template** is the specification of the set of VMs plus the software to be installed on top of.

The user can define blueprint templates from scratch, divide a service into different tiers and store them in a catalogue where later other users reuse and customize them.

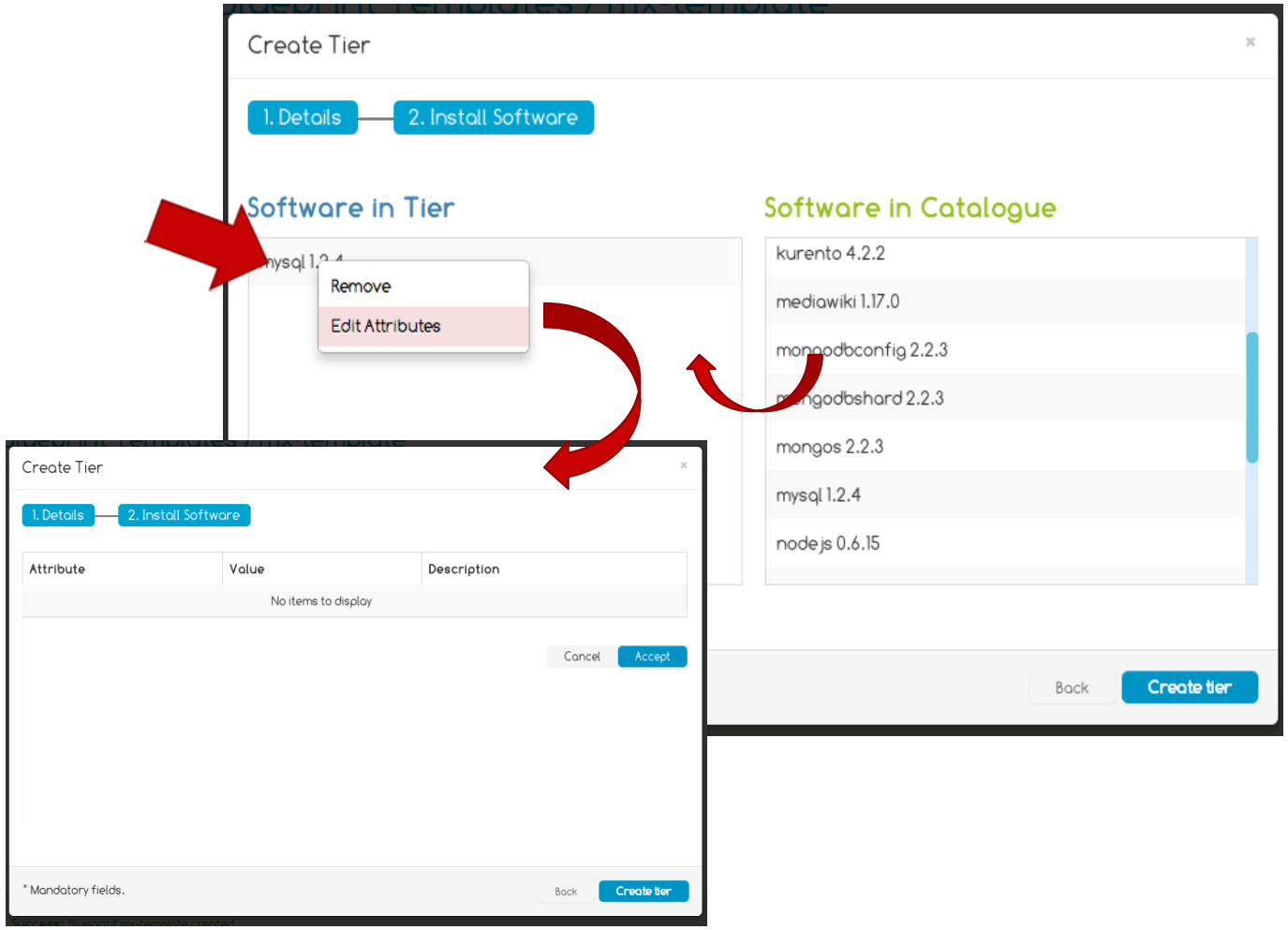
PaaS GE – Adding tiers

Each template can contain one or more tiers with different images of VM or/and different flavors.

To add a new tier, the user must choose a blueprint template and select “Create tier” on the “actions” menu.



PaaS GE – Add software to tier



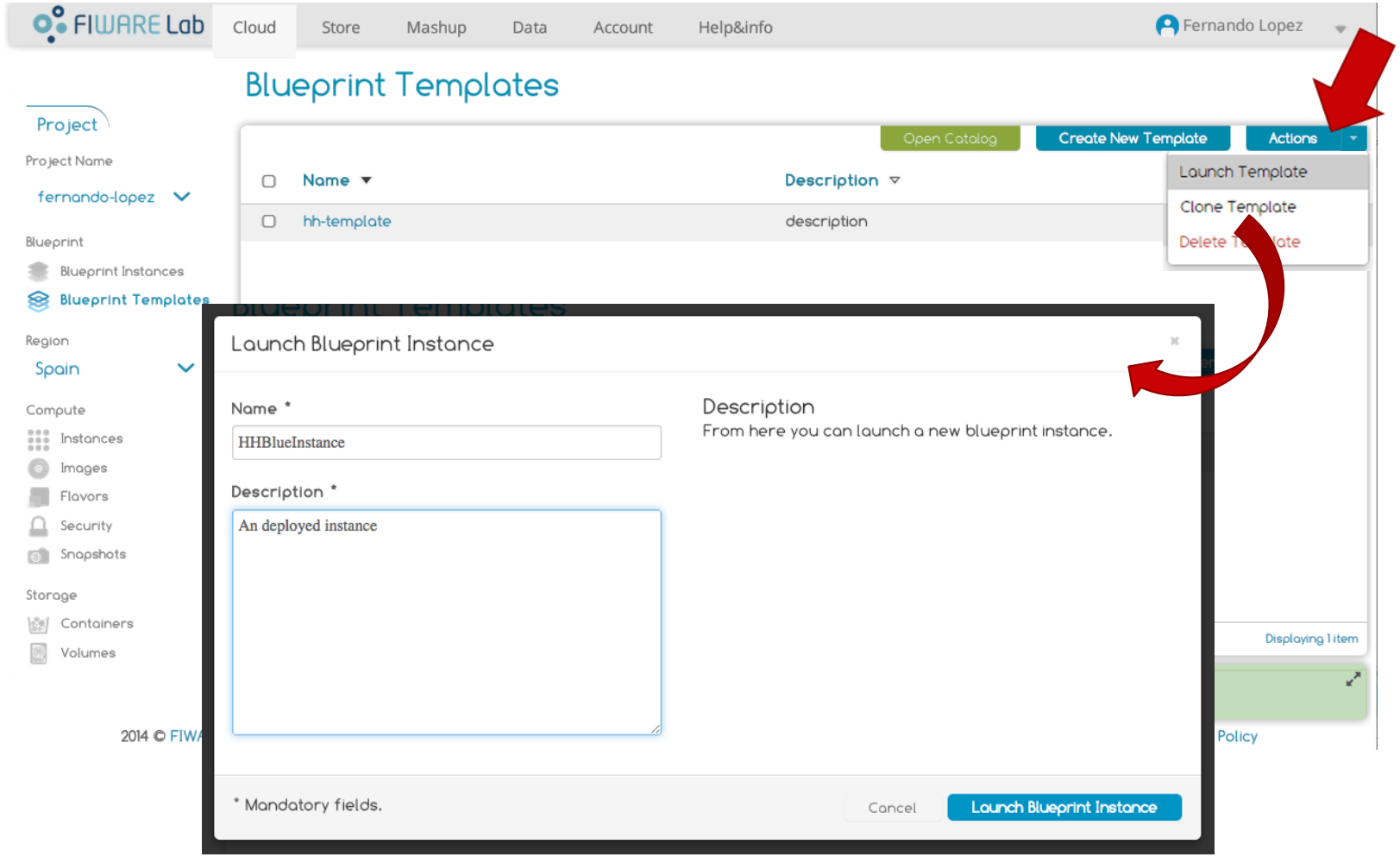
Users can easily install the software they need (such as Apache Tomcat, MySQL, or HAProxy) into a tier of the blueprint template just picking up the piece of software from the available catalogue and dropping it into the tier.

The platform will ensure that each resource is properly configured, and provides a safe, flexible, easily-repeatable mechanism that ensures all the nodes are always running exactly the way they were expected.

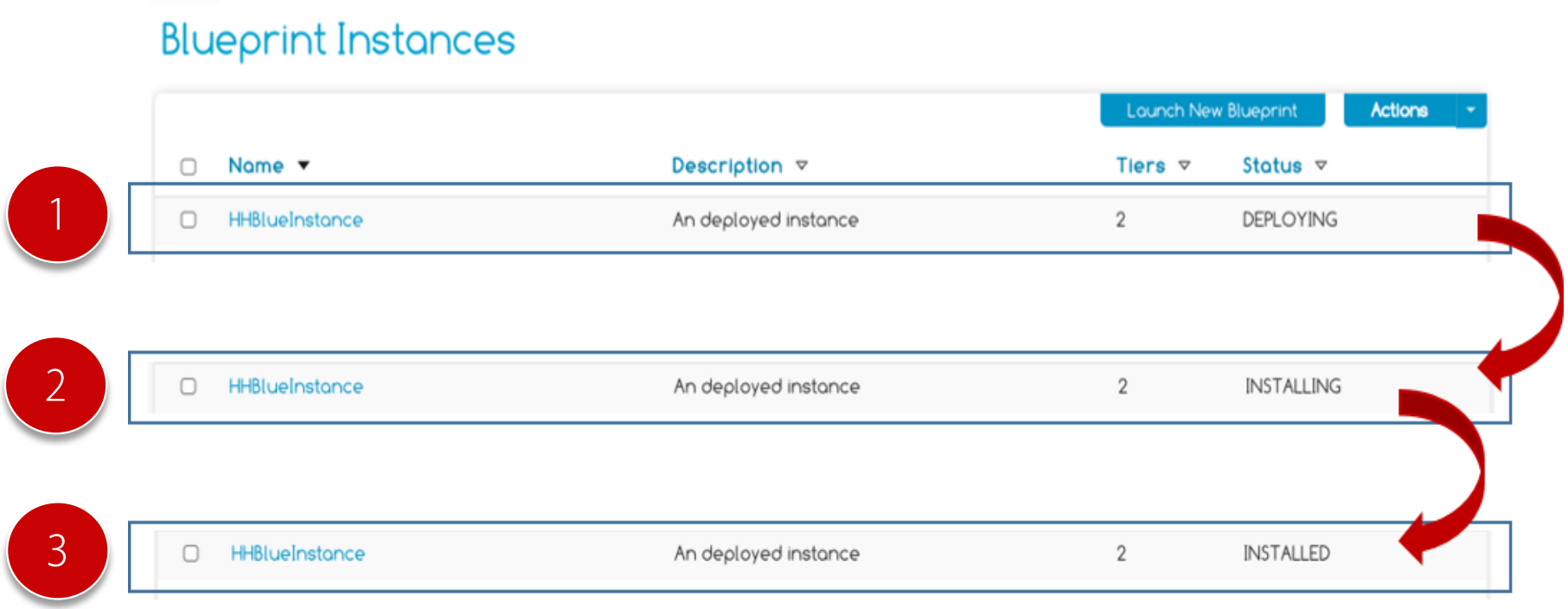
PaaS GE – From Template to Instance

To create a new instance of a concrete blueprint template, the user just have to select it from the presented list and choose the "Launch template" action.

The tier structure of the template will then be deployed and all the software pieces installed.

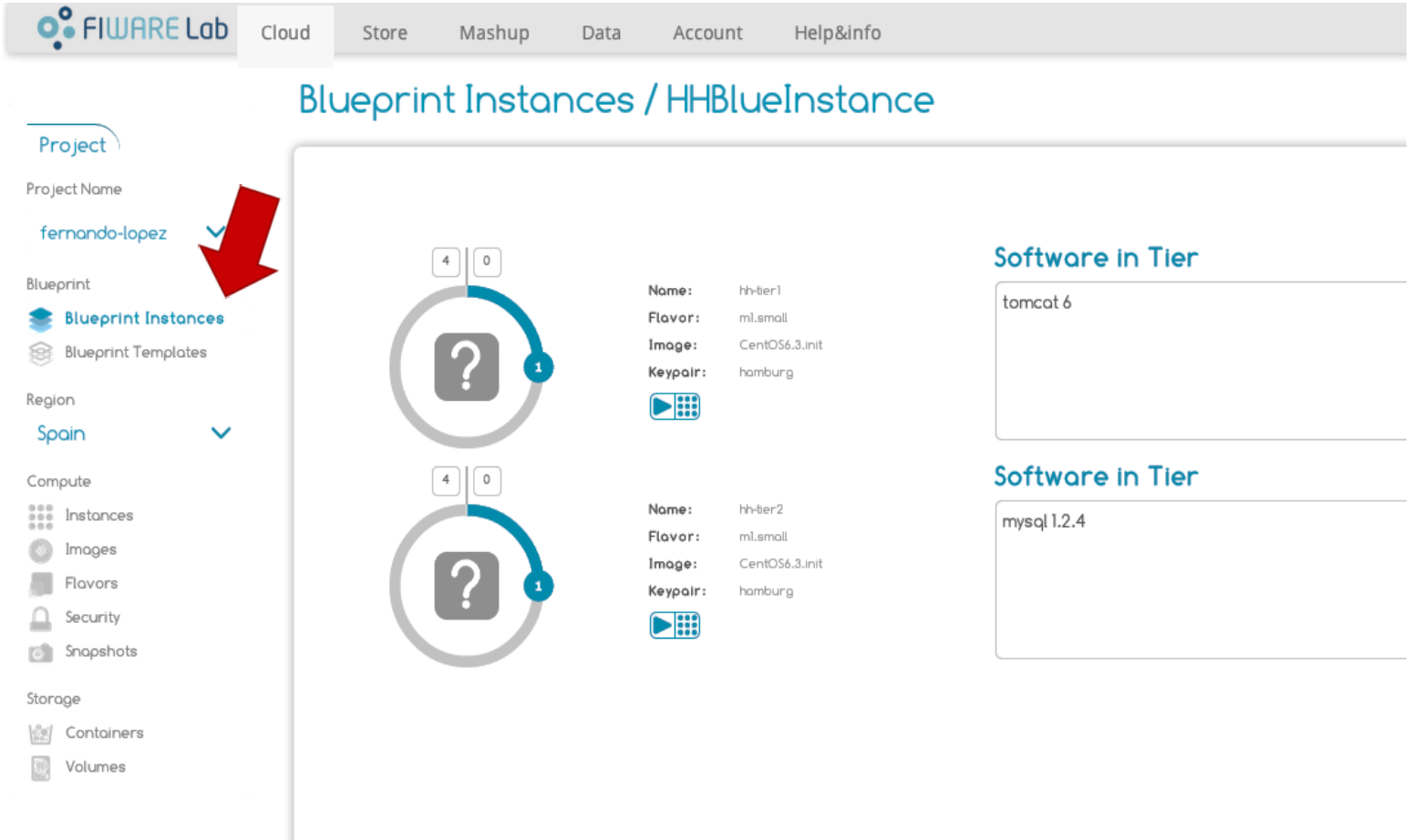


PaaS GE – Instance launch process



PaaS GE – Blueprint Instances visualization

After the blueprint launching process is complete, the user can check the structure (tiers and installed software) of the available blueprint instances in the “blueprint Instances” section.



Object Storage GE - FIWARE Implementation

The **Object Storage Generic Enabler** is a back-end component that provides object storage capabilities which software developers can incorporate into their applications.

FIWARE
Object Storage

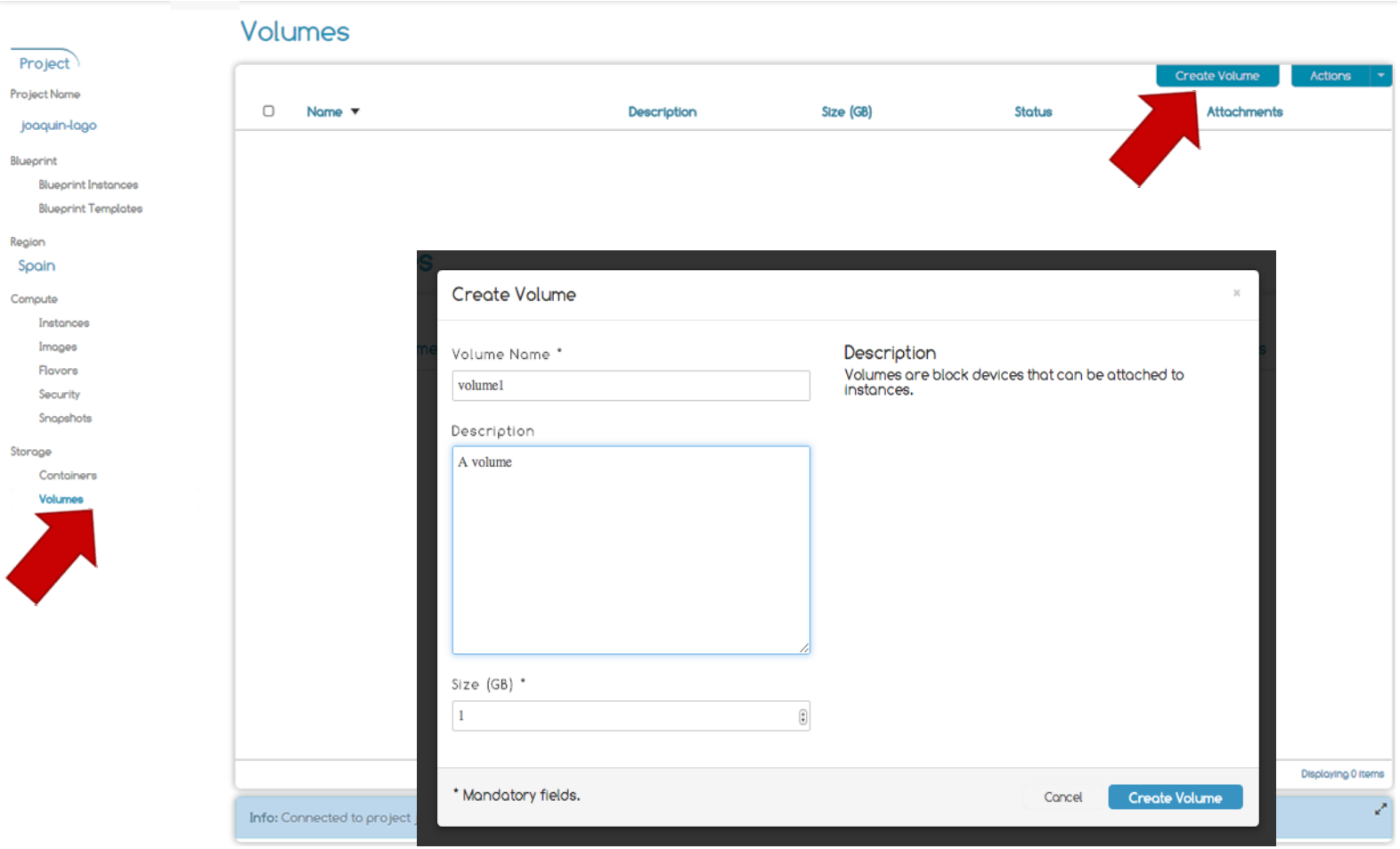
This GE implementation provides robust, scalable object storage functionality based on **OpenStack Swift**. The OpenStack Swift API offers a standardised mechanism to manipulate both the binary objects that are stored, and the hierarchy of containers in which they are organised.

The Object Storage Generic Enabler exposes Object Storage functionality via a standard API: **Cloud Data Management Interface (CDMI).**

At the core of CDMI are the basic management operations of Create, Retrieve, Update and Delete:

- Enables clients to discover capabilities of the object storage offering
- Manage containers and the objects that are placed within them
- Assigns and manipulates metadata to containers and objects

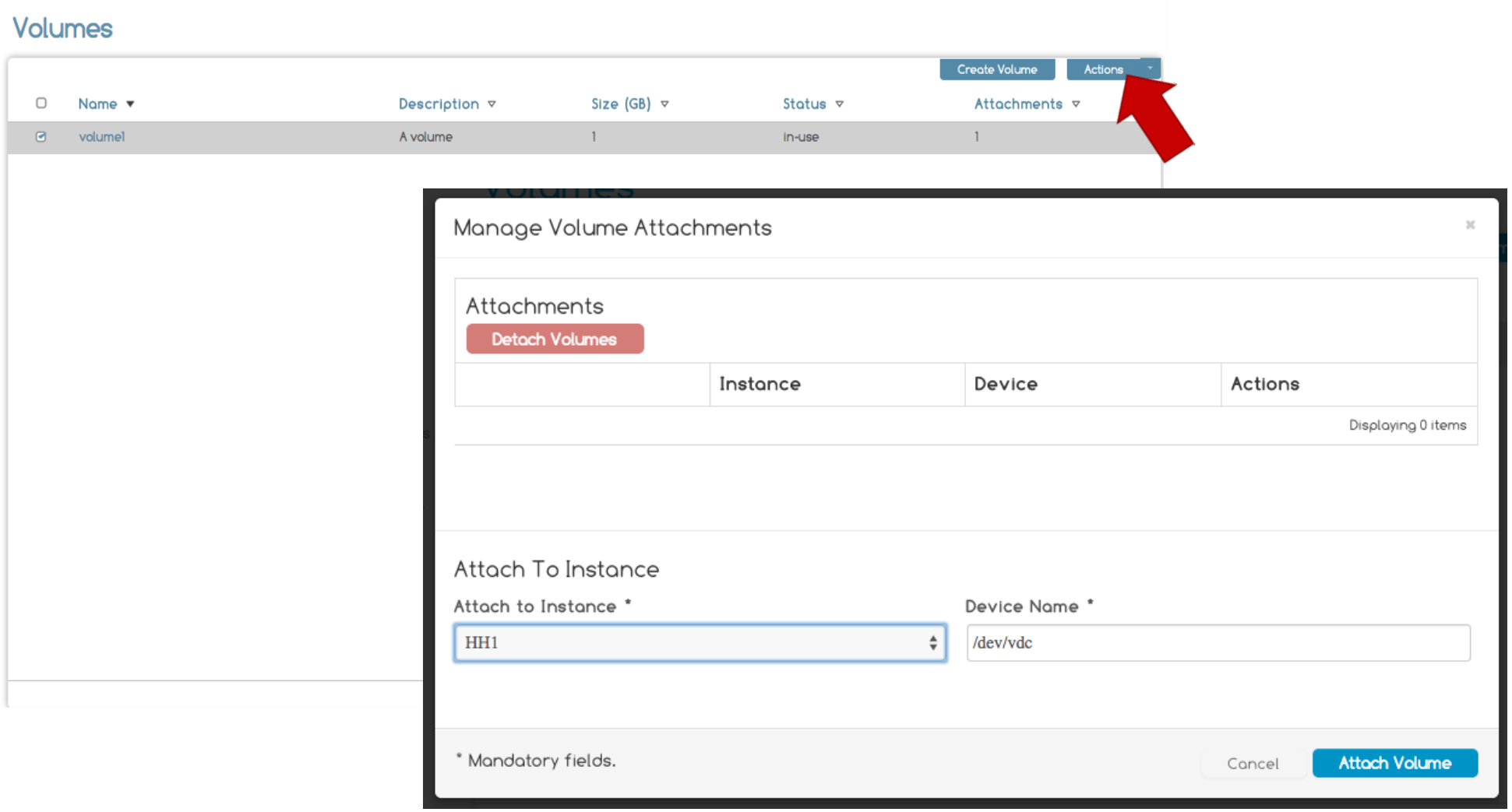
Object Storage GE – Volume creation



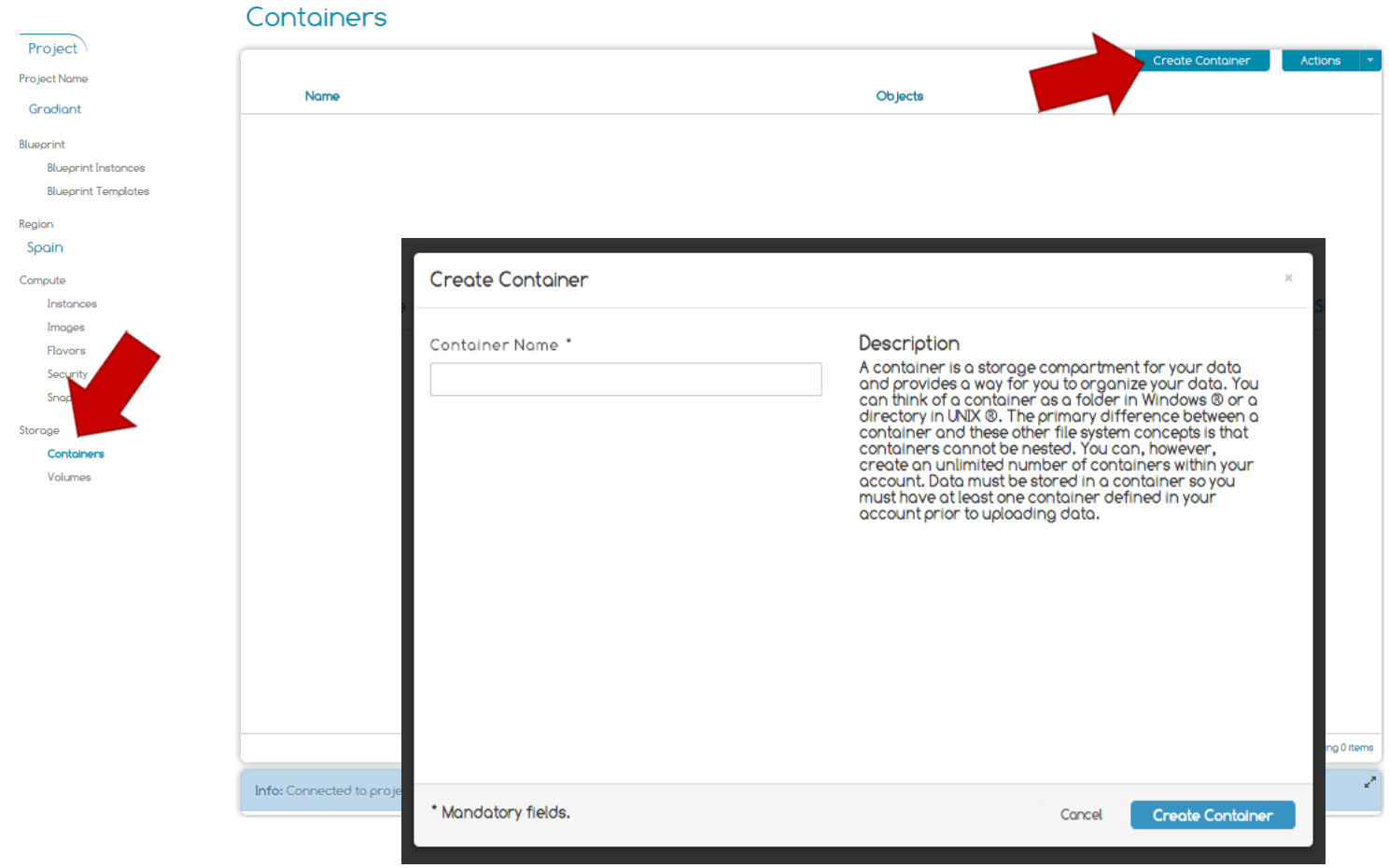
A **Volume** allows to associate storage capacity to an specific instance and provide persistence storage associated to a specific virtual machine.

In the section “Volumes” users can retrieve information about the available volumes and associate a volume to a specific virtual machine instance.

Object Storage GE – Volume attachment



Object Storage GE – Containers



Users can create and manage an appropriate hierarchy of containers in which to organize the binary objects they store in the Object Storage GE.

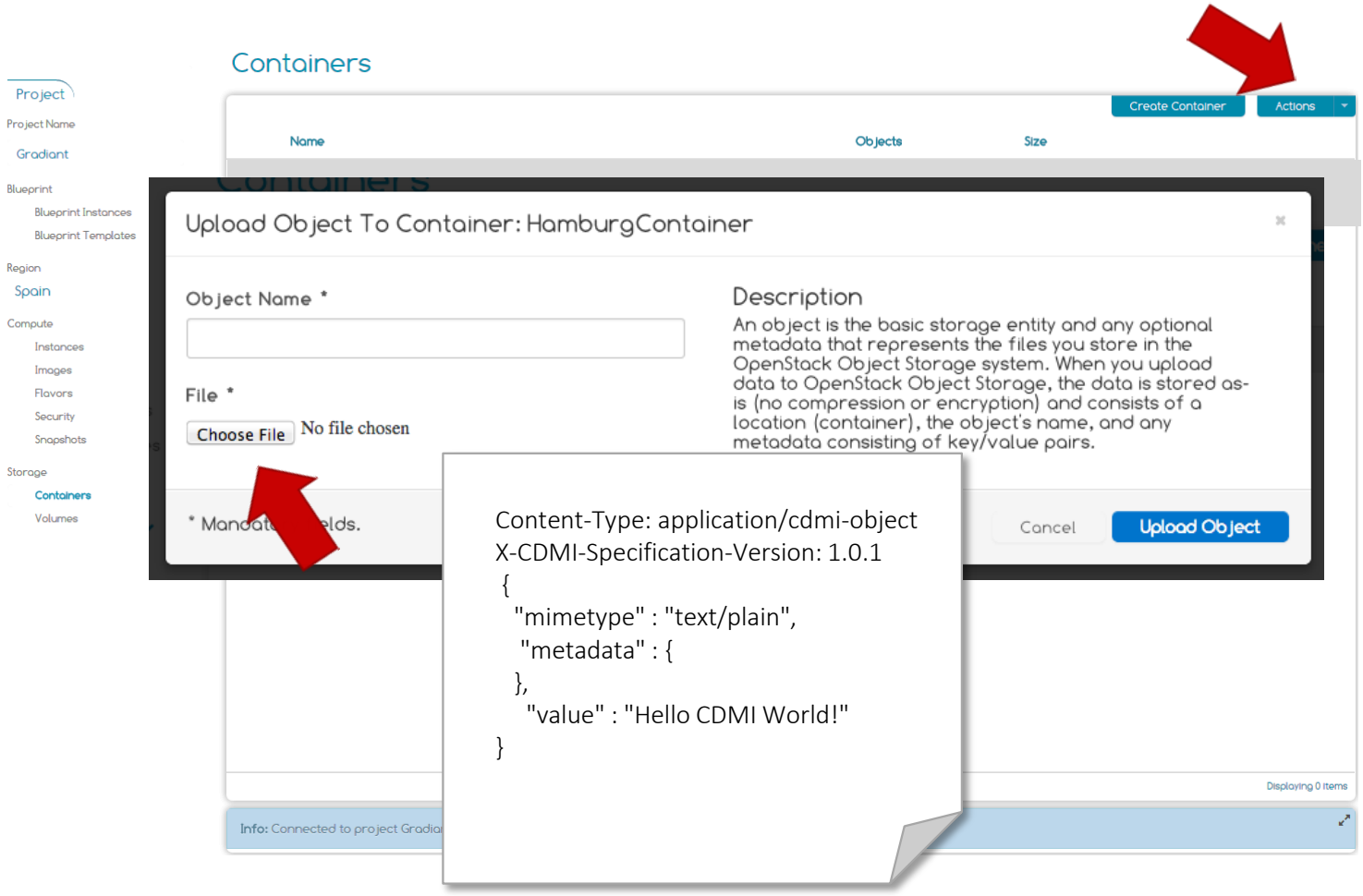
These containers can be created and removed on demand.

Object Storage GE – Object Upload

Objects can be files, databases or other datasets which need to be archived. Objects are stored in containers.

Containers and objects can have ***metadata*** associated with them, providing details of what the data represents.

To upload an object, a user just have to select the container to store it, click on the "upload object" action and upload the object definition file.



Object Storage GE - CL

Authenticate

```
$ curl -d '{"auth": {"project": "admin", "passwordCredentials": {"username":"admin", "password":„..."},
"tenantId":"d418851c6d294381bbe6e082849686d6"}}' - H"Content-type: application/json“
http://130.206.80.100:5000/v2.0/tokens
```

Capability discovery

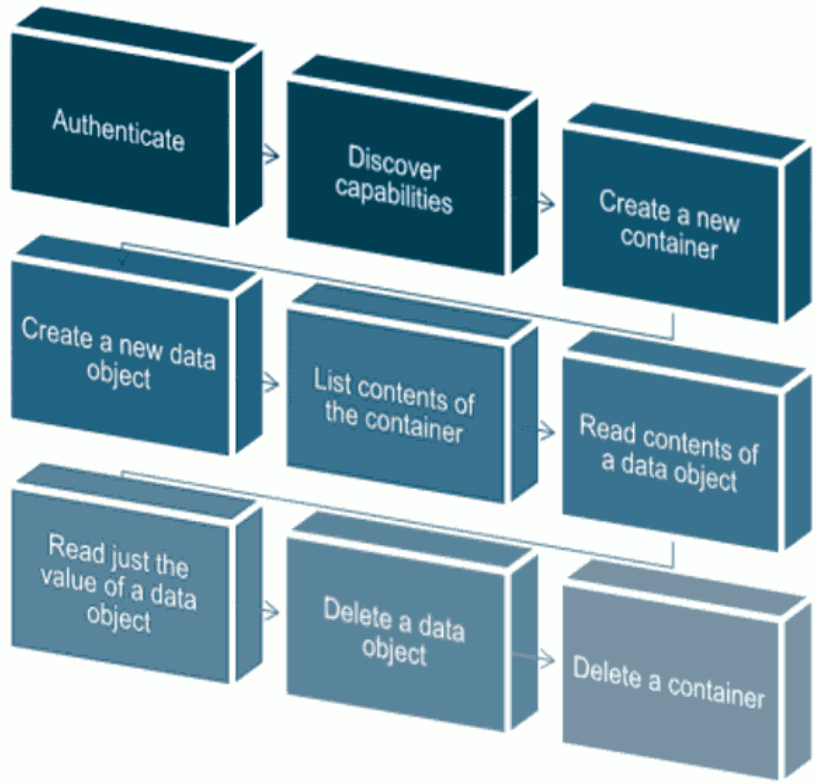
```
$ curl -v -X GET -H 'X-Auth-Token: b518d061ce2844ac80d1e116c9d857ed' - H 'Accept: application/cdmi-
capability' -H 'X-CDMI-SpecificationVersion: 1.0.1‘
http://130.206.80.102:8080/cdmi_capabilites/AUTH_d418851c6d294381bbe6e
082849686d6/
```

Create container

```
$ curl -v -X PUT -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' -
H 'Content-Type: application/directory'-H 'Content-Length: 0'
http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/
foo/
```

Create object

```
$ curl -v -X PUT -d '{"mimetype":"text/plain", "metadata":{}, "value":"bar"}' -H 'X-Auth-Token:
8197f75188304431bd3181ca79b0b884' –H 'Accept: application/cdmi-object' -H 'Content-Type:
application/cdmiobject‘ -H 'X-CDMI-Specification-Version: 1.0.1‘
http://130.206.80.102:8080/[...]/foo/text_doc
```



Object Storage GE – CL II

List container contents

```
$ curl -v -X GET -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884‘ -H 'Content-Type: application/directory’ http://130.206.80.102:8080/cdmi/AUTH_d418851c6d94381bbe6e082849686d6/foo/
```

Retrieve object

```
$ curl -v -X GET -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' - H 'Accept: application/cdmi-object' -H 'X-CDMI-Specification-Version: 1.0.1' http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo/text_doc
```

Retrieve object (2)

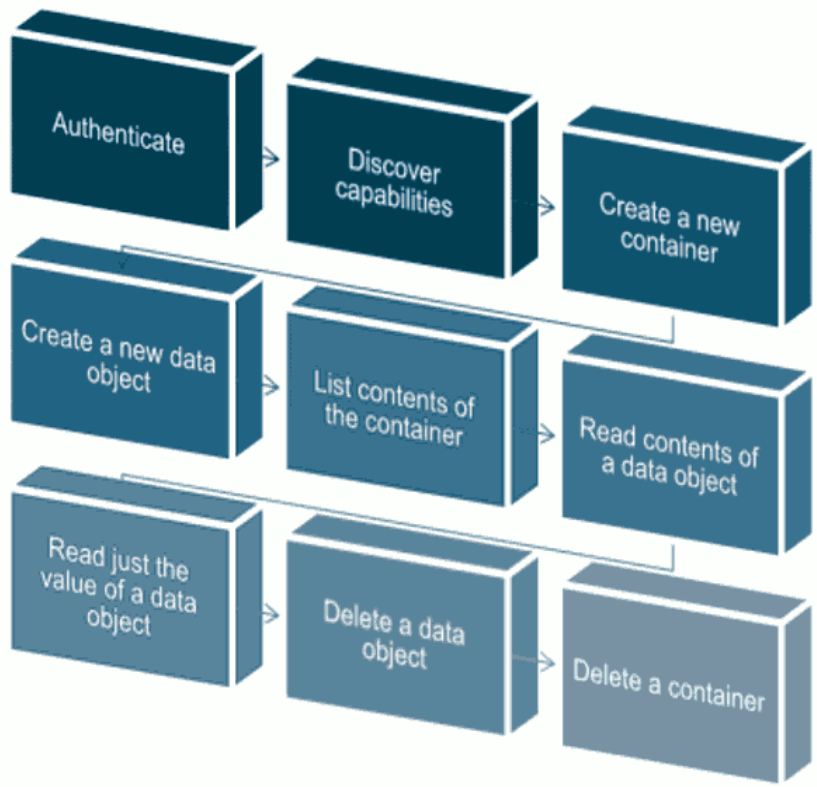
```
$ curl -v -X GET -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' -H ‘Accept: text/plain‘ http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo/text_doc
```

Delete object

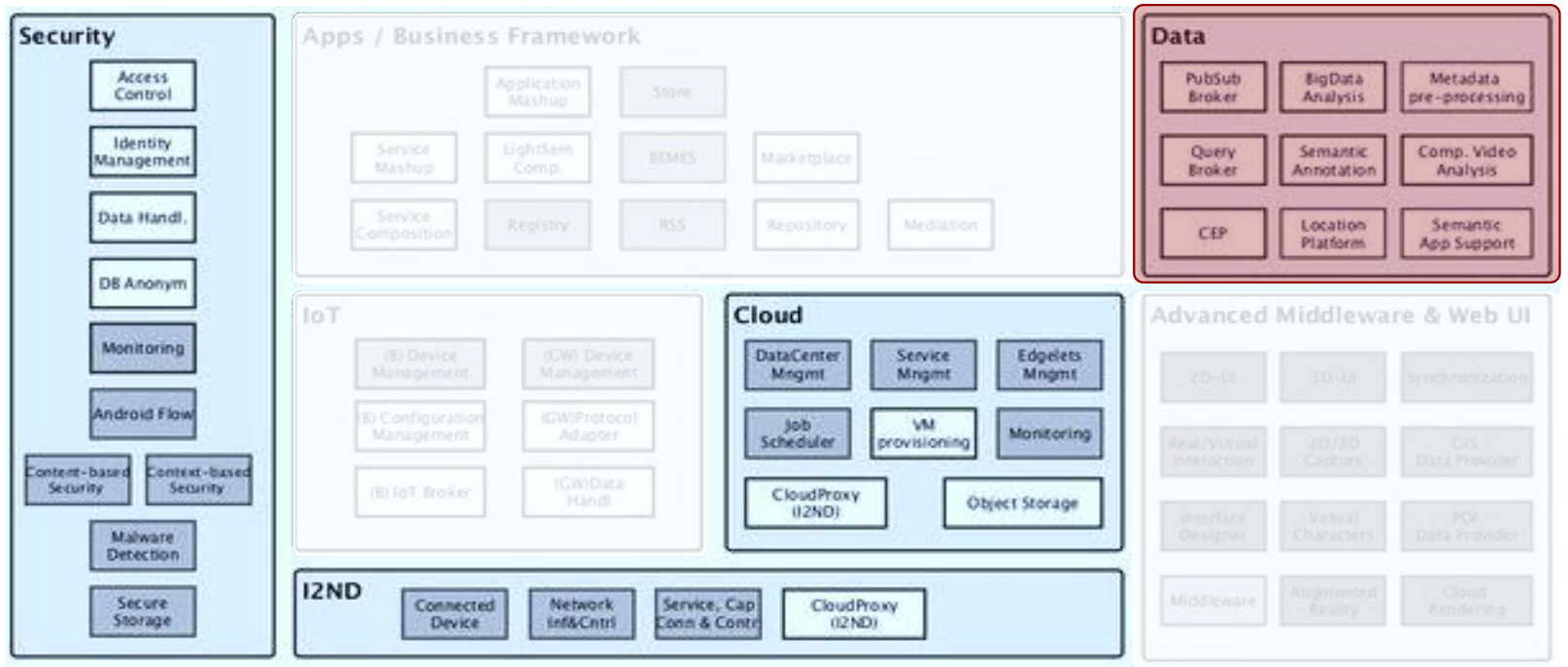
```
$ curl -v -X DELETE -H 'X-Auth-Token: 8197f75188304431bd3181ca79b0b884' http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo/text_doc
```

Delete container

```
$ curl -v -X DELETE -H 'X-Auth-Token:8197f75188304431bd3181ca79b0b884' http://130.206.80.102:8080/cdmi/AUTH_d418851c6d294381bbe6e082849686d6/foo
```



DATA/CONTEXT MANAGEMENT



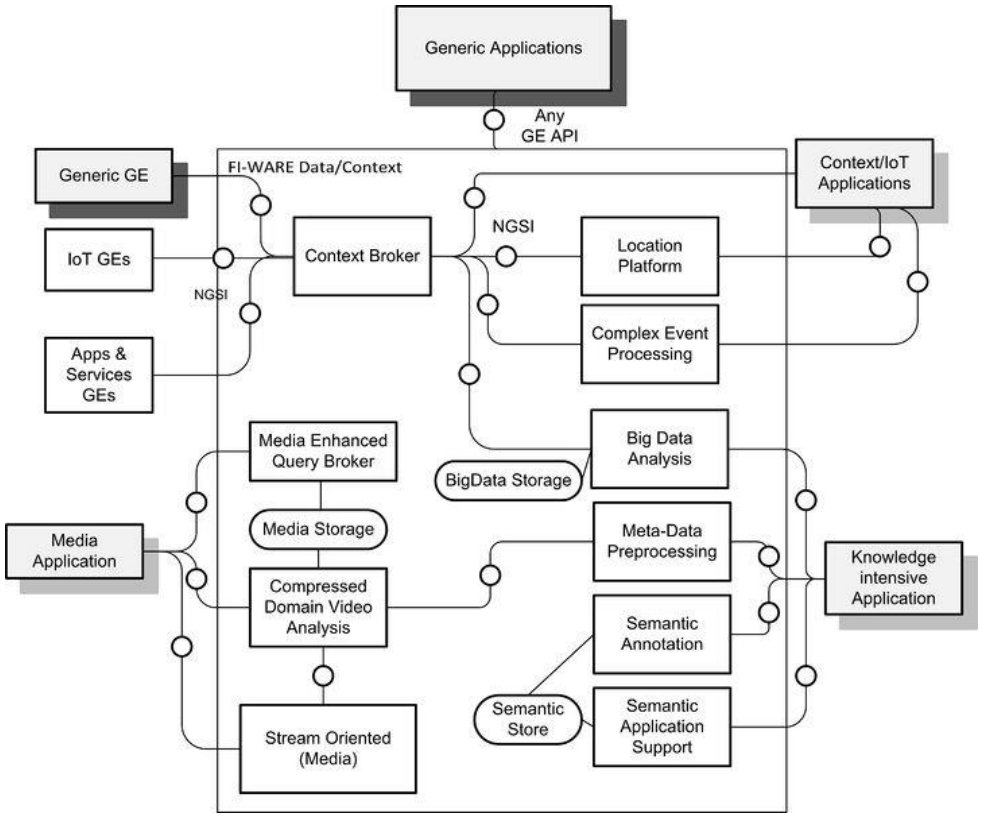
Architecture Overview

The **Data/Context Management** FI-WARE module contains the GEs that will ease development and provision of applications that require gathering, publication, processing and exploitation of information and data streams in real-time and at massive scale.

FI-WARE Data/Context Management GEs allow:

- Gather information from context and other sources (**Context Broker**)
- Mediate metadata among GEs and applications (**Metadata pre-processor**)
- Query stored information through an homogeneous layer (**Query Broker**)
- Annotate existing information (**Semantic Annotation**)
- Store and manage semantic information (**Semantic Application Support**)
- Generate new knowledge from big data stores using a Map & Reduce paradigm (**Big Data analysis**)
- React to different types of scenarios (**Complex Event Processing**).

It also provides GEs for media management, as easy creation of advanced video applications (**Stream Oriented GE**) and specifically, video analysis in the compressed domain.



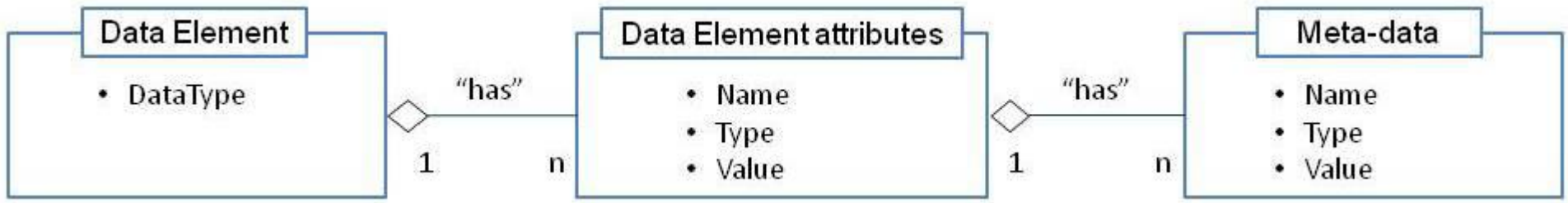
Data Elements

Data in FI-WARE refers to information that is produced, generated, collected or observed that may be relevant for processing, carrying out further analysis and knowledge extraction.

A **data element** refers to data whose value is defined as consisting of a sequence of one or more **<name, type, value> triplets** referred as data element attributes, where the type and value of each attribute is either mapped to a basic data type and a basic data value or mapped to the data type and value of another data element.

Each data element has an associated **data type** that determines s concrete sequence of attributes.

There may be **meta-data** linked to attributes in a data element.



Context Elements and Events

A **Context Element** extends the concept of data element by associating an **EntityId** and **EntityType** to it, uniquely identifying the entity in the FI-WARE system.

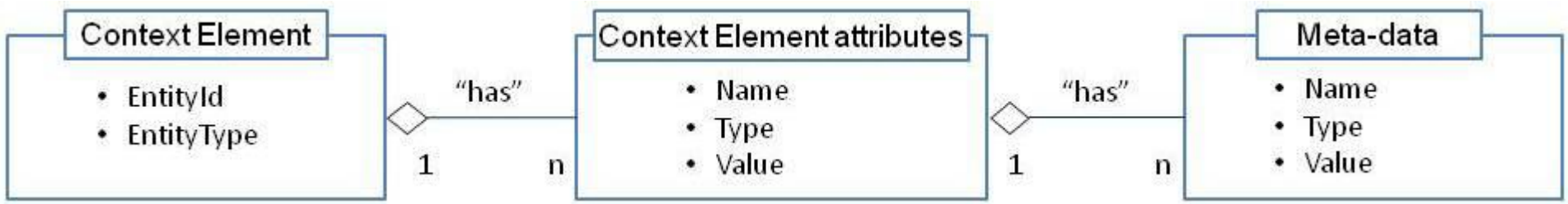
In addition, there may be some attributes as well as **meta-data** associated to attributes that we may define as mandatory for context elements as compared to data elements

An **Event** is an occurrence within a particular system that typically lead to creation of some data or context element, which enables applications or event-aware GEs to handle the information described or associated to it.

JSON Context Element EXAMPLE

```
"contextElements": [
                    { "type": "Room",
                    "isPattern": "false",
                     "id": "Room2",
                    "attributes": [
                                        { "name":
"temperature",
                                          "type": "float",
                                          "value": "21" },

                                        { "name": "pressure",
                                         "type": "integer",
                                        "value": "711" }
                    ] }
]
```



ORION - Publish/Subscribe Context Broker

Orion Context Broker

The **Orion Context Broker** is an implementation of the Publish/Subscribe Context Broker GE, providing the NGSI9 and NGSI10 interfaces.

Using these interfaces, clients can do several operations:

- **Register context producer applications**, e.g. a temperature sensor within a room
- **Update context information**, e.g. send updates of temperature
- **Being notified when changes on context information take place** (e.g. the temperature has changed) or with a given frequency (e.g. get the temperature each minute)
- **Query context information**. The Orion Context Broker stores context information updated from applications, so queries are resolved based on that information.

ORION – Operation flow

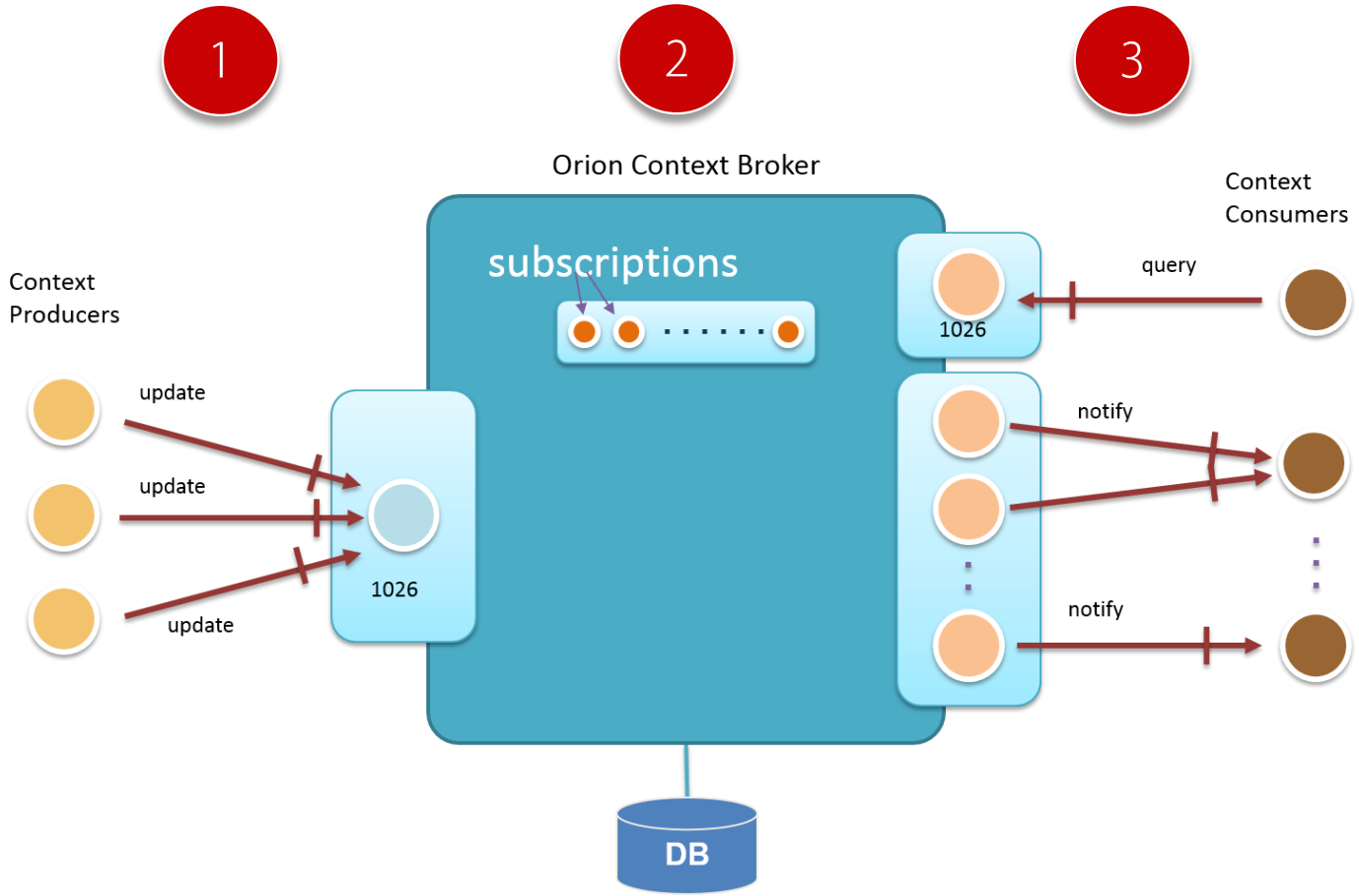
1. **Context producers** publish data/content elements by invoking the ***updateContext*** operation.

2. **Data** is kept by the Context Broker and ready to be required while not exceeding a given expiration time.

3. **Context consumers** can retrieve data/context elements by:

   a) Invoking the ***queryContext*** operation

   b) Being subscribed for a period of time to data/context elements which comply with certain conditions, using the ***subscribeContext*** operation. Subscribed consumers spontaneously receive data/context elements through ***notifyContext*** operation.



ORION – Operations

Entity Creation Operation

```
(curl localhost:1026/v1/updateContext -s -S --header 'Content-Type:
application/json' --header 'Accept: application/json' -d @- | python -
mjson.tool) <<EOF
{
   "contextElements": [
      {
         "type": "Room",
         "isPattern": "false",
         "id": "Room1",
         "attributes": [
         {
            "name": "temperature",
            "type": "float",
            "value": "23"
         },
         {
            "name": "pressure",
            "type": "integer",
            "value": "720"
         }
         ]
      }
   ],
   "updateAction": "APPEND"
}
EOF
```

entityId="Room1" entityType="Room"

Attribute 1

Attribute 2

Operation Mode: APPEND

Query Context operation

```
(curl localhost:1026/v1/queryContext -s -S --header 'Content-Type: application/json' --
header 'Accept: application/json' -d @- | python -mjson.tool) <<EOF
{
   "entities": [
   {
      "type": "Room",
      "isPattern": "false",
      "id": "Room1"
   }
   ],
   "attributes" : [
      "temperature"
   ]
}
EOF
```

Requesting from entity "Room1" of type "Room"

Requesting attribute name="temperature"

Note:
If you use an **empty attributeList** element in the request the response will include **all the attributes** of the entity.

ORION – Operations II

Update context operation

```
(curl localhost:1026/v1/updateContext -s -S --header 'Content-Type:
application/json' --header 'Accept: application/json' -d @- | python -
mjson.tool) <<EOF
{
   "contextElements": [
      {
         "type": "Room",
         "isPattern": "false",
         "id": "Room1",
         "attributes": [
         {
            "name": "temperature",
            "type": "float",
            "value": "26.5"
         },
         {
            "name": "pressure",
            "type": "integer",
            "value": "763"
         }
         ]
      }
   ],
   "updateAction": "UPDATE"
}
EOF
```

Updating entity "Room1" of type "Room"

Update "temperature"="26,5"

Update "pressure"="763"

Operation Mode: UPDATE
(however, current Orion Context Broker version also interprets APPEND as UPDATE if the entity already exists)

Context subscription operation

Orion Context Broker can subscribe to context information so when "something" happens the application will get an asynchronous notification. Some new field are Included in the payload:

- **Reference:** The callback URL to send notifications.
- **Duration:** Specified using the ISO 8601 standard format. Once that duration is expired, the subscription is simply ignored
- **NotifyCondition:** element that sets the "trigger" for the subscription. It is defined by a ***type*** and a ***condValueList*** element. There are two types of subscriptions:
  - **ONTIMEINTERVAL:** that includes exactly one condValue child element whose value is a time interval. A notification is sent with a ***frequency*** equal to that interval
  - **ONCHANGE:** where the condValueList contains an actual list of condValue elements and ***if at least one*** of the attributes in the list changes then a notification is sent. A throttling element is used to specify a minimum inter-notification arrival time.

ORION – Operations III

ONCHANGE Context subscription operation

```
(curl localhost:1026/v1/subscribeContext -s -S --header 'Content-Type:
application/json' --header 'Accept: application/json' -d @- | python -
mjson.tool) <<EOF
{
   "entities": [
      {
         "type": "Room",
         "isPattern": "false",
         "id": "Room1"
      }
   ],
   "attributes": [
      "temperature"
   ],
   "reference": "http://localhost:1028/accumulate",
   "duration": "P1M",
   "notifyConditions": [
      {
         "type": "ONCHANGE",
         "condValues": [
            "pressure"
         ]
      }
   ],
   "throttling": "PT5S"
}
EOF
```

Subscribing to entity "Room1" of type "Room"

Subscribing to attribute name="temperature"

Subscription duration

Subscribing mode ONCHANGE

Trigger Attribute "pressure"

minimum inter-notification

ONTIMEINTERVAL Context subscription operation

```
(curl localhost:1026/v1/subscribeContext -s -S --header 'Content-Type:
application/json' --header 'Accept: application/json' -d @- | python -mjson.tool) <<EOF
{
   "entities": [
      {
         "type": "Room",
         "isPattern": "false",
         "id": "Room1"
      }
   ],
   "attributes": [
      "temperature"
   ],
   "reference": "http://localhost:1028/accumulate",
   "duration": "P1M",
   "notifyConditions": [
      {
         "type": "ONTIMEINTERVAL",
         "condValues": [
            "PT10S"
         ]
      }
   ]
}
EOF
```

Subscribing to entity "Room1" of type "Room"

Subscribing to attribute name="temperature"

Subscription duration

Subscribing mode ONTIMEINTERVAL

Subscription notification interval

ORION – Advanced Features

Compound Attribute Values

```
{
   "contextElements": [
      {
         "type": "Car",
         "isPattern": "false",
         "id": "Car1",
         "attributes": [
         {
            "name": "tirePressure",
            "type": "kPa",
            "value":  {
              "frontRight": "120",
              "frontLeft": "110",
              "backRight": "115",
              "backLeft": "130"
            }
         }
         ]
      }
   ],
   "updateAction": "APPEND"
}
```

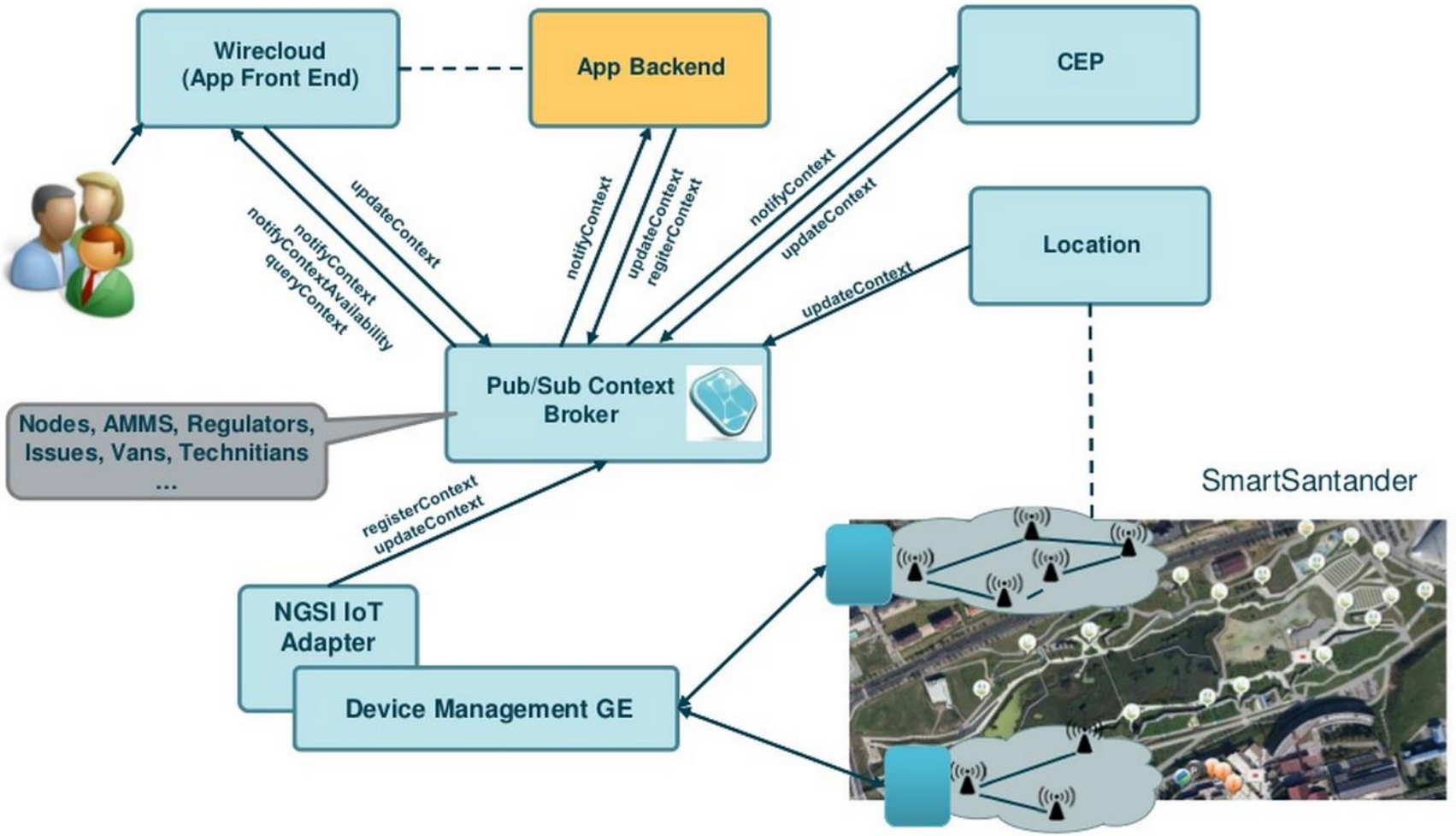
Metadata

```
...
"attributes": [
   {
    "name": "temperature",
    "type": "float",
    "value": "26.5",
    "metadatas": [
    {
     "name": "accuracy",
     "type": "float",
     "value": "0.9"
    }
    ]
   }
   ]
```

Geolocation

```
"contextElements": [
 {
  "type": "City",
  "isPattern": "false",
  "id": "Madrid",
  "attributes": [
  {
   "name": "position",
   "type": "coords",
   "value": "40.418889, -3.691944",
   "metadatas": [
    {
     "name": "location",
     "type": "string",
     "value": "WSG84"
    }
   ]
  }
  ]
 }
 ],
 "updateAction": "APPEND"
}
```

ORION – Use case



COSMOS - BigData Analysis

COSMOS
Big Data Analysis

COSMOS is an implementation of the Big Data GE, allowing the dynamic deployment of private computing clusters for **persisted storage**.

Cosmos allows users to:

- **I/O operations** over a storage cluster based on HDFS (*Hadoop Distributed File System*).
- **Private computing clusters** creation, usage and deletion based on MapReduce and SQL-like querying systems such as Hive or Pig.
- Manage the platform, in many aspects such as services, users, clusters, etc, from the Cosmos API or the Cosmos CLI

Cosmos is **Hadoop** ecosystem-based that deploys:

- HDFS as its distributed file system
- Hadoop core as its MapReduce engine
- HiveQL and Pig for querying the data
- Oozie as remote MapReduce jobs and Hive launcher

COSMOS – Platform management

Data Management

Hadoop Distributed File System

Hadoop HDFS is a highly fault-tolerant distributed file system that provides high throughput access to application data and is suitable for applications that have large datasets.

Hadoop HDFS allows users to:

- List, Create, Delete or Rename a file or directory
- Create a new file with initial content
- Append to a file
- Open and read a file
- Concat files
- Set owners and permissions

Data Querying

Hive

Hive is a querying tool:

- Queries are expresed in **HiveQL**, a SQL-like language
- Uses pre-defined **MapReduce** jobs for
  - Column selection
  - Fields grouping
  - Table joining
- All the data is loaded into **Hive tables**

A remote Hive client usually performs:

- A connection to the Hive server
- The query execution

COSMOS – Data management

Create a new file with initial content

```
PUT http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=CREATE
[&overwrite=<true|false>][&blocksize=<LONG>][&replicatio
n=<SHORT>] [&permission=<OCTAL>][&buffersize=<INT>]
HTTP/1.1 307 TEMPORARY_REDIRECT
Location:
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=CREATE...
Content-Length: 0

PUT -T <LOCAL_FILE>
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=CREATE...
```

Append to a file

```
POST
http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=APPEND[&buffer
size=<INT>]
HTTP/1.1 307 TEMPORARY_REDIRECT
Location:
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=APPEND...
Content-Length: 0

POST -T <LOCAL_FILE>
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=APPEND...
```

Open and read a file

```
GET http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=OPEN
[&offset=<LONG>][&length=<LONG>][&buffersize=<INT>]
HTTP/1.1 307 TEMPORARY_REDIRECT
Location:
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=OPEN...
Content-Length: 0

GET
http://<DATANODE>:<PORT>/webhdfs/v1/<PATH>?op=OPEN...
```

COSMOS – Data querying

Getting a connection

```
private Connection getConnection(
      String ip, String port, String user, String
password) {
   try {
      // dynamically load the Hive JDBC driver

Class.forName("org.apache.hadoop.hive.jdbc.HiveDriver");
   } catch (ClassNotFoundException e) {
      System.out.println(e.getMessage());
      return null;
   } // try catch

   try {
      // return a connection based on the Hive JDBC
driver, default DB
      return DriverManager.getConnection("jdbc:hive://"
+ ip + ":" +
         port + "/default?user=" + user + "&password=" +
password);
   } catch (SQLException e) {
      System.out.println(e.getMessage());
      return null;
   } // try catch
} // getConnection
}
```

Doing the query

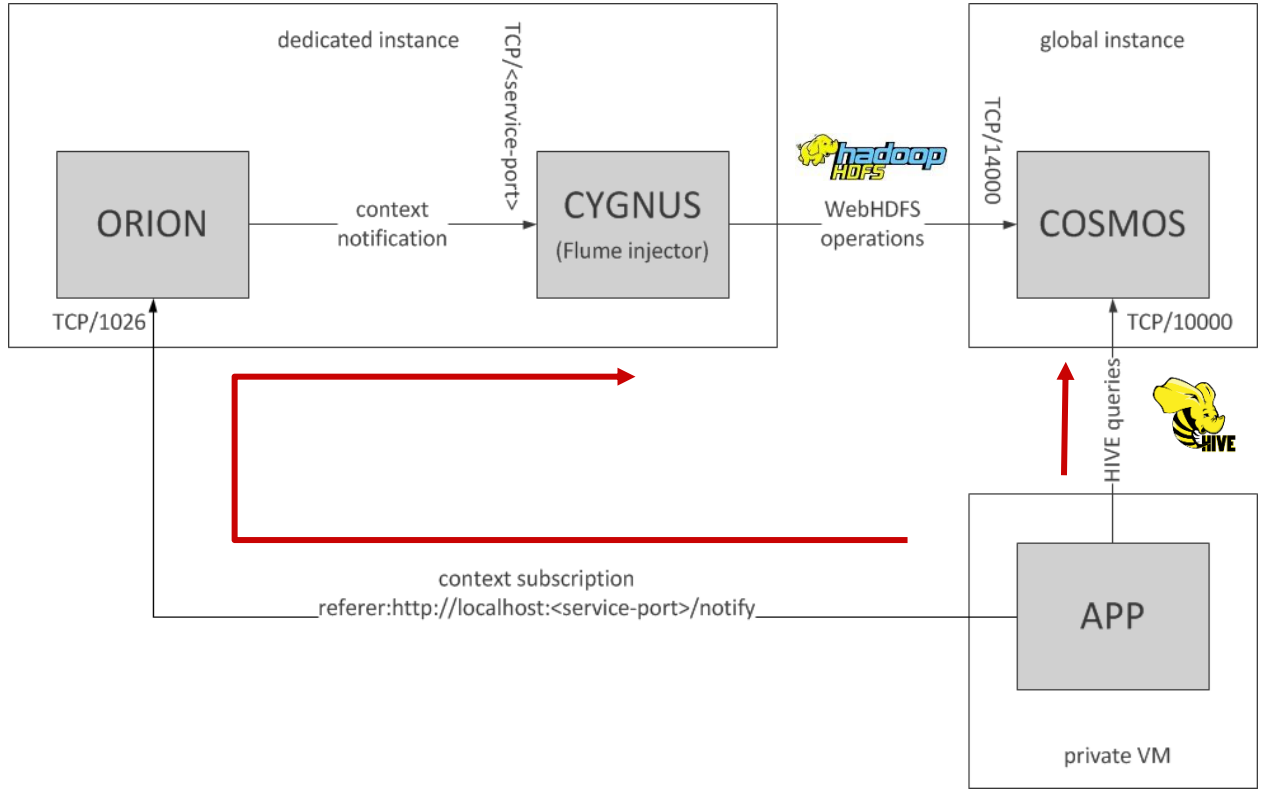
```
private void doQuery() {
   try {
      // from here on, everything is SQL!
      Statement stmt = con.createStatement();
      ResultSet res = stmt.executeQuery("select
column1,column2," +
         "otherColumns from mytable where
column1='whatever' and " +
         "columns2 like '%whatever%'");

      // iterate on the result
      while (res.next()) {
         String column1 = res.getString(1);
         Integer column2 = res.getInteger(2);
         // whatever you want to do with this row, here
      } // while
      // close everything
      res.close(); stmt.close(); con.close();
   } catch (SQLException ex) {
      System.exit(0);
   } // try catch
} // doQuery
```

COSMOS – Cygnus

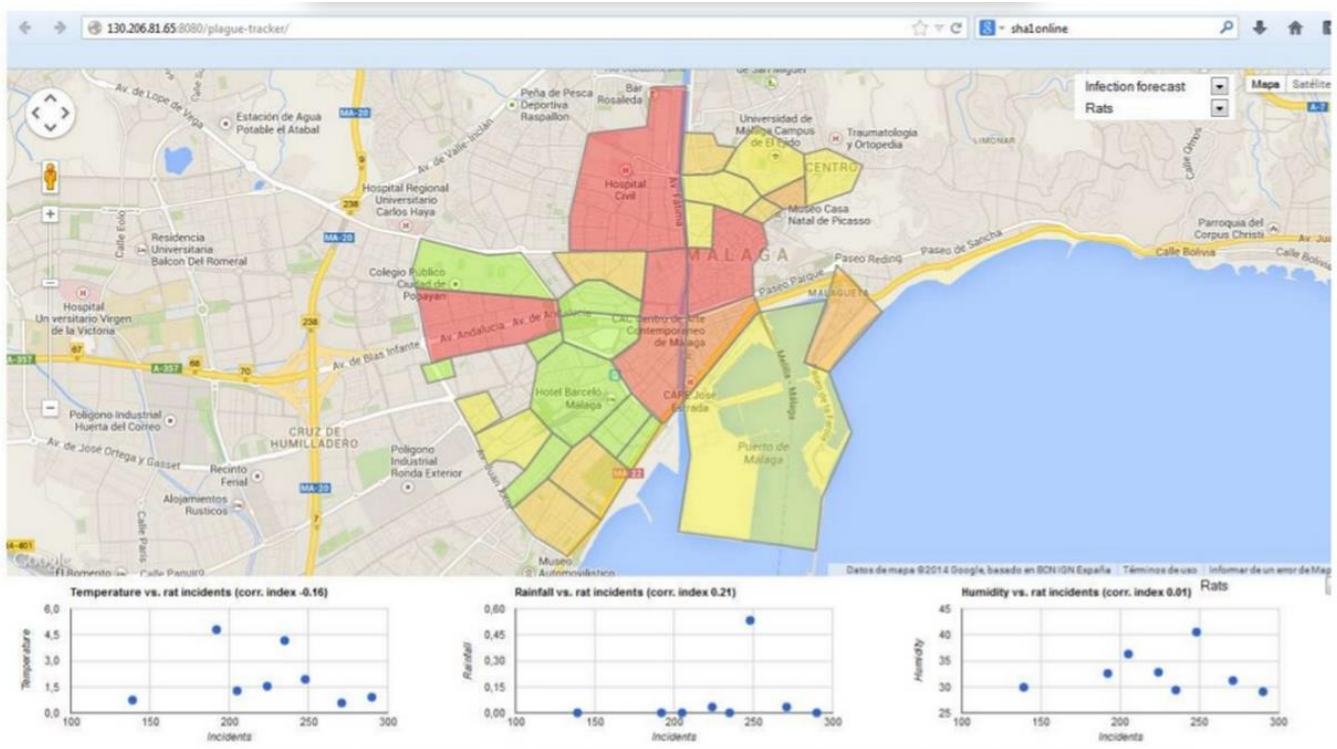
Cygnus offers integration between Context Broker and Cosmos Big Data. Context data is managed by Orion in last-value-storage, thus no historical data is stored. That historical storage could be possible if Orion writes all the updates in Cosmos.

Cygnus subscribes to Orion in order to receive notifications of context data that must be persisted in Cosmos HDFS, in addition to any other persistent storage such as MySQL or CKAN.



COSMOS – Use case

Plague Tracker



https://github.com/telefonicaid/fiware-connectors/tree/develop/resources/plague-tracker

Plague Tracker processes the historical data about the plagues (rats, mice, pigeons, cockroaches, bees…) affecting the city of Malaga

The map can show the neighborhoods affected by a selected type of plague and those that will probably be infected based on the historical number of incidences, the weather and the proximity to already infected neighborhoods.

In addition to the map, three charts show the correlation index between the selected type of plague and three ambient parameters such as the temperature, the rainfall and the humidity.

KURENTO – Stream oriented

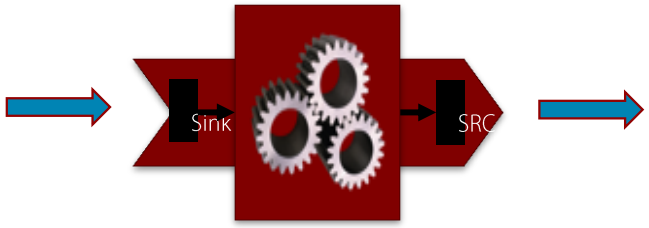
KURENTO
Stream oriented

KURENTO Stream Oriented GE makes possible to any WWW developer to create powerful applications with **advanced media capabilities**, such as interoperable audiovisual communications, computer vision, augmented reality, flexible media playing, recording, etc.

KURENTO's multimedia engine provides the following features:

- Networked streaming protocols, including HTTP working as client and server, RTP and WebRTC.
- Media transcodification between any of the codecs currently supported.
- Generic support for computational vision
- Augmented reality filters.
- Media storage supporting writing operations for WebM and.

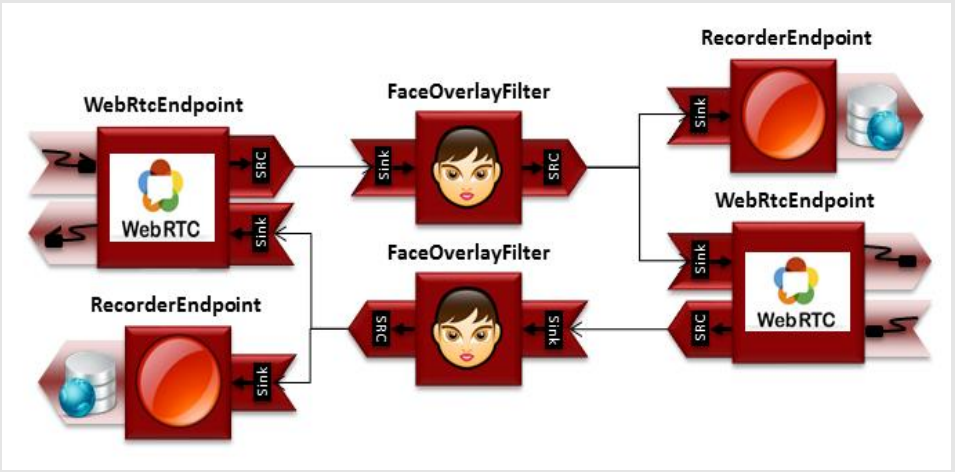
KURENTO – Example

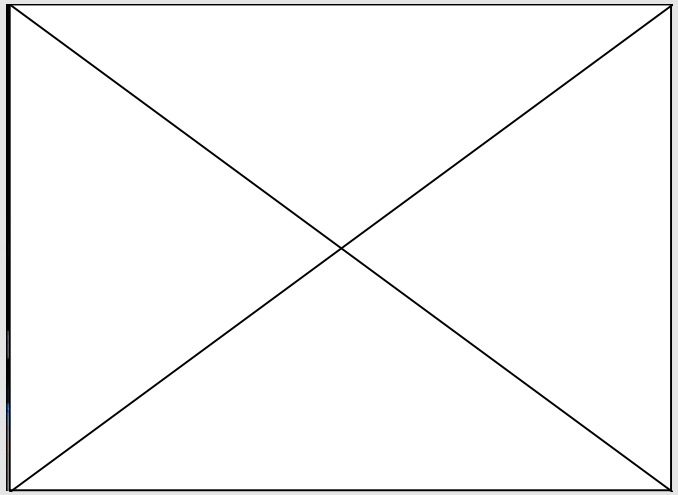


Toolbox of Media Elements that provides a specific media functionalities:

- Send/receive media.
- Process media
- Transform media

Media Elements can be easily **chained** to implement the desired media logic.





CKAN – OpenData datasets

CKAN is a tool for making open data websites that helps users to manage and publish collections of data.

CKAN

The data is published in units called “datasets”. A dataset is a parcel of data, for example, the spending figures for a government department or temperature readings from various weather stations.

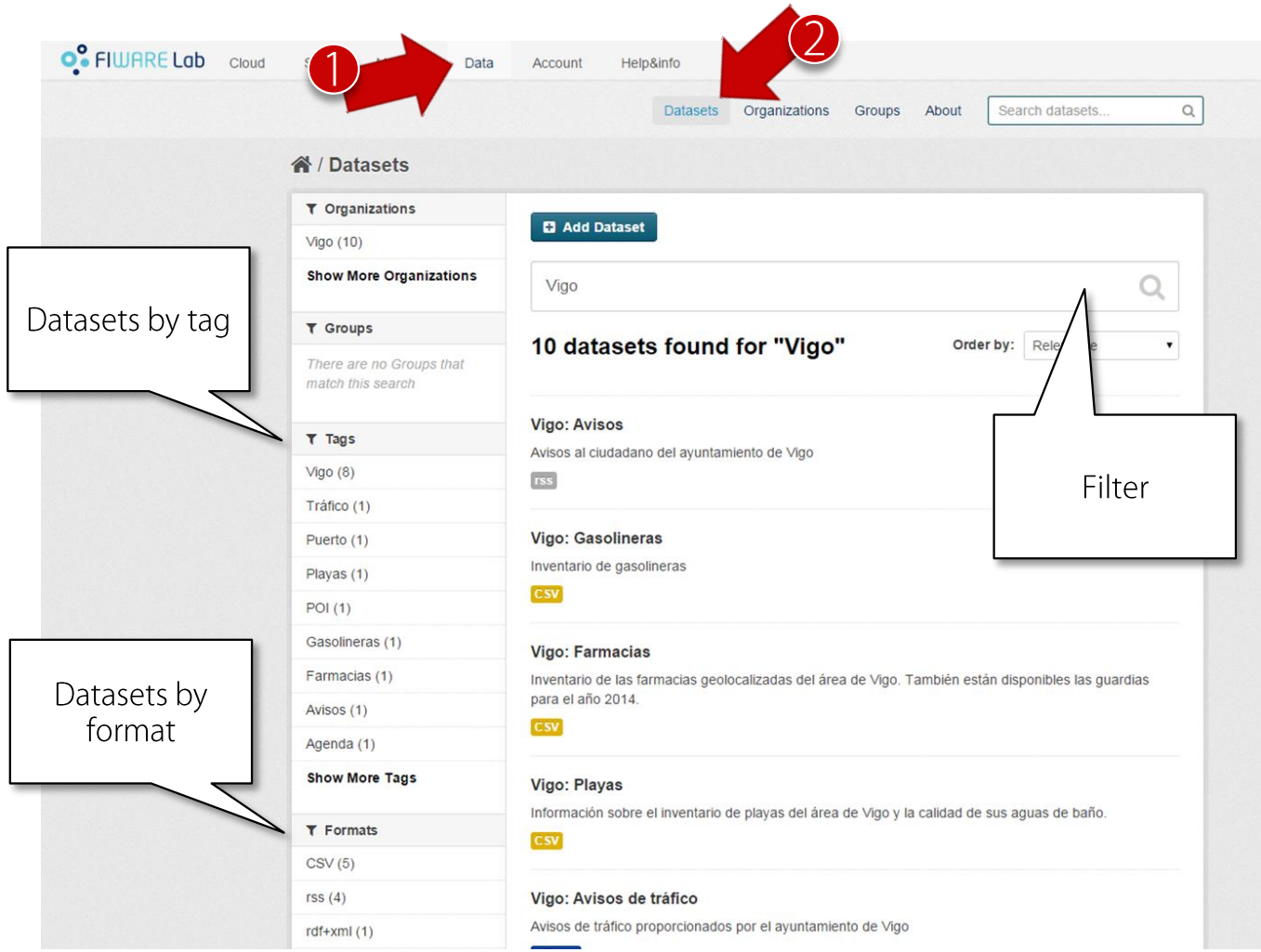
A dataset contains two things:

- **Information or metadata** about the data. For example, the title and publisher, date, what formats it is available in, etc.
- **A number of resources**, which hold the data itself. CKAN support many data formats: CSV or Excel spreadsheet, XML file, PDF document, image file, linked data in RDF format, etc.

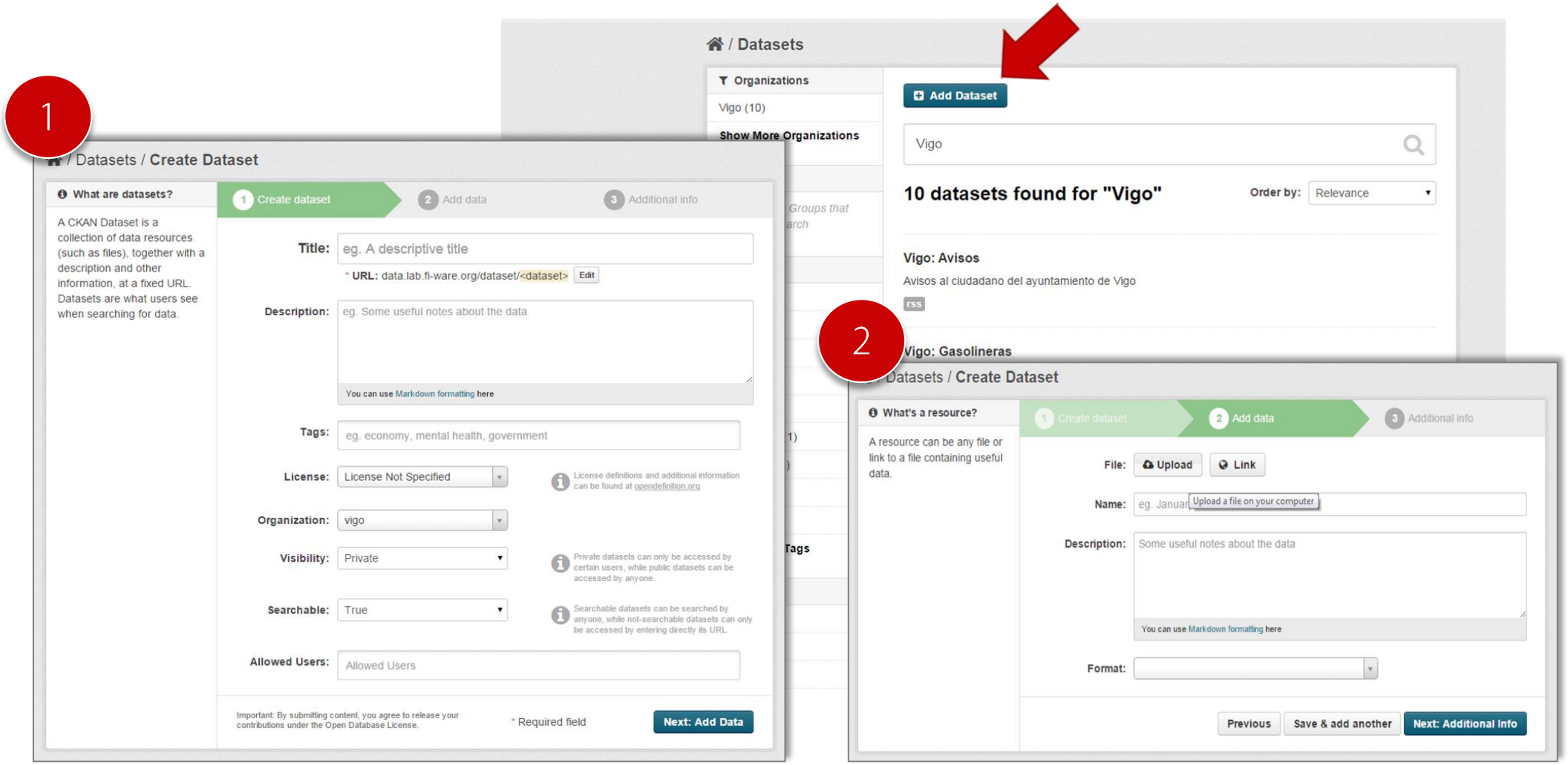
CKAN- Datasets

All the datasets of the FIWARE platform are available on the Data tab of the FIWARE Lab interface.

More concrete results can be obtained using the textual filter, where the user can specify some terms to search; or using the default classification by tags and data formats shown at the side bar.



CKAN – Dataset creation



CKAN – Data access and preview

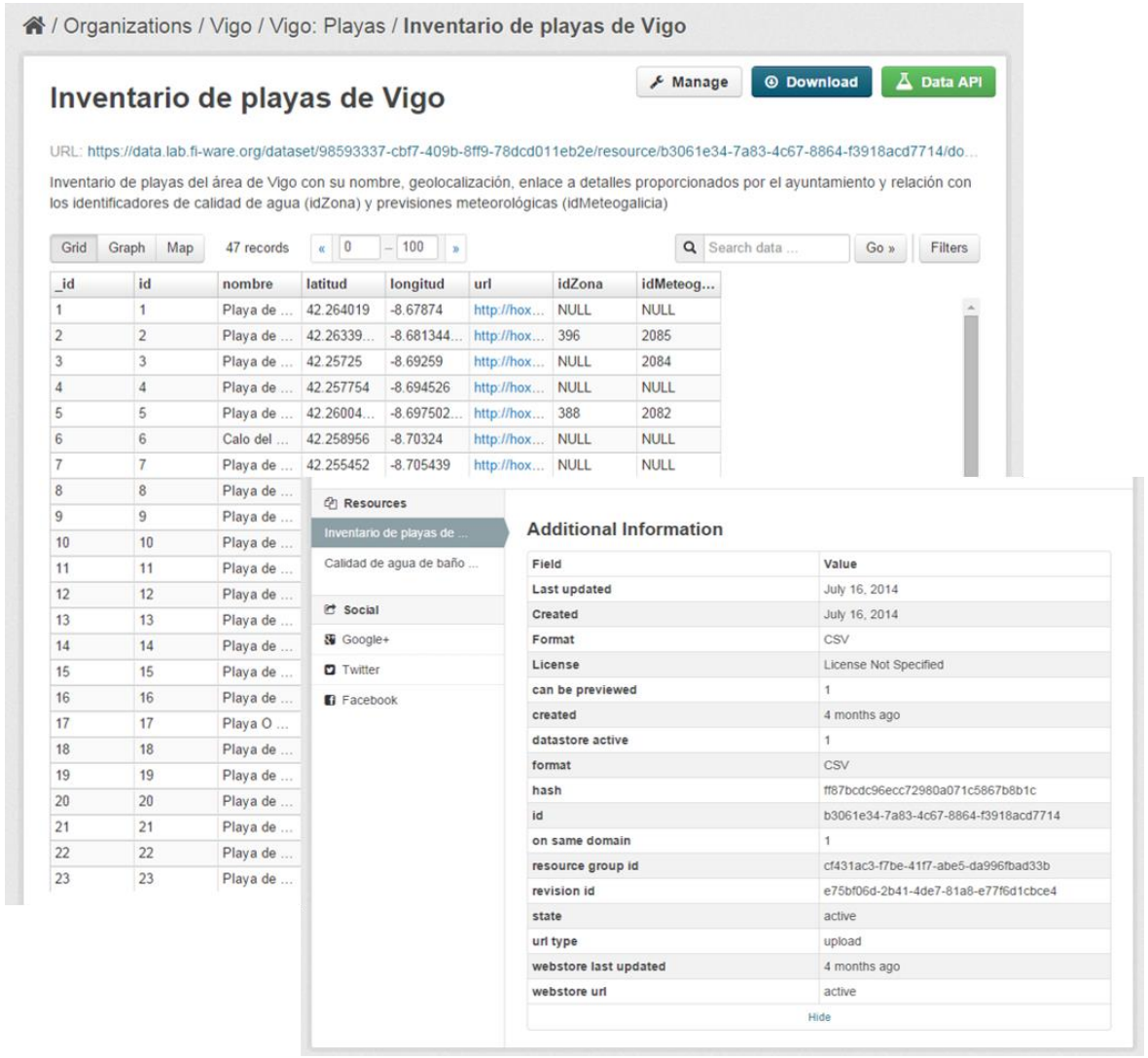
The FIWARE platform offers a data preview of each dataset.

Information can shown in three formats:

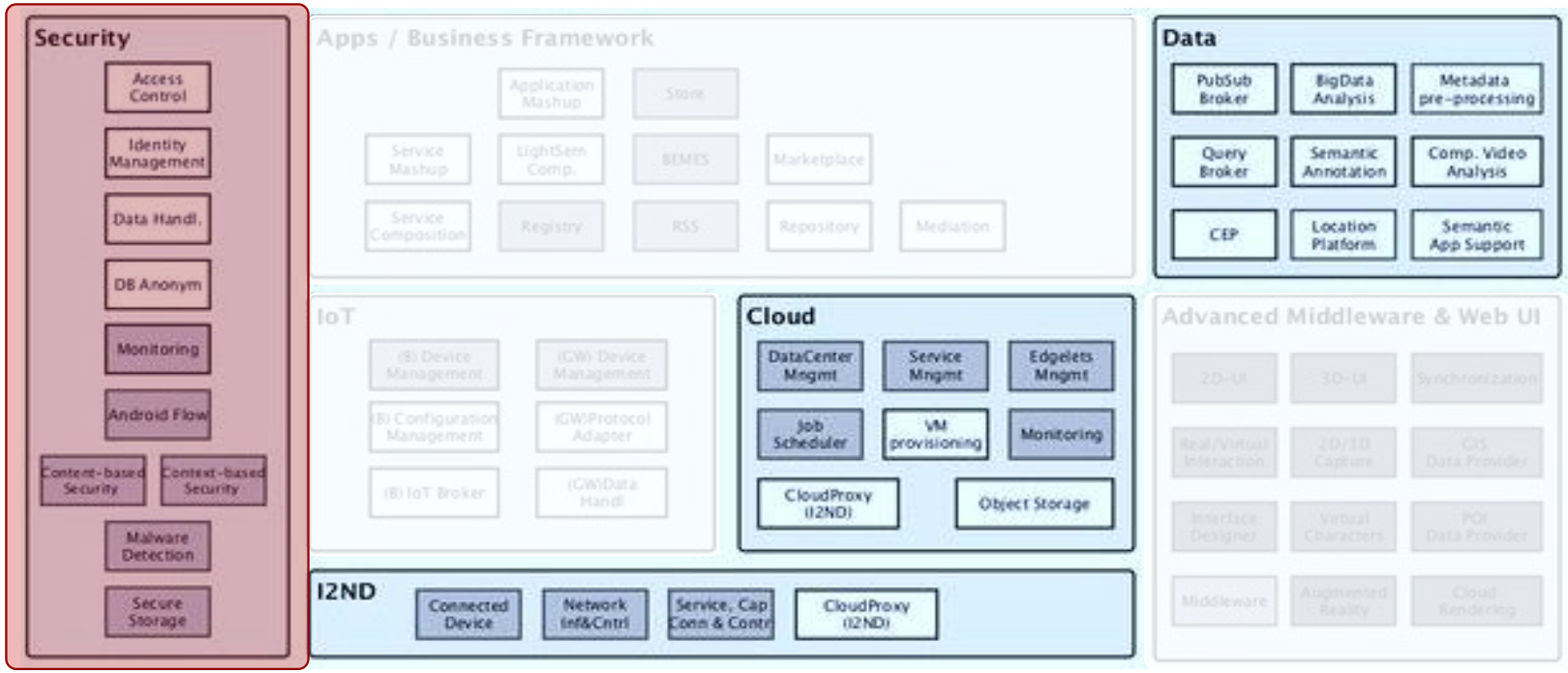
- As a **grid**, where data is display in rows and columns.
- As a **graph**, where the user can set what fields wants to be represented.
- As a **map**, using specific fields as latitude and longitude coordinates.

Users can **download** or **manage** the dataset and access the **data API** from this preview.

At the bottom of the page important additional information is presented.



SECURITY

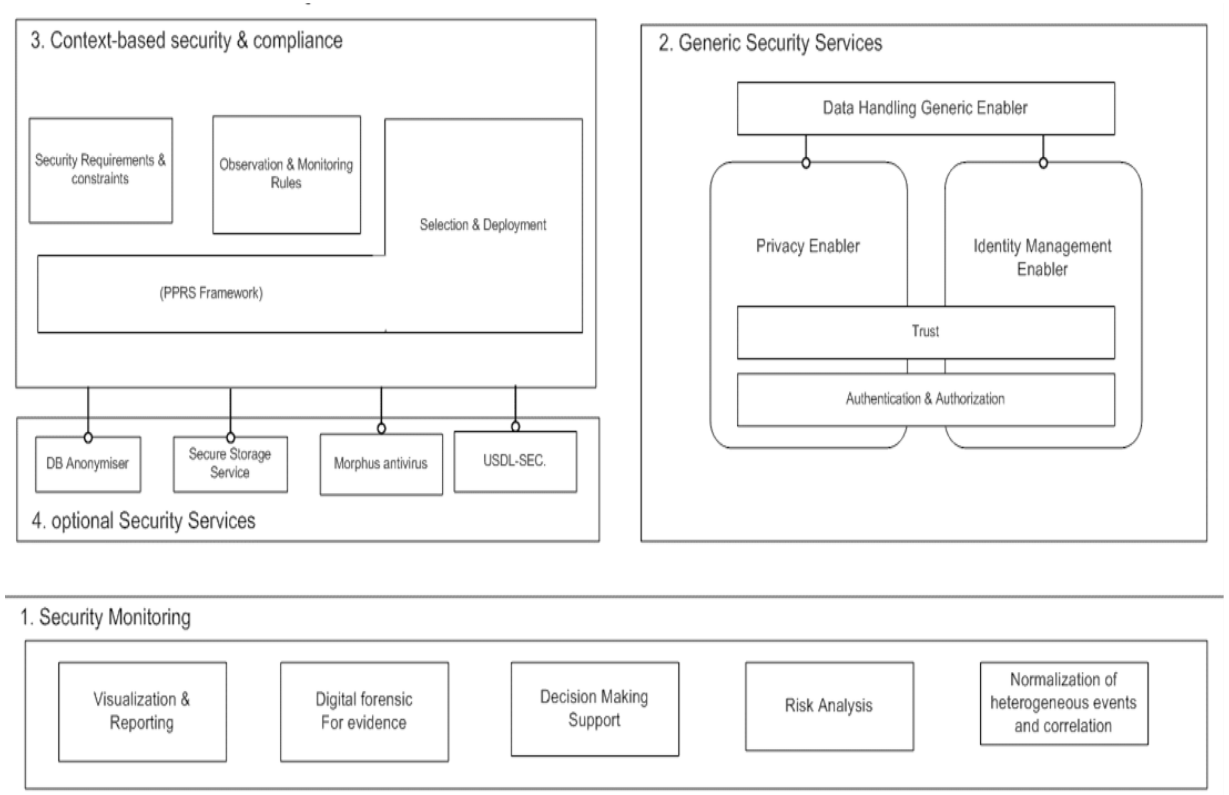


Architecture Overview

Security, Privacy and Trust in FI-WARE is mainly focusing on delivering tools and techniques to create secure services without sacrificing much of the desired functionality, usability, performance and cost efficiency is a great challenge

Security architecture is formed by four main blocks of GEs:

- **Security monitoring**: Collects vulnerabilities, evaluates threats, rises alarms and identifies attacks.
- **Generic Security Services:**
  - Identity Management
  - Access Control
  - Privacy
  - Data Handling
- **Context-Based Security and Compliance:** provides an additional security layer with context-aware capabilities.
- **Optional Generic Security Services:** DB Anonymizer, Secure Storage Service, Malware Detection Service, Android Flow Monitoring and Content-based Security.



Identity Management (IdM)- KeyRock

Identity Management (IdM) provides a way for controlling access to resources that are available on a given FI-WARE Instance.

KeyRock
Identity Management

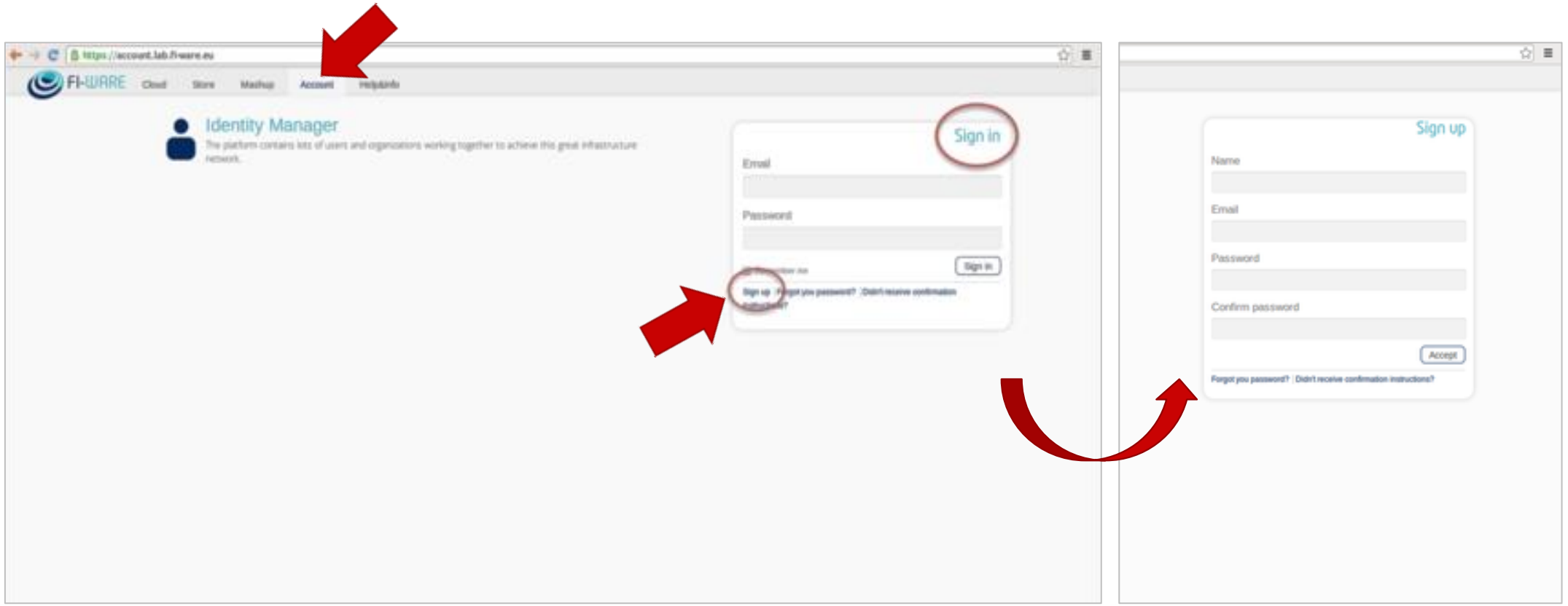
IdM include many aspects about users' access to networks, services or applications:

- Secure and private authentication
- Authorization & Trust management
- User Profile management
- Single Sign-On (SSO) to service domains and
- Identity Federation.

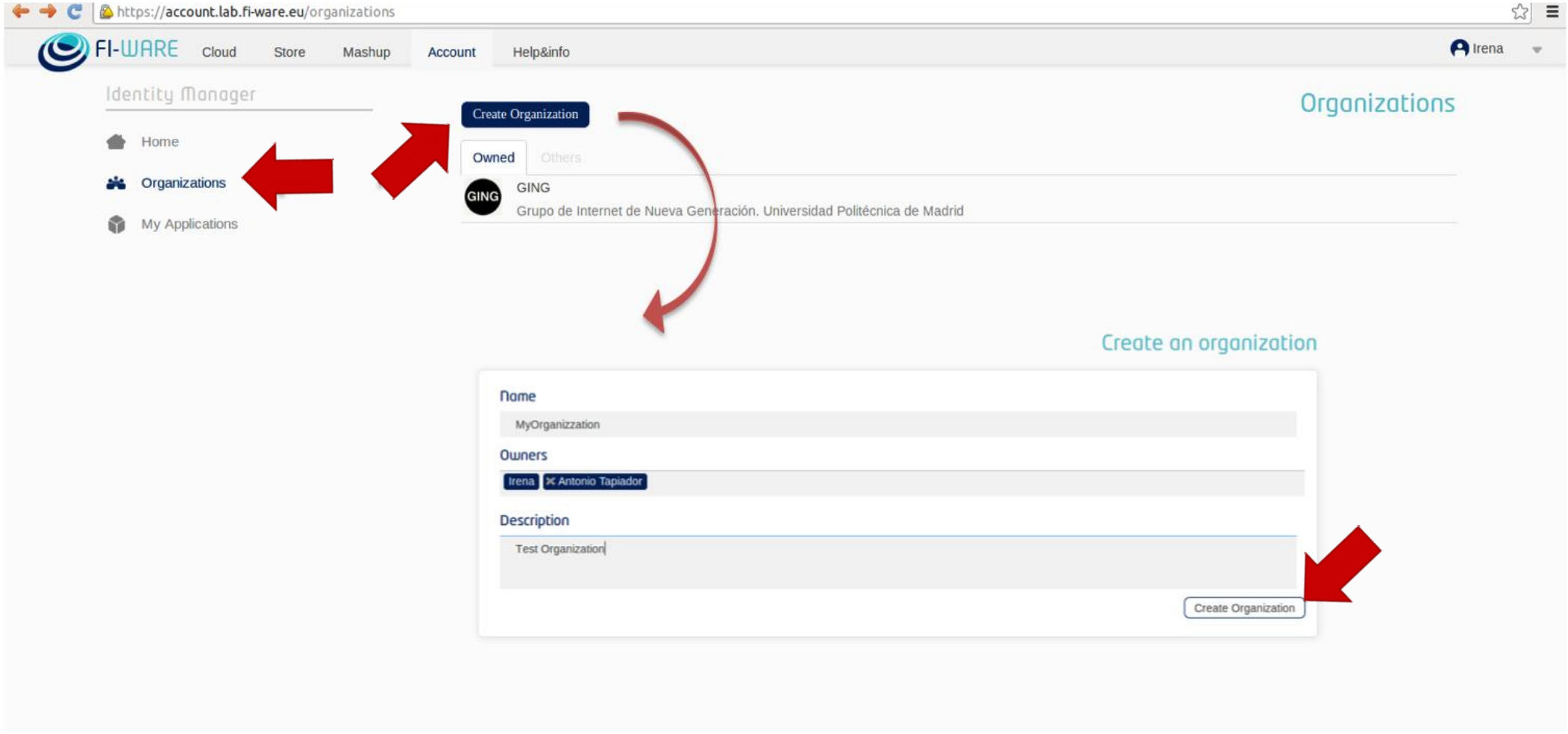
IdM reduces the effort of account creation and management, supporting the policies and procedures for user registration and user profile management:

- **Administrators** can include different authentication providers, registration of applications with access to user profiles and handling error notifications.
- **End users** get a convenient registering solution with re-use of attributes like address or email, allowing an easy management of profile information.

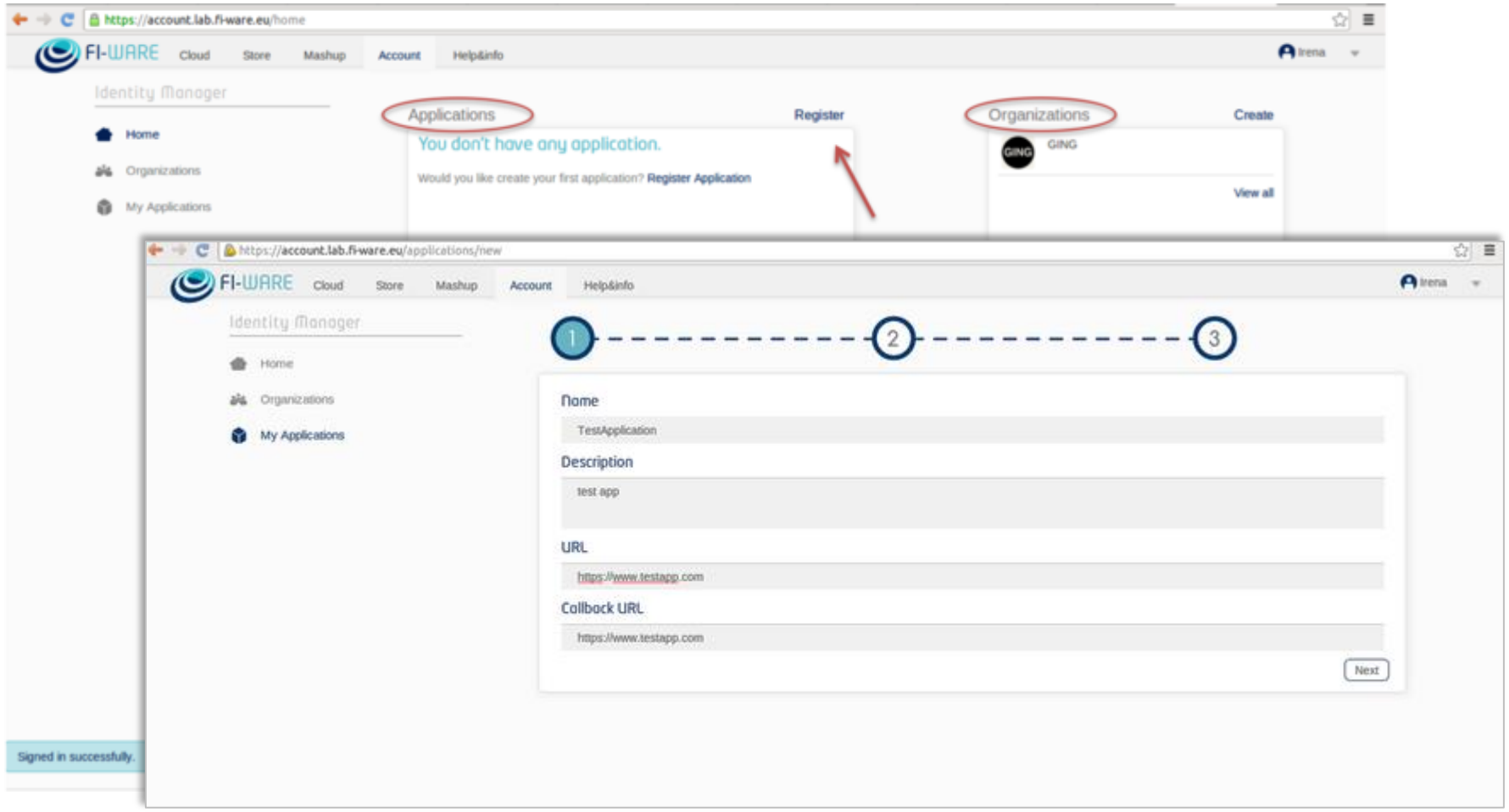
IdM – Logging in



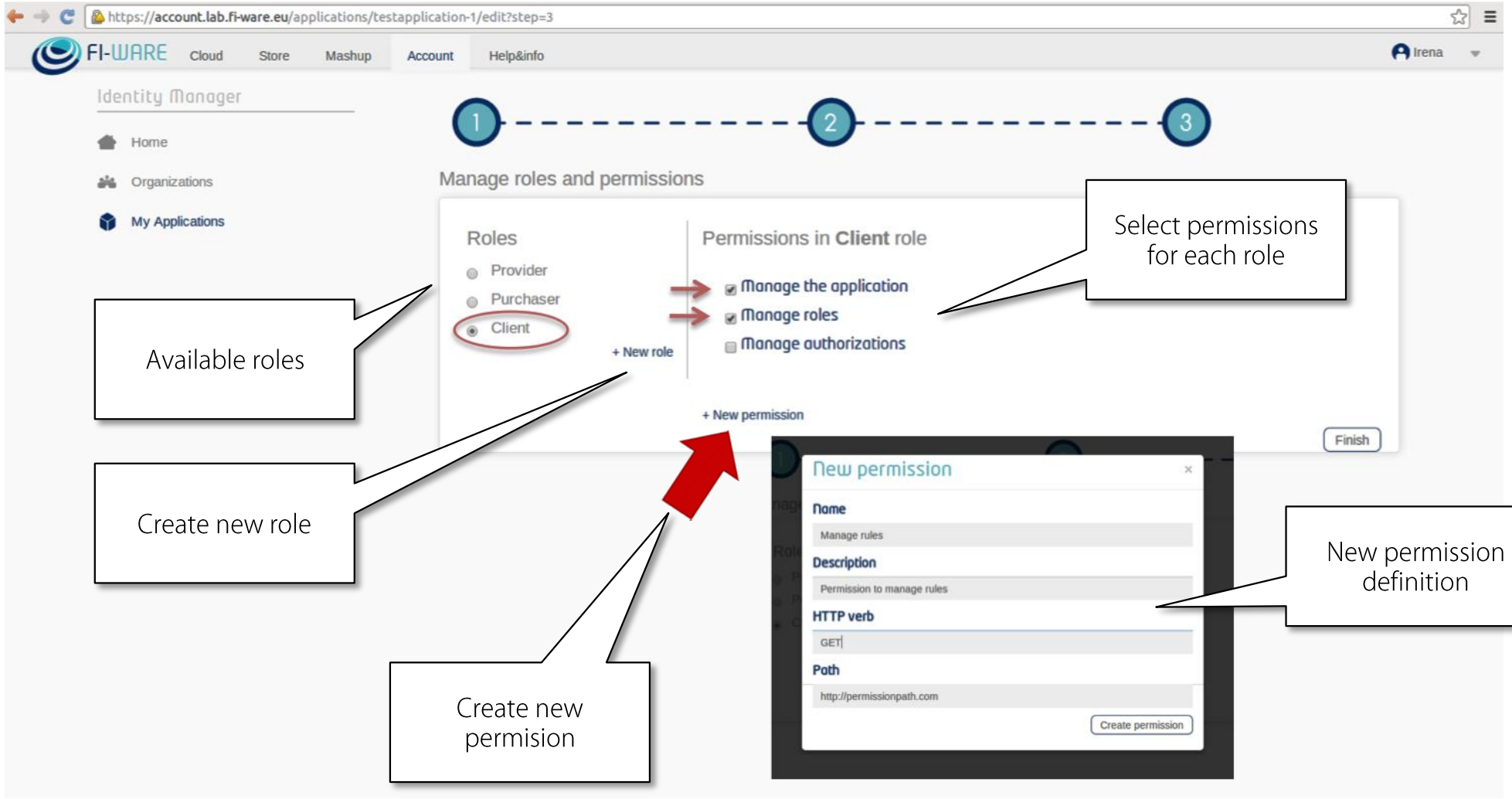
IdM– Organizations



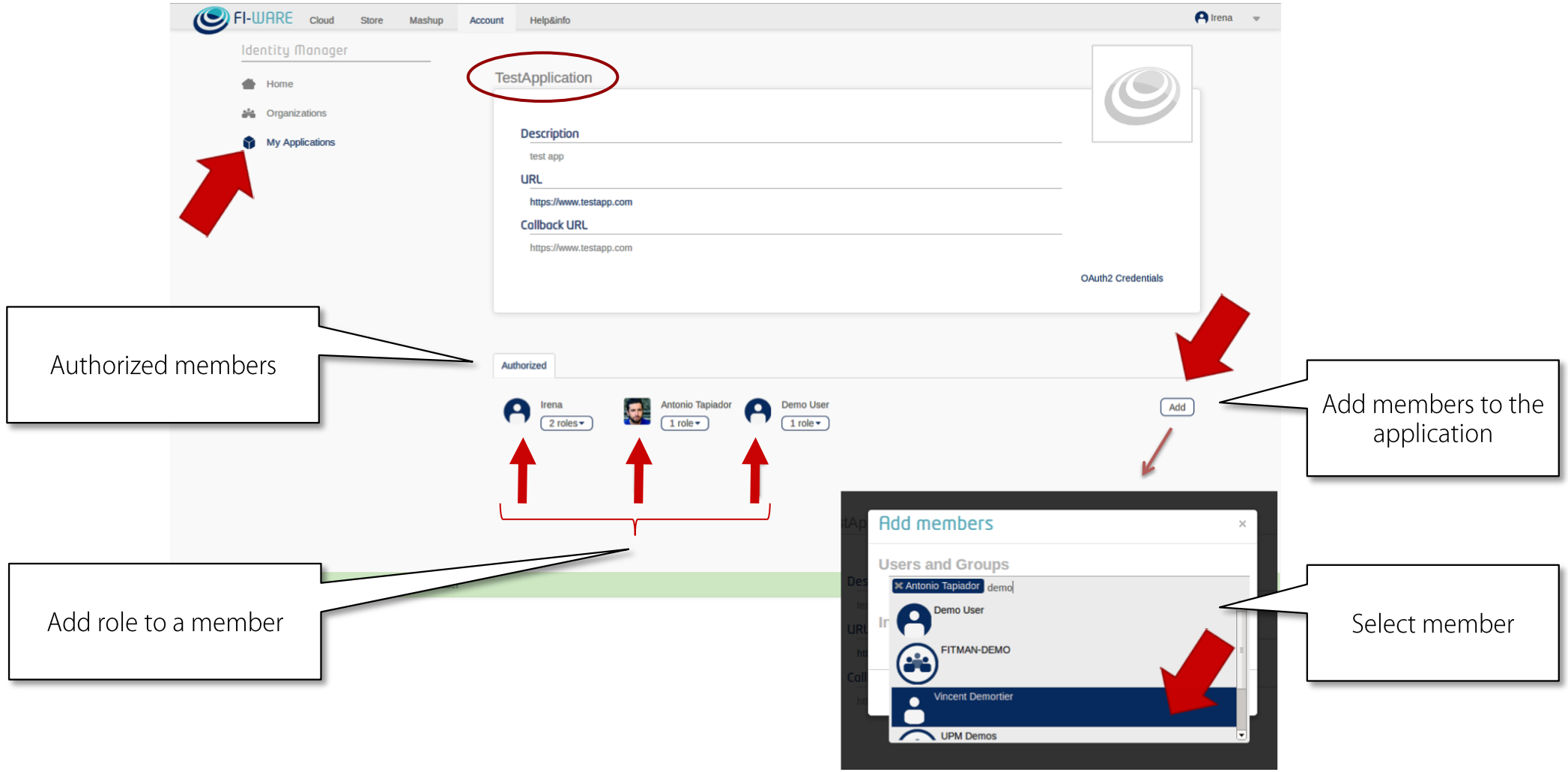
IdM– Application Registering



IdM– Application Roles and Permissions



IdM– Adding members and Managing roles



IdM– CL

Get a single user

```
GET /users/:id
```

```
id: 1,
 actorId: 1,
 nickName: "demo",
 displayName: "Demo user",
 email: "demo@fi-ware.eu",
 roles: [{
     id: 1,
     name: "Manager"
   },],
 organizations: [
   {
      id: 1,
      actorId: 2,
      displayName: "Universidad Politecnica de
Madrid",
      roles: [
        {
          id: 14,
          name: "Admin"
        } ]
   }
 ]
```

Get authenticated user

```
GET /user?token=12342134234023437
```

```
id: 1,
 actorId: 1,
 nickName: "demo",
 displayName: "Demo user",
 email: "demo@fi-ware.eu",
 roles: [{                          ...(same as before)
```

Get applications from actor

```
GET
/applications.json?actor_id=1&access_token=2YotnFZFEjr1z
CsicMWpAA
```

```
{
  id: 1,
  name: "Dummy",
  description: "FI-WARE demo application",
  url:"http://dummy.fi-ware.eu/"
}
```

Access Control GE- THALES Implementation

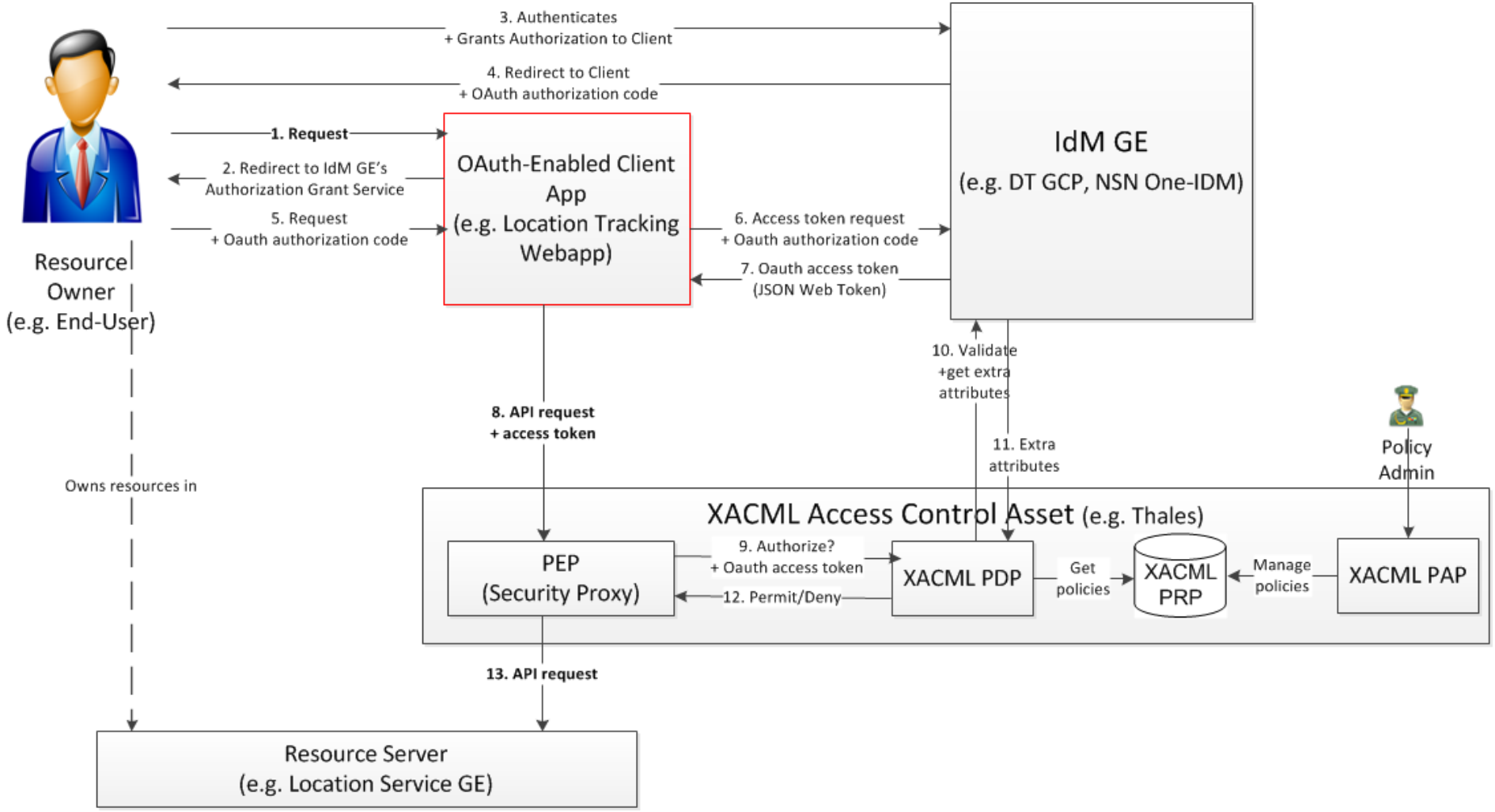
Access Control GE
THALES Implementation

Access Control GE provides an API to manage XACML-based access control policies and provide authorization decisions based on such policies and the context of a given access request.

This GE offers **Attribute-Based Access Control** based on:

- **Subject attributes:** the actor (human, program, device, etc.) requesting access to a resource.
- **Resource attributes:** entity on which subject requests to act upon (e.g. data, device, application, etc.)
- **Action attributes:** the action that the subject requests to perform on the resource (e.g. create, read, delete)
- **Environment attributes:** e.g. current time.

Access Control GE - Main interactions



Access Control GE - APIs

The Authorization Server provides the two APIs:

- **Policy Administration Point** (PAP): management of authorization policies, used for access control management.
- **Policy Decision Point** (PDP): evaluation of authorization decision requests, used for access control enforcement.

Policy Administration Point (PAP)

The PAP provides a RESTful API for **creating/updating policies** for a specific domain. The PAP is used by policy administrators to manage the policy repository from which the PDP loads the enforced policies.

HTTP requests to this API must be formatted as follows:

```
Method: PUT
Path: /domains/{domainId}/pap/policySet
Headers:
Content-Type: application/xml
Accept: application/xml
Body: XACML PolicySet as defined in the XACML 2.0
schema.
```

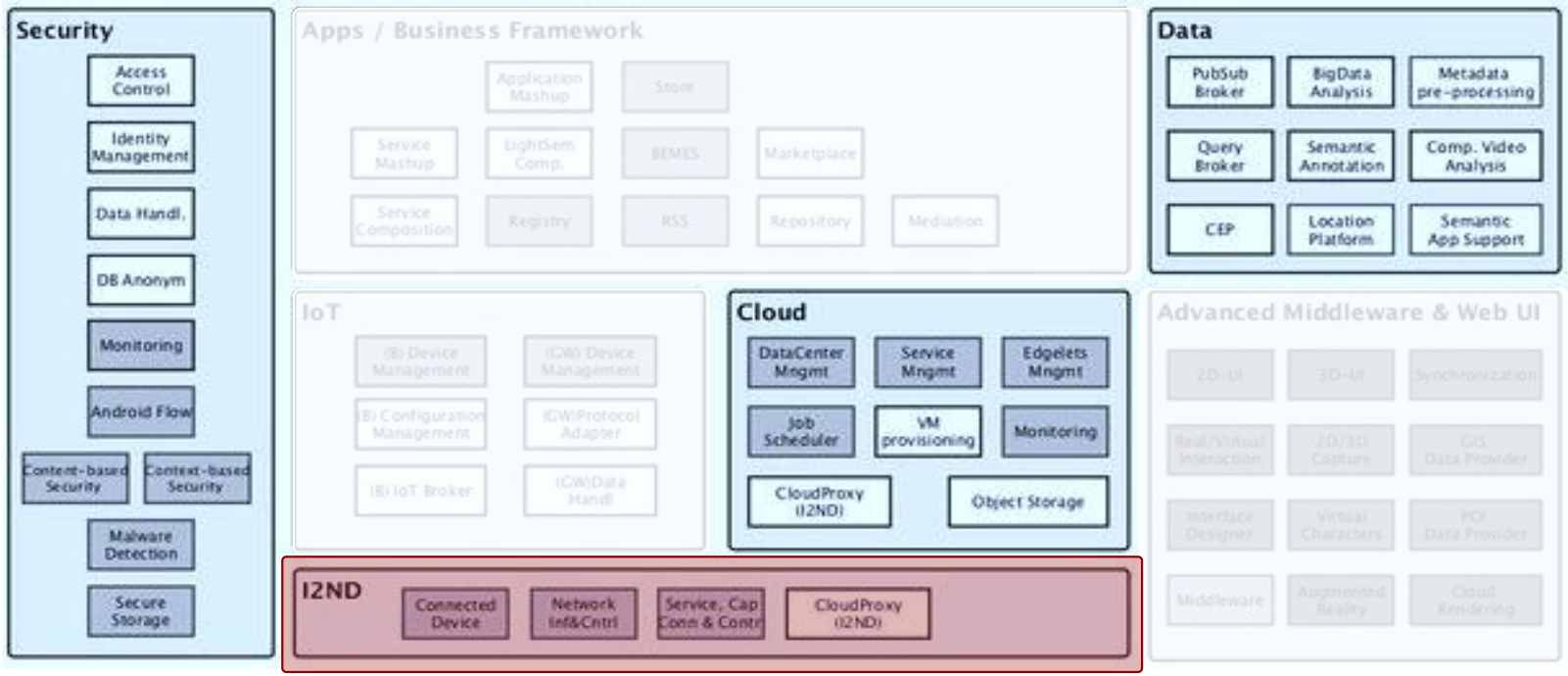
Policy Decision Point (PDP)

The PDP API returns an **authorization decision** based on the currently enforced policy, access control attributes provided in the request and possibly other attributes resolved by the PDP itself, such as the current time environment attribute. The authorization decision is typically Permit or Deny.

The HTTP request must be formatted as follows:

```
Method: POST
Path: /domains/{domainId}/pdp
Headers:
Content-Type: application/xml
Accept: application/xml
Body: XACML Request as defined in the XACML 2.0 schema.
```

INTERFACE TO NETWORK AND DEVICES

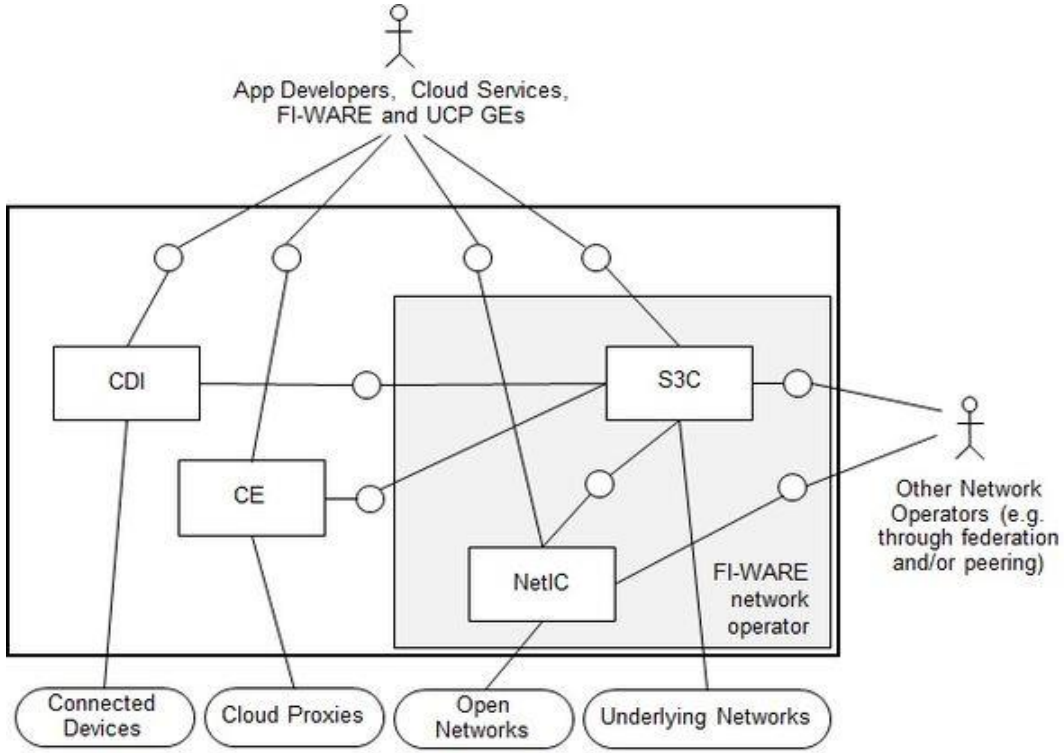


Architecture Overview

Security, Privacy and Trust in FI-WARE is mainly focusing on delivering tools and techniques to create secure services without sacrificing much of the desired functionality, usability, performance and cost efficiency is a great challenge

The I2ND architecture covers the following Generic Enablers (GEs):

- **CDI (Connected Device Interface)** towards the Connected Devices (mobile terminals, tablets, set top boxes with features such as remote access or exposure of status, sensors, etc.).
- **CE (Cloud Edge)** towards the Cloud Proxies that connect and control a set-up of nodes towards the Internet or/and an operator network
- **NETIC (NETwork Information and Control)** towards Open Networks that can be used for virtualization.
- **S3C (Service Capability, Connectivity and Control)** towards Underlying Networks that follow standards such as Next Generation Networks (NGNs)



Network Information and Control - OFNIC

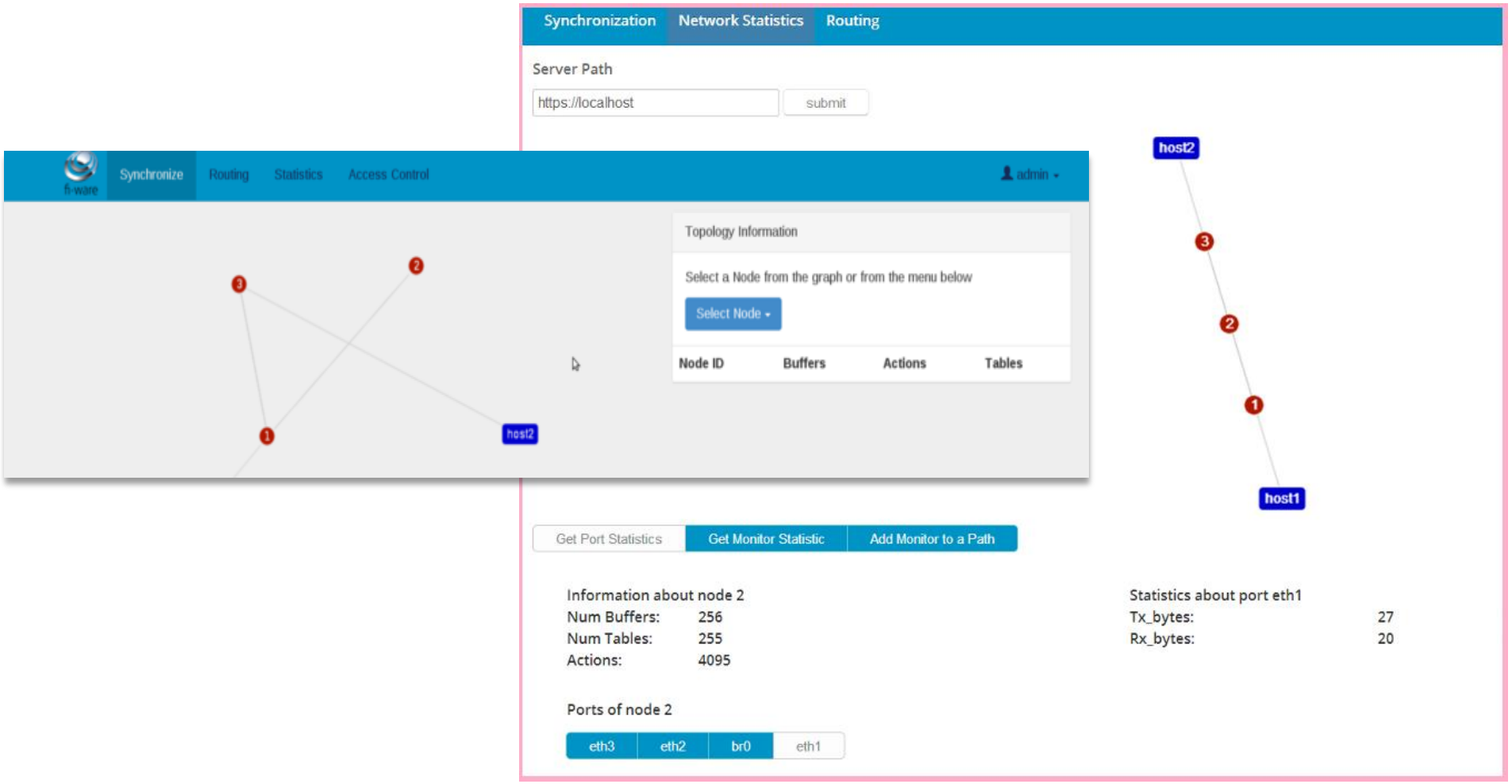
OFNIC
Network Information and Control

OFNIC is a reliable and distributed Software Defined Network (SDN) controller that enables the abstraction and virtualization of the resources and functionalities of OpenFlow-enabled networks.

OFNIC main capabilities:

- Offers a RESTful interface to **get information about the network topology**, components and elements either real or virtual.
- **Monitors the status** of the network
- **Provides near real-time data** about network statistics with different levels of granularity (flow, node, port).
- **Controls the network forwarding** capabilities
- Establishes end-to-end paths with **given quality of service/experience (QoS/QoE) requirements** between any two nodes in the network.

Network Information and Control - OFNIC



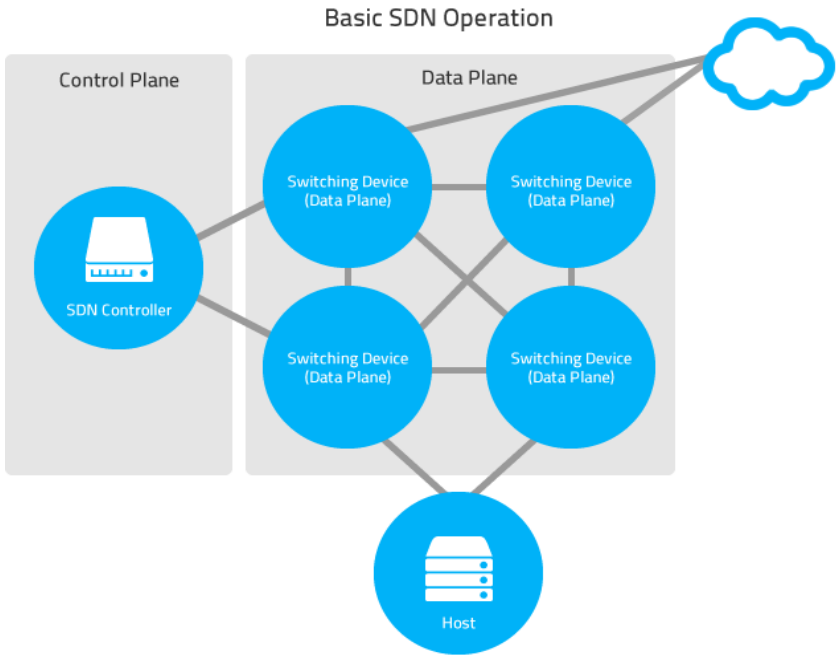
Network Information and Control – Use Cases

The beneficiaries of the **NETIC** interface include content providers, cloud hosting providers, context providers/brokers, and (virtual) network providers/operators. They might want to set up flows/virtual networks and may want to control such flows/virtual networks in order to respect pre-defined Service Level Agreements (SLAs), for example in terms of provided Quality of Service (QoS).

There are several use cases, for example :

1. A cloud hosting provider, in order to distribute the allocation of virtual machines (VM) to various locations, wants to know about the characteristics of the paths between the locations (e.g., delay, available capacity). Via the NetIC interface, he can request from the network provider (regularly or per scheduled event) the characteristics of the paths between his data centers.

2. A service provider may need a certain minimum link quality, e.g., for a high-definition live video streaming service. He will request via NetIC from the network provider the setup of a virtual connection with certain quality characteristics between the server and his client. NetIC implementations support Service Level Agreements (SLA) to guarantee that capacity will be available for a client on demand.



DOCUMENTATION - I

FIWARE architecture

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FI-WARE_Architecture

Cloud Hosting

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Cloud_Hosting_Architecture

http://catalogue.fi-ware.org/enablers/iaas-data-center-resource-management-ge-ibm-implementation

http://catalogue.fi-ware.org/enablers/object-storage-ge-fi-ware-implementation

http://catalogue.fi-ware.org/enablers/policy-manager-bosun

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Cloud.PaaS

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Cloud.ObjectStorage

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Cloud.DCRM

http://www.slideshare.net/fermingalan/developing-your-first-application-using-fi-ware-20130903?related=1

Context / Data Management

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Data/Context_Management_Architecture

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Data.BigData

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Data.ContextBroker

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Data.StreamOriented

http://catalogue.fi-ware.org/enablers/bigdata-analysis-cosmos

http://catalogue.fi-ware.org/enablers/publishsubscribe-context-broker-orion-context-broker

http://catalogue.fi-ware.org/enablers/stream-oriented-kurento

http://www.slideshare.net/FI-WARE/fi-ware-cosmosv7tech

http://www.slideshare.net/FI-WARE/orioncontextbroker-presentationdraft20141007141007111519conversiongate01?related=1

http://www.slideshare.net/izanmail/kurento-fiware-startup-weekend?related=4

DOCUMENTATION - II

Security

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Security_Architecture

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Identity_Management_Generic_Enabler

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Security.Context-based_security_%26_compliance

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Security.Access_Control_Generic_Enabler

http://catalogue.fi-ware.org/enablers/access-control-tha-implementation

http://catalogue.fi-ware.org/enablers/identity-management-keyrock

http://www.slideshare.net/flopezaguilar/setting-up-your-virtual-infrastructure-using-fi-lab-cloud-32388357?related=2

http://www.slideshare.net/flopezaguilar/idm-and-ac?related=3

http://www.slideshare.net/jcague/introduction-32012893?related=7

I2ND

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/Interface_to_Networks_and_Devices_(I2ND)_Architecture

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.I2ND.NetIC

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.I2ND.CDI

https://forge.fi-ware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.I2ND.CE

http://catalogue.fi-ware.org/enablers/network-information-and-control-ofnic-uniroma

THANK YOU
FOR YOUR ATTENTION

www.gradiant.org

Gradiant

Galician research and development center
in advanced telecommunications

2015

Gradiant